ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΚΑΙ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Δημιουργία και Αξιολόγηση Σημασιολογικών Επεξηγήσεων Μέσω Αντιπαραδειγμάτων

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Γεώργιος Φιλανδριανός

Συμβουλευτική Επιτροπή:: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάϊος 2025

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Πληροφορικής και Υπολογιστών
Εργαστήριο Τεχνητής Νοημοσύνης και Μηχανικής Μάθησης

Δημιουργία και Αξιολόγηση Σημασιολογικών Επεξηγήσεων Μέσω Αντιπαραδειγμάτων

Διδακτορικη Διατριβη

Γεώργιος Φιλανδριανός

Συμβουλευτική Επιτροπή: Γεώργιος Στάμου
Αθανάσιος Βουλόδημος
Μιχαήλ Βαζιργιάννης

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή τη 23η Μάϊου, 2025.

....
Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

....
Αθανάσιος Βουλόδημος
Επ. Καθηγητής Ε.Μ.Π.

....
Μιχαήλ Βαζιργιάννης
Καθηγητής Ecole Polytechnique

....
Κωνσταντίνα Νικήτα
Καθηγήτρια Ε.Μ.Π.

....
Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

....
Ευάγγελος Καρκαλέτσης
Δ/ντής Ερευνών ΕΚΕΦΕ

....
Χρυσούλα Ζέρβα
Ass. Professor
Instituto Superior Tecnico

Αθήνα, Μάϊος 2025

....

Γιωργος Φιλανδριανος
Διδάκτωρ Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η Τεχνητή Νοημοσύνη (ΤΝ) έχει σημειώσει σημαντική πρόοδο, μεταβαίνοντας από ερευνητικά πρωτότυπα σε ευρείας κλίμακας εφαρμογές στους τομείς της υγείας, των χρηματοοικονομικών, της ασφάλειας και των μεταφορών. Παρά την επιτυχία τους, τα μοντέλα ΤΝ συχνά λειτουργούν ως αδιαφανείς "μαύρα κουτιά", εγείροντας ανησυχίες σχετικά με την εμπιστοσύνη, την αποδοχή και τον κίνδυνο σε εφαρμογές υψηλού ρίσκου. Η επεξηγήσιμη τεχνητή νοημοσύνη αντιμετωπίζει αυτά τα ζητήματα αναπτύσσοντας μεθόδους που βελτιώνουν την ανθρώπινη κατανόηση σύνθετων μοντέλων. Η παρούσα διατριβή εστιάζει στις σημασιολογικές επεξηγήσεις μέσω αντιπαραδειγμάτων, οι οποίες καθορίζουν τις ελάχιστες τροποποιήσεις εισόδου που απαιτούνται για την αλλαγή της πρόβλεψης ενός μοντέλου ΤΝ. Παρουσιάζεται ένα ανεξάρτητο από το πεδίο εφαρμογής και του υποκείμενου μοντέλου πλαίσιο για τη δημιουργία αντιπαραδειγματικών εξηγήσεων, το οποίο δοκιμάστηκε σε πολλαπλές μορφές δεδομένων, όπως εικόνες, κείμενο και ήχος. Στο πλαίσιο αυτό, εξερευνώνται διάφορες αλγοριθμικές προσεγγίσεις, συμπεριλαμβανομένων των νευρωνικών δικτύων γραφών για δομημένα δεδομένα και μη νευρωνικών τεχνικών βελτιστοποίησης για τη σύνθεση αντιπαραδειγμάτων με την χρήση γραφών γνώσης.

Πέρα από τη δημιουργία, η παρούσα εργασία εισάγει μια νέα μεθοδολογία αξιολόγησης για την εκτίμηση της βελτιστότητας των αλγορίθμων παραγωγής επεξηγήσεων μέσω αντιπαραδειγμάτων, αξιοποιώντας μια προσέγγιση εμπνευσμένη από την τεχνική της αντίστροφης μετάφρασης. Αυτή η μέθοδος αξιολόγησης παρέχει βαθύτερη κατανόηση της σχέσης μεταξύ της ελαχιστοποίησης των τροποποιήσεων και της σημασιολογικής εγκυρότητάς τους, αποκαλύπτοντας ιδιότητες των αλγορίθμων δημιουργίας αντιπαραδειγμάτων που θα παρέμεναν αθέατες υπό τα παραδοσιακά πρότυπα αξιολόγησης.

Επιπλέον, το προτεινόμενο πλαίσιο επεκτείνεται πέρα από τις κλασικές εφαρμογές της επεξηγησιμότητας. Χρησιμοποιείται για την ανίχνευση παραισθήσεων σε μεγάλα οπτικο-γλωσσικά Μοντέλα και για τη λεπτομερή αξιολόγηση γενετικών μοντέλων σε εικόνες και οπτικοποίηση ιστοριών. Επιπρόσθετα, διερευνώνται οι συλλογιστικές ικανότητες των μεγάλων γλωσσικών μοντέλων, ιδιαίτερα στην επίλυση γρίφων, όπου αποδεικνύεται ότι η χρήση αντιπαραδειγμάτων στην είσοδο βελτιώνει την απόδοσή τους. Παράλληλα, μέσα από αντιπαραδειγματικές επεξηγήσεις αναδεικνύεται η επίδραση γνωστών γνωσιακών προκαταλήψεων, ιδίως όταν τα εν λόγω μοντέλα αξιοποιούνται ως συστήματα συστάσεων. Εκτενείς πειραματικές αξιολογήσεις επικυρώνουν την αποτελεσματικότητα του πλαισίου σε διαφορετικούς τομείς, αποδεικνύοντας τη δυνατότητα του να ενισχύσει την ερμηνευσιμότητα, την αξιοπιστία και τη γενίκευση τόσο στις επεξηγήσιμες εφαρμογές ΤΝ όσο και σε άλλες περιοχές της τεχνητής νοημοσύνης.

Λέξεις Κλειδιά: Επεξηγήσεις μέσω Αντιπαραδειγμάτων, Επεξηγήσιμη Τεχνητή Νοημοσύνη, Επεξεργασία Φυσικής Γλώσσας, Αξιολόγηση

Abstract

Artificial Intelligence (AI) has made significant strides, transitioning from research prototypes to large-scale deployments in healthcare, finance, security, and transportation. Despite their success, AI models often function as opaque black boxes, raising concerns about trust, adoption, and risk in high-stakes applications. Explainable AI (XAI) addresses these issues by developing methods to enhance human interpretability of complex models.

This dissertation focuses on counterfactual explanations, which determine the minimal input modifications required to alter an AI model's prediction. A domain-agnostic, black-box framework for counterfactual generation is introduced, applicable across multiple data modalities, including images, text, and audio. Within this framework, various algorithmic approaches are explored, including Graph Neural Networks (GNNs) for structured data and non-neural optimization techniques for counterfactual synthesis.

Beyond generation, this work introduces a novel evaluation methodology for assessing counterfactual optimality, specifically leveraging a back-translation-inspired approach to verify whether the applied modifications are truly minimal. This evaluation method provides deeper insights into the balance between the minimality of edits and their semantic validity, revealing properties of counterfactual generation algorithms that would otherwise remain obscured.

Additionally, the proposed framework extends beyond traditional XAI applications. It is leveraged for hallucination detection in Large Vision-Language Models (LVLMs) and fine-grained evaluation of generative models in both image and story generation. Furthermore, Large Language Models (LLM) reasoning capabilities are investigated, particularly in riddle-solving, where counterfactual-based interventions enhance logical reasoning in large-scale language models. At the same time, through counterexample-based explanations, the influence of well-known cognitive biases is highlighted, especially when such models are employed as recommendation systems.

Comprehensive empirical evaluations validate the framework's effectiveness across diverse domains, demonstrating its ability to enhance interpretability, robustness, and generalizability in both explainability and downstream AI applications.

Keywords: Counterfactual Explanations, Explainable AI, NLP, Evaluation

Acknowledgements

The research presented in this dissertation represents the culmination of five years of dedicated work at the AILS laboratory, under the supervision of Professor Giorgos Stamou. It was a unique and transformative period, covering pre-pandemic times, quarantines, and the gradual return to normality. Especially during the pandemic, the collaboration with my colleagues proved essential, and I am deeply thankful for their ongoing support, encouragement, and teamwork. Special recognition goes to Professor Giorgos Stamou, who skillfully organized and led the laboratory's virtual operations in a manner that effectively served everyone's needs.

The thesis is largely the result of collaborative efforts, and I extend my sincere gratitude to all my co-authors for their valuable contributions. In particular, I thank Eddie Dervakos, who guided and supported me since my earliest research steps as an undergraduate student. His passion, integrity, and dedication left a lasting impression and significantly influenced my research path. I am equally grateful to Maria Lymperaiou, one of my closest doctoral collaborators, with whom I co-authored numerous papers and co-adapted the proposed framework for generating counterfactual explanations across diverse tasks. Thanks also go to Konstantinos Thomas and Angeliki Dimitriou for their collaboration in developing the graph-based counterfactual explanation framework.

I also thank Orfeas Menis-Mastromichalakis for jointly developing a framework to evaluate counterfactual explanations, and Jason Liartis for adapting our approach from counterfactual explanations to prototype explanation systems. I extend my appreciation to Nikolaos Spanos, who significantly contributed to integrating our theoretical counterfactual explanation framework with image-generation systems, enabling the production of semantic counterfactual images in a black-box manner.

Special thanks to Theofanis Ganitidis, Prof. Konstantia Zarkogianni, and Prof. Konstantina Nikita from the smarty4covid project. My sincere gratitude goes to all members of AILS, particularly Vassilis Lyberatos, Paraskevi Theofilou, Nikos Haidos, Natalia Grigoriadou, Ilias Mitsouras, Vasilis Karabinis, Spyridon Kantarelis, Panagiotis Panayiotopoulos, Panagiotis Giadikiaroglou, Voula Pavlaki, and Christos Papadimitriou. I also warmly thank my CSLab collaborators: Angelos Vlachos, Petros Anastasiadis, Vassiliki Kostoula, Nikos Chalvatzis, Kostis Nikas, and Giorgos Gkoumas. Their contributions created a supportive and friendly lab environment, filled with memorable moments, laughter, and meaningful discussions.

I offer special acknowledgment to Chryssoula Zerva for her critical support across numerous projects and for her invaluable guidance throughout my doctoral journey. Her mentorship has profoundly shaped my academic development and achievements. I'm also deeply grateful to Prof. Maria Symeonaki, whose ideas, high-quality work, and dedication greatly influenced my research beyond AI.

Lastly, I express my deepest gratitude to my family—my parents, Vassilis, whose memory and guidance continue to inspire me every day, and my mother, Tasia; Sofia Savva, who steadfastly supported me throughout my undergraduate, graduate, and doctoral studies; my sister Chryssa, my brother Konstantinos. I also thank my closest friends, Vassilis Cheilas and Nikos Xydeas, and my dear friends Thanasis Daskalakis and Marios Kourabas, whose friendship, support, and encouragement have been invaluable.

Filandrianos Giorgos, May 2025

Contents

List of Figures

172figure.caption.253

Glossary - Γλωσσάριο

ante hoc explanation	Εκ των προτέρων εξήγηση
Bias	Προκατάληψη
Black box explanation	Εξήγηση μαύρου κουτιού
Classifier	Ταξινομητής
Classifier	Ταξινομητής
Conceptual counterfactuals	Εννοιολογικές επεξηγήσεις μέσω αντιπαραδειγμάτων
Convolutional Neural Network	Συνελικτικό Νευρωνικό Δίκτυο
Counterfactual Explanation	Αντιπαραϑετική Επεξήγηση
Counterfactual explanations	Επεξηγήσεις μέσω αντιπαραδειγμάτων
Description Logics	Λογικές Περιγραφής
Description Logics	Περιγραφικές Λογικές
Exemplar	Δείγμα
Explainability	Εξηγησιμότητα
Explanation Dataset	Σύνολο Δεδομένων Εξήγησης
Explanation Dataset	Σύνολο Δεδομένων Εξήγησης
Feature Importance	Εξηγήσεις Σημαντικότητας Χαρακτηριστικών
Graph Neural Networks	Νευρωνικά δίκτυα γράφων
Inherently Interpretable	Εγγενώς Ερμηνεύσιμο
Interpretability	Ερμηνευσιμότητα
Knowledge Base	Βάση Γνώσης
Knowledge Graph	Γράφος Γνώσης
Large Language Models	Μεγάλα Γλωσσικά Μοντέλα
Level of Abstraction	Επίπεδο Αφαίρεσης
Local Explanation	Εξήγηση Μοντέλου σε Ένα Δεδομένο
Model-agnostic Explanation Method	Μέϑοδος εξήγησης ανεξάρτητη από το μοντέλο
Model-specific Explanation Method	Μέϑοδος εξήγησης για συγκεκριμένη οικογένεια μοντέλων
Pixel	Εικονοστοιχείο
Prompt	Προτροπή
White box Explanation	Εξήγηση άσπρου κουτιού (με πρόσβαση στο μοντέλο)
Search engine optimization	Βελτιστοποίηση μηχανών αναζήτησης
Cognitive biases	Γνωσιακές προκαταλήψεις

Chapter 1

Εκτεταμένη Περίληψη στα Ελληνικά

Η Τεχνητή Νοημοσύνη (ΤΝ) έχει γνωρίσει εντυπωσιακή ανάπτυξη τα τελευταία χρόνια, περνώντας από πιλοτικά ερευνητικά στάδια σε εφαρμογές μεγάλης κλίμακας που αφορούν την υγεία, τα χρηματοοικονομικά, την ασφάλεια και τις μεταφορές. Παρά το γεγονός ότι τα συστήματα ΤΝ διακρίνονται σε τομείς όπως η αναγνώριση εικόνων, η επεξεργασία φυσικής γλώσσας και η αυτοματοποιημένη λήψη αποφάσεων, ο πολύπλοκος τρόπος λειτουργίας τους συχνά παραμένει ασαφής. Αυτή η αδιαφάνεια μπορεί να κλονίσει την εμπιστοσύνη των χρηστών, να καθυστερήσει την ενσωμάτωση της ΤΝ σε νέες εφαρμογές και, σε κρίσιμες περιπτώσεις, να εγκυμονεί σοβαρούς κινδύνους. Σε αυτό το πλαίσιο, η *Επεξηγήσιμη Τεχνητή Νοημοσύνη (Explainable AI, XAI)* έχει αναδειχθεί ως καίρια ερευνητική κατεύθυνση, επιδιώκοντας την ανάπτυξη μεθόδων και εργαλείων που βοηθούν τους ανθρώπους να κατανοούν και να αλληλεπιδρούν αποτελεσματικότερα με πολύπλοκα μοντέλα.

Μέσα σε αυτό το ερευνητικό πεδίο, οι *οι εξηγήσεις μέσω αντιπαραδειγμάτω ν (counterfactual explanations)* έχουν ξεχωρίσει, καθώς προσφέρουν σαφή εικόνα για το πώς ένα σύστημα ΤΝ μπορεί να αλλάξει την έξοδό του εφόσον διαφοροποιηθεί κάποιο χαρακτηριστικό της εισόδου. Αντί να εστιάζουν αποκλειστικά στο "γιατί" προέκυψε μια συγκεκριμένη πρόβλεψη, οι αντιπαραδειγματικές προσεγγίσεις αναδεικνύουν το "πώς" μπορεί να επιτευχθεί μια εναλλακτική έκβαση, συνήθως τροποποιώντας περιορισμένο αριθμό παραμέτρων. Αυτό το γνώρισμα είναι ιδιαίτερα χρήσιμο σε εφαρμογές οικονομικής και ιατρικής φύσεως, όπου οι χρήστες μπορούν να αξιοποιήσουν οδηγίες τύπου "Μειώνοντας το υπόλοιπο της πιστωτικής κάρτας σας κατά X, μπορεί να εγκριθεί το δάνειό σας" ή "Εάν η πίεση του ασθενούς μειωνόταν, το αποτέλεσμα της διάγνωσης θα ήταν διαφορετικό".

Παρά τα οφέλη τους, πολλές από τις σημερινές τεχνικές αντιπαραδειγματικών εξηγήσεων επικεντρώνονται σε χαμηλό επίπεδο λεπτομέρειας, όπως είναι η αλλαγή μεμονωμένων εικονοστοιχείων (pixels) σε μια εικόνα ή η αντικατάσταση συγκεκριμένων λέξεων σε ένα κείμενο. Αν και τέτοιες αλλαγές μπορούν να είναι ακριβείς, δεν εναρμονίζονται πάντοτε με τον τρόπο που οι άνθρωποι αντιλαμβάνονται και αλληλεπιδρούν με τον κόσμο. Για παράδειγμα, για τους περισσότερους ανθρώπους είναι πιο φυσικό να περιγράψουν ένα αντικείμενο ως "κόκκινο" παρά να αναλύσουν τους αριθμητικούς τιμές των χρωματικών καναλιών. Παρομοίως, στον χώρο της επεξεργασίας κειμένου, οι χρήστες συνηθίζουν να μιλούν για *θέμα ή συναίσθημα* και όχι για τροποποιήσεις σε μεμονωμένα γράμματα ή αλλάγες σε κάποια διανυσματική αναπαράσταση. Αυτό το χάσμα ανάμεσα στα βασικά χαρακτηριστικά και στις πιο αφηρημένες, ανθρώπινες έννοιες προκαλεί δυσκολίες στην προσπάθεια να διαμορφωθούν εξηγήσεις που είναι ταυτόχρονα ακριβείς και κατανοητές.

Για να αντιμετωπιστεί αυτή η πρόκληση, έχει αρχίσει να κερδίζει έδαφος η ιδέα των *εννοιολογικών επεξηγήσεων μέσω αντιπαραδειγμάτων (conceptual counterfactuals)*. Σε αυτό το πλαίσιο, οι αλλαγές δεν αφορούν μικροεπεμβάσεις σε επίπεδο raw χαρακτηριστικών, αλλά ανώτερες, πιο αφηρημένες κατηγορίες που συνδέονται με την ανθρώπινη αντίληψη. Για παράδειγμα, σε μια εικόνα, μπορεί να μας ενδιαφέρει αν υπάρχει "ριγέ μοτίβο", "σφαιρικό σχήμα" ή "τριχωτή υφή". Στον ήχο, ενδέχεται να εστιάζουμε σε "εύρος συχνοτήτων" ή "ταχύ ρυθμό", ενώ στο κείμενο μπορεί να εξετάζουμε "συναίσθημα", "θέμα" ή "βαθμό ευγένειας". Με τη μετάβαση σε αυτές τις ευρύτερες, εννοιολογικές δομές, οι αντιπαραδειγματικές εξηγήσεις αναδεικνύουν *ποια* αφηρημένα χαρακτηριστικά πρέπει να τροποποιηθούν ώστε να αλλάξει η έξοδος ενός μοντέλου, αποφεύγοντας παράλληλα εκτεταμένες και δυσνόητες αλλαγές σε χαμηλότερο επίπεδο.

Η παρούσα διατριβή διευρύνει και ενισχύει αυτό το πλαίσιο των εννοιολογικών επεξηγήσεων μέσω αντιπαραδειγμάτων. Προτείνει νέους αλγορίθμους, αρχιτεκτονικές και μετρικές αξιολόγησης που, συνολικά, καταδεικνύουν την ευελιξία και την πρακτική αξία της εννοιοκεντρικής προσέγγισης σε ποικίλα είδη δεδομένων, από εικόνες και φυσική γλώσσα έως γράφους και ηχητικά αρχεία. Επιπρόσθετα, παρουσιάζει πώς η έμφαση στις αφηρημένες έννοιες ενισχύει τη σαφήνεια στην επεξήγηση των αποφάσεων ενός μοντέλου, ενώ ταυτόχρονα μπορεί να απλοποιήσει τη διαδικασία εντοπισμού και εφαρμογής στοχευμένων παρεμβάσεων, διατηρώντας το συνολικό νόημα αλώβητο. Τέλος μελετά προεκτάσεις των τεχνικών που αναπτύχθηκαν και σε άλλα πεδία, όπως η αυτόματη αξιολόγηση και αναγνώριση λαθών από δημιουργικά σηστήματα.

1.1 Πλαίσιο παραγωγής ενοιολογικών επεξηγήσεων μέσω αντιπαραδειγμάτων

Το παρόν κεφάλαιο παρουσιάζει μια μεθοδολογία για τη δημιουργία *επεξηγήσεων μέσω αντιπαραδειγμάτων* που εστιάζουν σε **εννοιολογικά** και **σημασιολογικά** χαρακτηριστικά, αντί απλώς να χειρίζονται ακατέργαστα δεδομένα (π.χ. pixel, σύμβολα κειμένου ή αριθμητικές τιμές). Στηρίζεται σε μια *Βάση Γνώσης*, διατυπωμένη σε Λογικές Περιγραφής (Description Logics, DL), και σε ένα *Explanation Dataset*, που αποδίδει σε κάθε δείγμα (π.χ. εικόνα, ήχο ή κείμενο) ένα σύνολο εννοιών και σχέσεων. Μέσω αυτών, οι αντιπαραδειγματικές εξηγήσεις αποσαφηνίζουν όχι μόνο το πώς μπορεί να αλλάξει η πρόβλεψη ενός μοντέλου, αλλά και το *γιατί*, φωτίζοντας τις υψηλού επιπέδου σημασιολογικές αλλαγές που προκαλούν τις διαφοροποιήσεις στην έξοδο.

1.1.1 Κίνητρο για εννοιολογικές επεξηγήσεις μέσω αντιπαραδειγμάτων

Συμβατικές αντιπαραδειγματικές μέθοδοι συχνά δρουν στο μικροεπίπεδο των δεδομένων (π.χ. αλλαγές σε λίγα pixel) και ναι μεν πετυχαίνουν την αλλαγή πρόβλεψης, όμως δεν εξηγούν με ανθρώπινα κατανοητό τρόπο *γιατί* συγκεκριμένα pixels χρειάζονται τροποποίηση. Αντίθετα, το κεφάλαιο αυτό προτείνει τη χρήση επεξηγήσεων στο *εννοιολογικό επίπεδο*. Για παράδειγμα, αντί η επεξήγηση να είναι της μορφής "άλλαξε τις τιμές του κόκκινου καναλιού των pixel", μετασχηματίζεται σε "αν το *δωμάτιο* ήταν *σκούρο κόκκινο* αντί για *πράσινο*, το μοντέλο θα άλλαζε πρόβλεψη". Έτσι, οι επεξηγήσεις γίνονται πιο κατανοητές και προσεγγίζουν τη φυσική σκέψη των χρηστών. Επιπλέον μέσω αυτής της μορφής των επεξηγήσεων ο ανθρώπος μπορεί να κατανοήσει το μηχανισμό λήψης αποφάσεων ένος συστήματος τεχνητής νοημοσύνης και να επαναλάβει της αλλάγες χωρίς τη χρήση των αλγορίθμων. Για παράδειγμα αν για ένα σύστημα η αλλάγη μιας κλάσης από "υπνοδωμάτιο" σε "κτηνιατρειο" προυποθέτει την εισαγωγή ενός ζώου, τότε ο ανθρώπος μπορει ευκόλα να αντιλειφθεί, να επικοινωνήσει, να δοκιμάσει ο ίδιος σε νέες εικόνες αυτή την αλλάγη καθώς και να αξιλογήσει αν αυτό αποτελεί λανθασμένο κριτήριο απόφασης για τον ταξινομητή ώστε να προσπαθήσει, αν χρειαστει, να το επιλύσει. Από την άλλη, σε επίπεδο pixel αυτό μπορεί να είναι ως "κάνε πιο καφέ με μαυρές γραμμές 100 γειτονικά pixels μιας εικόνας", ώστε να δινεται η εντύπωση ενός ζώου στην εικόνα στα μάτια του ταξινομητή (classifier). Παρόλα αυτά η δευτέρη εξήγηση δεν συμβαίνει συστηματικά καθώς το χρώμα και το μέγεθος ένος ζώου δεν είναι σταθερό, πράγμα που μπορεί να δημιουργεί σημαντική σύγχυση στο χρήστη, σχετικά με τη μέθοδο απόφασης του ταξινομητή. Αυτό δεν αφήνει τον χρήστη να κατανοήσει τον τρόπο με τον οποίο ο ταξινομητής παίρνει αποφάσεις.

1.1.2 Βάση Γνώσης και Explanation Dataset

Ένα *Explanation Dataset* αποτελείται από δείγματα (*exemplars*), δηλαδή πραγματικά ή συνθετικά δείγματα (π.χ. εικόνες ή αρχεία ήχου), που συνοδεύονται από:

- **Εννοιολογικό Περιεχόμενο:** Ένα σύνολο εννοιών (π.χ. Dog, Bedroom, Cough) και σχέσεων (π.χ. depicts, hasSymptom).
- **Αντιστοίχιση στα χαρακτηριστικά του μοντέλου:** Το exemplar χαρτογραφείται σε ένα *feature vector* που χρησιμοποιεί ο ταξινομητής.

Η εννοιολογική πληροφορία προέρχεται από μια *Βάση Γνώσης* (ενσωματωμένη σε DL), η οποία περιέχει:

- **ABox (Assertional Box):** Δηλώσεις τύπου $\mathsf{Dog}(\mathsf{Lassie})$, $\mathsf{isIn}(\mathsf{Lassie}, \mathsf{Garden})$.
- **TBox (Terminological Box):** Ιεραρχικές σχέσεις, π.. $\mathsf{Dog} \sqsubseteq \mathsf{Animal}$, $\mathsf{Animal} \sqsubseteq \mathsf{LivingThing}$.

Οι οντολογικές σχέσεις επιτρέπουν τον ορισμό της σημασιολογικής *αποστάσης* μεταξύ εννοιών (π.χ. "Cat" και "Ant" είναι πιο απομακρυσμένες από ό,τι "Cat" και "Dog"), βοηθώντας στον ορισμό κόστους για αλλαγές (edits).

1.1.3 Ορισμός Εννοιολογικών Αντιπαραδειγματικών

Οι επεξηγήσεις μέσω αντιπαραδειγμάτων ορίζονται ως *σημασιολογικές επεμβάσεις* (semantic edits) στην ABox αναπαράσταση ενός exemplar. Αν ένας ταξινομητής κατατάσσει ένα exemplar e στην κλάση A, αλλά στόχος είναι η κλάση B, αναζητείται το δείγμα c το οποίο ανήκει ήδη στην κλάση B και απέχει την *ελάχιστη σημασιολογική απόσταση* από το e. Τα *edits* (π.. αντικατάσταση Cat → Dog) που μετατρέπουν το e στο c συνιστούν την τοπική αντιπαραδειγματική εξήγηση. Οι λειτουργίες περιλαμβάνουν:

- **Αντικατάσταση εννοιών** ($e_{\mathsf{Dog}\rightarrow\mathsf{Cat}}$).
- **Αντικατάσταση ρόλων** ($e_{r\rightarrow s}$).
- **Προσθήκη εννοιών ή ρόλων** ($e_{\top\rightarrow\mathsf{Cat}}$ ή $e_{\top\rightarrow\mathsf{on}}$).
- **Αφαίρεση εννοιών ή ρόλων** ($e_{\mathsf{Cat}\rightarrow\top}$ ή $e_{\mathsf{on}\rightarrow\top}$).

Για την ευρέση του δείγματος που απέχει την ελάχιστη σημασιολογική απόσταση, θα πρέπει να οριστεί ένα *κόστος* για κάθε επέμβαση, αξιοποιώντας την υπάρξη του TBox, ώστε να υπολογίζεται πόσο "μεγάλη"ή "μικρή" είναι κάθε αλλαγή.

1.1.4 Τοπικές και Καθολικές (Global) Επεξηγήσεις μέσω αντιπαραδειγμάτων

Τοπικές Εξηγήσεις: Αφορούν ένα μόνο παράδειγμα: π.. "Αν αφαιρούσα τη $\mathsf{Cat}(b)$ και πρόσθετα τη $\mathsf{Pillow}(b)$, η εικόνα e_1 θα ταξηνομούταν από τον ταξινομητή ως *Bedroom* αντί για *Veterinarian Office*."

Καθολικές Εξηγήσεις: Η ιδέα των *global* επεξήγησης βασίζεται στη συγκέντρωση πολλών τοπικών αντιπαραδειγμάτων. Αθροίζοντας τα edits για πολλούς exemplars με ορισμένα κοινά χαρακτηριστητικά όπως για παράδειγμα ότι αρχικά ανήκουν κλάση A και μεταβαίνουν στη κλάση B, μπορούν να εντοπιστούν οι πιο *συχνές* τροποποιήσεις. Έτσι αποκαλύπτονται **συστηματικές** συσχετίσεις ή προκαταλήψεις του μοντέλου (π.. "Η έννοια Animal προστίθεται συχνά όταν το μοντέλο αλλάζει από *Bedromm* σε *Veterinarian Office*").

1.1.5 Εφαρμογή και Υλοποίηση

Το κεφάλαιο εισάγει αλγορίθμους για:

1. **Δημιουργία γράφου εξηγήσεων:** Κάθε examplar είναι ένας κόμβος, και οι ακμές συνδέουν ζεύγη εξέμπλαρ εφόσον μπορούμε να τα μετατρέψουμε το ένα στο άλλο με μια ακολουθία edits. Το βάρος των ακμών αυτό ίσουτε με το κόστος των σημασιολογικών αλλάγων μεταξύ των δυο examplar που αυτή συνδεεί.
2. **Υπολογισμός κόστους μεταξύ δυο εννοιών:** Γίνεται μέσω της ελάχιστης απόστασης μεταξύ εννοιών ή ρόλων στο γράφο της TBox.
3. **Υπολογισμός κόστους μεταξύ δυο examplars:** Προτείνονται αλγόρθμοι για τον υπολογισμό του *graph edit distance*, χρησιμοποιώντας διαφορετικά επίπεδα πληροφορίας κάθε φορά (π.χ. με χρήση μόνο των αντικειμένων ή την εσαγωγή μερικής πληροφορίας των ακμών).

Επιπλέον, μέσω του προτεινόμενου πλαισίου δίνεται η δυνατότητα στον χρήστη να ορίζει αντικαταστάσεις οι οποίες είναι αδύνατο να πραγματοποιηθούν. Για παράδειγμα, είναι αδύνατο να μειωθεί η ηλικία ενός ατόμου. Πρακτικά, αυτό υλοποιείτε μεσω της εκχώρησης άπειρου κόστους για συγκεκριμένες αντικαταστάσεις π.χ. του edit $\mathsf{Young}\rightarrow\mathsf{Old}$. Με αυτόν τον τρόπο εξασφαλίζεται η παραγωγή feasible αντιπαραδειγμάτων.

1.1.6 Πειράματα

CLEVR-Hans3 (Τεχνητά Δεδομένα): Ένα πρώτο βήμα της ανάλυσης είναι η αξιολόγηση του προτεινόμενο πλαισίου σε ένα ελεγχόμενο περιβάλλον, όπου τόσο τα χαρακτηρηστικά των examplars, όσο και τα χαρακτηριστικά που ελέγχει ο ταξινομητής είναι προκαθορισμένα. Για τον σκοπό αυτό επιλέχθηκε το σύνολο δεδομένων CLEVR-Hans3, όπου περιέχει συνθετικές εικονες από αντικείμενα με γνωστό σχήμα, μέγεθος, υφή και χρώμα. Επίσης οι ταξινομητές που είναι εκπαιδευμένοι με το συγκεκριμένο σύνολο δεδομένων περιέχουν γνωστά σε ένα biases (προκαταλήψεις) τα όποια εμπλέκονται στη μέθοδο ταξινόμησης του. Έτσι μέσω της αξιολόγησης στο συγκεκριμένο σύνολο δεδομένων δύναται η ευκαιρία να αξιολογηθεί η αποτελεσματικότητα των μεθόδων που προτείνεται στη παρούσα διατριβή σε ένα ελεγχόμενο περιβάλλον.

COCO & Places (Πραγματικά Εικόνες): Έπειτα από την παράγωγη επεξηγήσεων σε ένα συνθετικό περιβάλλον σειρά έχει ο πειραματισμός με ένα πραγματικό σύνολο δεδομένων, ώστε να αξιολογηθεί η αποτελεσματικότητα του πλαισίου που προτάθηκε σε πραγματικά προβλήματα. Έτσι χρησιμοποιώντας ontologies που περιγράφουν τον πραγματικό κόσμο (π.χ. WordNet) για την περιγραφή αντικειμένων ("bed", "cat", "dining_table"), και δεδομένα από dataset που φέρουν σημασιολογική πληροφορία όπως το COCO δύναται η δυνατότητα εξαγωγής σημασιολογικών επεξηγήσεων μέσω αντιπαραδειγμάτων σε προβλήματα του πραγματικού κόσμου, όπου τα biases των ταξινομητών δεν είναι γνωστά εκ των προτέρων. Τέλος στην ίδια κατεύθυνση δοκιμάστηκε και η εφαρμογή του παρόντος πλαισίου και σε εφαρμογές όπου κανένα σημασιολογικό χαρακτηριστικό δεν συνοδεύει τους examplars αλλά αυτά εξήχθησαν αυτόματα μέσω διαφόρων τεχνικών εξαγωγής πληροφορίας όπως στην περίπτωση των εικόνων συστημάτων εξαγωγής γράφων σκηνής από εικόνες.

Ιατρική Διάγνωση COVID-19 (Audio): Τέλος δοκιμάστηκε και η παραγωγή επεξηγήσεων σε συστήματα που δέχονται δεδομένα ήχου σαν εισόδο και σε εφαρμογή όπου η επεξηγησημότητα είναι ζωτηκής σημασίας. Συγκεκριμένα οι αλγόριθμοι δοκιμάστηκαν σε ταξινομητές οι όποιο προβλέπουν αν ένα άτομο είναι θετικό στο COVID-19 με βάση το βήχα τους.

1.1.7 Κύρια Συμπεράσματα και Μελλοντική Εργασία

- **Καθοριστική Σημασία της Σημασιολογίας:** Οι χρήστες κατανοούν πιο εύκολα εξηγήσεις που αναφέρονται πιο αφηρημένα χαρακτηρηστικά όπως το "animal" και "pillow" παρά για "pixel-level" αλλαγές.
- **Εντοπισμός Προκαταλήψεων:** Σε πολλαπλές μελέτες (CLEVR-Hans3, COVID-19) αποκαλύφθηκαν κρυφές στρεβλώσεις, οφειλόμενες σε μερικώς συσχετισμένες έννοιες ή στατιστικές ιδιαιτερότητες των δεδομένων.
- **Κλιμακωσιμότητα:** Παρόλο που η σημασιολογική επεξεργασία έχει κόστος, η προσέγγιση που προτάθηκε επιτρέπει τη χρήση της σε εφαρμογές με ευρεία κλίμακα δεδομένων.
- **Ευελιξία Κόστους:** Ρυθμίσεις του κόστους δίνουν πρακτική *δράση* στις εξηγήσεις, αποφεύγοντας μη ρεαλιστικές ή ανέφικτες αλλαγές.
- **Γενίκευση σε Περισσότερα Είδη Δεδομένων:** Η ίδια μεθοδολογία μπορεί να εφαρμοστεί σε κείμενο ή πίνακες, αρκεί να υπάρχει μια σχετική οντολογία για τις εκάστοτε έννοιες.

Συνοψίζοντας, το κεφάλαιο αυτό αναδεικνύει τη δυναμική των *εννοιολογικών αντιπαραδειγμάτων* ως εργαλείο ερμηνείας και εντοπισμού σφαλμάτων ή προκαταλήψεων. Η μετάβαση από αλλαγές χαμηλού επιπέδου σε πιο αφηρημένες και κατανοητές έννοιες επιτρέπει στους ειδικούς αλλά και στους τελικούς χρήστες να *εμπιστευτούν* καλύτερα τα μοντέλα τεχνητής νοημοσύνης και να τα βελτιώσουν όπου αυτό κρίνεται αναγκαίο.

1.2 Παραγωγή σημασιολογηκών επεξηγήσεων μέσω αντιπαραδειγμάτων με χρήση γράφων

Το κεφάλαιο αυτό εστιάζει στην ενσωμάτωση των *σχέσεων* ανάμεσα στις έννοιες (concepts), με στόχο τη δημιουργία σημασιολογικών επεξηγήσεων μέσω αντιπαραδειγμάτων που προσεγγίζουν με μεγαλύτερη ακρίβεια τον τρόπο με τον οποίο ένα μοντέλο τεχνητής νοημοσύνης επεξεργάζεται σύνθετες εισόδους. Παρουσιάζονται δύο συμπληρωματικές προσεγγίσεις για την εκμετάλλευση της πληροφορίας των ακμών: η πρώτη αφορά το **την ενσωμάτωση μερικής πληροφορίας των ακμών στις έννοιες**, που μετατρέπει κάθε συνιστώσα του γραφήματος σε "σύνολα συνόλων" εννοιών, ενώ η δεύτερη αξιοποιεί τα **Νευρωνικά δίκτυα γράφων - Graph Neural Networks (GNNs)** με σκοπό τη διατήρηση και την επεξεργασία της πλήρους δομής του γραφήματος για πλουσιότερες εξηγήσεις.

1.2.1 Ενσωμάτωση πληροφορίας των ακμών στις έννοιες

Ένα κεντρικό εύρημα αυτού του κεφαλαίου είναι ότι τα προβλήματα ταξινόμησης στον πραγματικό κόσμο εξαρτώνται συχνά όχι μόνο από το *ποια* αντικείμενα υπάρχουν, αλλά και *πώς* αυτά συσχετίζονται. Ωστόσο, ο υπολογισμός του πλήρους κόστους τροποποίησης ενός γραφήματος (graph edit distance) αποτελεί υπολογιστικά δύσκολο πρόβλημα (NP-hard), περιορίζοντας σημαντικά τη χρηστικότητά του σε μεγάλα δεδομένα.

Για να αντιμετωπιστεί αυτό το ζήτημα, η πρώτη μέθοδος που παρουσιάζεται *"ενσωματώνει"* (roll up) καθεμιά από τις ακμές στον κόμβο που την περιέχει. Με άλλα λόγια, κάθε κόμβος εμπλουτίζεται με έννοιες της μορφής $\exists r.C$, όπου r είναι μια σχέση (π.χ. "riding", "on") και C μια έννοια (π.χ. Fish). Έτσι, αν ένας κόμβος έχει την έννοια Cat και μια εξερχόμενη ακμή $r(\mathsf{Cat}, \mathsf{Fish})$, τότε στο σετ του κόμβου προστίθεται η νέα έννοια $\exists \mathsf{r}.\mathsf{Fish}$. Έτσι, η δομή του γραφήματος μετατρέπεται σε ένα *σύνολο συνόλων* εννοιών, καθιστώντας το πρόβλημα αναζήτησης αντιπαραδειγμάτων ένα ζήτημα *set-edit-distance* αντί για το πλήρες πρόβλημα graph-edit-distance.

Μέσω αυτής της προσέγγισης, οι αλγόριθμοι για την ελαχιστοποίηση των αλλαγών ανάμεσα σε δύο περιγραφές (set-edit-distance) μπορούν να εφαρμοστούν αποτελεσματικά, αξιοποιώντας τεχνικές όπως bipartite matching. Παρόλο που αυτή η μέθοδος ενδέχεται να παραβλέπει πολυπλοκότερες σχέσεις πολλών βαθμίδων, μιας και *περιλαμβάνει* αποκλειστικά την πληροφορία που σχετίζεται με τους γειτονικούς κόμβους.

1.2.2 Χρήση Graph Neural Networks (GNNs)

Παρότι η ενσωμάτωση των ακμών μειώνει δραστικά την πολυπλοκότητα, δεν παύει να αγνοεί περαιτέρω διασυνδέσεις (multi-hop) οι οποίες μπορεί να είναι κρίσιμες για την ερμηνεία. Εδώ εισέρχεται η δεύτερη προσέγγιση, που αξιοποιεί **Graph Neural Networks (GNNs)** ώστε να διατηρήσει τη *πλήρη* δομή του γραφήματος. Συγκεκριμένα, εκπαιδεύεται ένα *Σιαμαίο* (Siamese) GNN που χαρτογραφεί κάθε γράφημα σε ένα ενιαίο *embedding space*, ανάλογα με την ομοιότητά του με τα υπόλοιπα γραφήματα. Με αυτόν τον τρόπο, αντί να γίνεται εξαντλητικός υπολογισμός της απόστασης (graph-edit-distance) με όλα τα γραφήματα του dataset, αρκεί να εντοπιστεί το πιο κοντινό embedding από άλλη κλάση, και κατόπιν να γίνει προσέγγιση του κόστους τροποποίησης μόνο για το συγκεκριμένο ζεύγος.

Μέσω του GNN, διατηρούνται τόσο οι κόμβοι όσο και οι ακμές *σε όλο το εύρος* του γραφήματος, επιτρέποντας την ανάλυση πιο σύνθετων δομών. Για παράδειγμα, ενδέχεται να είναι κρίσιμο να διατηρηθεί η πληροφορία βαθύτερών συσχετίσεων μεταξύ αντικειμένων για την καλύτερη παραγωγή αντιπαραδειγμάτων σε ένα πρόβλημα. Η πληροφορία αυτή με τον αλγόριθμο ενσωμάτωσης των ακμών στους κόμβους παραλείπεται εντελώς. Η αποτελεσματικότητα της μεθόδου αυτής παρουσιάζεται εκτενώς τόσο θεωρητικά όσο και πρακτικά.

1.2.3 Συμπερασματα/Παρατηρήσεις

Οι δύο μέθοδοι—**η ενσωμάτωση των ακμών στους κόμβους** και **η προσέγγιση με GNN**—επιβεβαιώνουν ότι η συμπερίληψη των ρόλων και των σχέσεων ανάμεσα στις έννοιες βελτιώνει σημαντικά την ερμηνευσιμότητα. Με την ενσωμάτωση των ακμών διατηρείται ένα τμήμα της πληροφορίας που τις αφορά, χωρίς να απαιτείται πλήρης υπολογισμός του graph edit distance. Αντίθετα, το GNN προσφέρει τη διατήρηση της πλήρους γραφικής δομής, με αντίτιμο αυξημένη υπολογιστική πολυπλοκότητα, αλλά και *ουσιωδώς* πληρέστερη αντίληψη των σχέσεων.

Η επιλογή της μεθόδου εξαρτάται από τις απαιτήσεις της εκάστοτε εφαρμογής: πόσο μεγάλα είναι τα δεδομένα, πόσο περίπλοκες είναι οι σχέσεις που συμμετέχουν (πολλαπλών βαθμίδων, πολλοί κόμβοι κ.λπ.) και ποια ακρίβεια ερμηνείας είναι επιθυμητή. Όπως καταδεικνύεται σε ερευνητικές και πρακτικές εφαρμογές, όπου φαίνεται ότι η ρητή ενσωμάτωση της πληροφορίας των ακμών αποκαλύπτει προκαταλήψεις που ειδάλλως θα έμεναν κρυφές και ενισχύει την εμπιστοσύνη στις αποφάσεις του μοντέλου ΤΝ.

1.3 Παραγωγή επεξηγήσεων μέσω αντιπαραδειγμάτων σε δεδομένα κειμένων

Το κεφάλαιο αυτό επεκτείνει τις μεθόδους παραγωγής επεξηγήσεων μέσω αντιπαραδειγμάτων για περιπτώσεις *κειμένων*, βασιζόμενο στις αρχές και τεχνικές που παρουσιάστηκαν σε προηγούμενα κεφάλαια (ιδίως στο Κεφ. 4). Ενώ στα προηγούμενα κεφάλαια εξετάστηκαν κυριως συστήματα που δέχονται σαν είσοδο εικόνες και ηχητικά δεδομενα, η παρουσία ταξινομητών κειμένου (π.χ. για ανάλυση συναισθήματος ή ταξινόμηση θεματολογίας) καθιστά απαραίτητη την ανάπτυξη αποδοτικών και ελεγχόμενων αντιπαραδειγμάτων στο πεδίο της Επεξεργασίας Φυσικής Γλώσσας (ΕΦΓ). Σε αντίθεση με τις εικόνες/ήχο, όπου ένα pixel/ένα δείγμα ηχητικού σήματος διατηρεί σταθερή σημασία, τα κειμενικά δεδομένα εμφανίζουν έντονη *εξάρτηση από τα συμφραζόμενα*. Στο πλαίσιο αυτό, το κεφάλαιο προτείνει μια καινοτόμο προσέγγιση που αξιοποιεί *διμερείς γράφους* (bipartite graphs) και *Graph Neural Networks (GNNs)* για να παραχθούν *συνεκτικές*, *ελάχιστες* και *αποτελεσματικές* αντικαταστάσεις λέξεων.

1.3.1 Κίνητρο

Σε αντίθεση με εικόνες/ήχο, οι λέξεις αποκτούν νόημα μόνο μέσα από τα συμφραζόμενά τους. Ενώ νευρωνικά μοντέλα γενικού σκοπού (π.χ. μεγάλα γλωσσικά μοντέλα - Large Language Models (LLMs)) μπορούν να παράγουν γραμματικά επιτυχημένες διορθώσεις σε ένα κείμενο, συχνά στερούνται *διαφάνειας* και *ακριβούς ελέγχου* στον αριθμό και το είδος των αλλαγών. Για αυτό ο στόχος του σύστηματων παραγωγής αντιπαραδειγμάτων του συγκεκριμένου κεφαλαίου είναι ο παρακάτω!

- **Ελαχιστοποίηση τροποποιήσεων:** Αλλαγές σε όσο το δυνατόν λιγότερες λέξεις.
- **Σημασιολογική εγγύτητα:** Η καινούργια εκδοχή πρέπει να διατηρεί τον βασικό πυρήνα νοήματος.
- **Γλωσσική "ροή":** Η τελική πρόταση να παραμένει ευανάγνωστη και φυσική.
- **Κλιμάκωση/Αποδοτικότητα:** Ο χρόνος που χρειάζεται για να δημιουργηθούν αυτά τα αντίπαραδείγματα πρέπει να είναι διαχειρίσιμος ακόμη και για μεγάλα σώματα κειμένου.

Με βάση αυτά, το προτεινόμενο πλαίσιο μοντελοποιεί τις αλλαγές λέξεων ως συνδυαστικό πρόβλημα βέλτιστης αντιστοίχισης πάνω σε *διμερείς γράφους*, εξασφαλίζοντας συγχρόνως αποδοτικότητα μέσω της χρήσης ενός *GNN* που προσεγγίζει το βέλτιστο αποτέλεσμα.

1.3.2 Μοντελοποίση της φυσικής γλώσσας ως διμερή γράφο

Η βασική ιδέα βασίζεται στη σύνδεση λέξεων του εισαγωγικού κειμένου (κόμβοι "πηγής", S) με κόμβους-στόχους (T) που αποτελούν πιθανούς *υποψήφιους αντικαταστάτες*. Η βαρύτητα των ακμών (*edge weight*) αποτυπώνει πόσο "ακριβή"' θεωρείται η αντικατάσταση μιας λέξης. Έτσι, η επίλυση του προβλήματος μετατρέπεται σε *πρόβλημα αντιστοίχισης* (Linear Assignment), όπου επιχειρείται η ελάχιστη άθροιση βαρών στις ακμές που επιλέγονται.

Για τον υπολογισμό των βαρών υιοθετούνται δύο προσεγγίσεις:

1. **Χρήση WordNet:** Κάθε ζεύγος λέξεων αντιστοιχεί σε μια διαδρομή στο λεξικό WordNet. Αυτό επιτρέπει *απόλυτη ερμηνευσιμότητα* (transparency) αλλά περιορίζεται από τη σχετικά πεπερασμένη γνώση και αυστηρή ιεραρχία του WordNet.
2. **Χρήση Ενσωματώσεων (Embeddings):** Εδώ, η "απόσταση" ορίζεται από τη συσχέτιση π.χ. συνημίτονου (cosine) σε προεκπαιδευμένα διανύσματα λέξεων. Οι ενσωματώσεις (π.χ. AnglE, GinaAI) τείνουν να αποτυπώνουν πιο σφαιρικά τη σημασιολογική εγγύτητα, αλλά στερούνται της ερμηνευσιμότητας που παρέχει το WordNet.

Σε αυτό το στάδιο, τίθεται ο περιορισμός ότι κάθε λέξη (στο S) μπορεί να αντιστοιχηθεί *τουλάχιστον* σε μία πιθανή αντικατάσταση (στο T). Η αντιστοίχιση μπορεί να είναι ελαφρώς *χαλαρή* (Relaxed Linear Assignment Problem, RLAP), διευκολύνοντας την εύρεση κατάλληλων αλλοιώσεων σε κειμενικά δεδομένα.

1.3.3 Ακριβής Επίλυση εναντίον Προσέγγιση με Νευρωνικά Δίκτυα Γράφων

Αρχικώς, το πρόβλημα του *Linear Assignment* (ή Hungarian algorithm) προσφέρει μια *ακριβή* λύση στο πρόβλημα που μελετάται, όμως η πολυπλοκότητα ανέρχεται σε $O(mn \log n)$, και πράγμα που το κάνει μη εφαρμόσιμο σε περιπτώσεις όπου το σύνολο δεδομένων είναι μεγάλο. Αντ' αυτού, το κεφάλαιο προτείνει την εκπαίδευση **GNNs** για την προσέγγιση της βέλτιστης λύσης του RLAP. Εκπαιδεύοντας το GNN σε συνθετικά παραδείγματα γραφημάτων (όπου γνωρίζουμε εκ των προτέρων την ελάχιστη αντιστοίχιση), το μοντέλο μαθαίνει να επιλέγει ακμές (αντικαταστάσεις) κοντά στη βέλτιστη λύση, παρακάμπτοντας το υψηλό υπολογιστικό κόστος, ανεξαρτήτως το μέγεθος του συνόλου.

1.3.4 Διαδικασία Δημιουργίας Αντιπαραδειγμάτων

Η μέθοδος οργανώνεται σε τρία στάδια. Αρχικά, δημιουργείται ο διμερής γράφος στον οποίο ορίζονται *ποιες λέξεις* υφίστανται ενδεχόμενη αλλαγή (κόμβοι S) και *με ποιες* λέξεις μπορούν να αντικατασταθούν (κόμβοι T), με τα βάρη των ακμών (word similarity/embedding distance). Έπειτα, εφαρμόζεται είτε ο κλασικός αλγόριθμος RLAP είτε το *GNN* για να εντοπίσει ένα σύνολο πιθανών $s \to t$ αντιστοιχίσεων. Τέλος, για την εύρεση του βέλτιστου κειμένου χρησιμοποιείται και ο **beam search**, ο οποίος διασφαλίζει την παραγωγή κειμένων με τις *ελάχιστες αλλαγές* (π.χ. επιβολή ορίου αλλαγών π.χ. 10 λέξεις *max*, ή έως 20% του κειμένου) και τερματισμό μόλις αλλάξει ετικέτα ή εξαντληθεί το όριο αλλαγών.

1.3.5 Πειραματική Αξιολόγηση και Αποτελέσματα

Η μεθοδολογία δοκιμάστηκε στα Αγγλικά σε δύο ταξινομητές: **IMDB** (συναισθηματική ανάλυση) και **20 Newsgroups** (ταξινόμηση θεματολογίας). Γίνεται σύγκριση της προτεινόμενης μεθόδου έναντι των *MiCE* (white-box editor) και *Polyjuice* (ένας πιο γενικός editor με χρήση LLM). Η αξιολόγηση περιλαμβάνει:

- *Flip-rate*: Πόσο συχνά το μοντέλο ταξινόμησης αλλάζει ετικέτα.
- *Minimality*: Ποσοστό λέξεων που αντικαθίστανται.
- *Semantic closeness*: Π.χ. μέσω BERTscore.
- *Fluency*: Απόκλιση από τη ροή του πρωτοτύπου (T5-BASE απώλεια).
- *Χρόνος*: Συνολική διάρκεια εκτέλεσης.

Τα αποτελέσματα δείχνουν ότι η προτεινόμενη τεχνική (με τη χρήση deterministic ή GNN RLAP) επιτυγχάνουν *καλύτερη ελαχιστοποίηση*, *καλύτερη ροή*, *συγκρίσιμη ή υψηλότερη μεταβολή ετικέτας* και, κυρίως, *δραστικά μικρότερο χρόνο* σε σχέση με MiCE και Polyjuice. Ο MiCE συχνά επιτυγχάνει οριακά υψηλότερο flip-rate, όμως απαιτεί περισσότερες αλλαγές (παραβλέποντας την ελαχιστοποίηση) και 20πλάσιο χρόνο υπολογισμού. Με την αξιοποίηση **ενσωματώσεων** (αντί για WordNet), επιτυγχάνονται ακόμη πιο μικρές αλλαγές και πιο φυσικός λόγος, βέβαια εις βάρος της πλήρους διαφάνειας που προσφέρει το WordNet.

1.3.6 Ανάλυση Επίδρασης Συνιστωσών

Η προτεινόμενη μεθοδολογία αναδεικνύει αρκετές συνιστώσες τα οποία χρήζουν ανάλυσης και η επιλογή τους μπορεί να επιφέρει σημαντικές αλλαγές στην απόδοση των αλγορίθμων:

- **Έλεγχος εναντίον Ελαχιστότητα:** Η επιβολή χαμηλών ορίων αλλαγών συχνά μειώνει το flip-rate αλλά διασφαλίζει πιο "μικρές" επεμβάσεις.
- **Βελτιστότητα εναντίον Ταχύτητα:** Ακριβείς αλγόριθμοι assignment (π.χ. Hungarian) κλιμακώνονται δυσκολότερα σε σύγκριση με το GNN, το οποίο όμως υστερεί στην εύρεση της βέλτιστης λύσης.
- **Επεξηγησιμότητα εναντίον Απόδοσης:** Λύσεις με WordNet επιτρέπουν ολική διαφάνεια (ελέγχεται η διαδρομή $s \to t$), ενώ οι ενσωματώσεις δίνουν καλύτερες, πιο ευέλικτες αλλαγές με λιγότερες λέξεις.

1.3.7 Συμπεράσματα και Μελλοντικές Προεκτάσεις

Συνδυάζοντας τη λογική των δημερών γράφων, των τεχνικών Linear Assignment, και την προσέγγιση GNN για επιτάχυνση, σε αυτό το κεφάλαιο προτείνεται μια καινοτόμος μέθοδο δημιουργίας αντιπαραδειγμάτων κειμένου, επαρκή τόσο για το “flip” της ετικέτας σε black-box μοντέλα όσο και ως εργαλείο γενικής επεξεργασίας κειμένου. Η μέθοδος επιτυγχάνει ελάχιστες αλλαγές, διατηρεί κοντινή σημασιολογική απόσταση, και **επιτυγχάνει αποδοτική εκτέλεση**, ξεπερνώντας σε ταχύτητα ανταγωνιστικές μεθόδους. Βραχυπρόθεσμα, είναι εφικτό να ενσωματωθούν περισσότερες εξωτερικές πηγές (π.χ. ConceptNet) ώστε να διευρυνθεί το ρεπερτόριο υποψήφιων λέξεων, ενώ η περαιτέρω βελτίωση των GNN (ώστε να πλησιάζει περισσότερο την ιδανική λύση RLAP) αποτελεί επίσης μια ενδιαφέρουσα οδό. Μια ακόμη υποσχόμενη κατεύθυνση αφορά τον υβριδικό συνδυασμό γλωσσικών μοντέλων παραγωγής κειμένου με την προσέγγιση των δημερών γράφων, για ακόμη υψηλότερη ποιότητα επεξεργασίας και *ισορροπία* μεταξύ φυσικότητας και ελέγχου.

1.4 Αξιολόγηση της ποιότητας των επεξηγήσεων μέσω αντιπαραδειγμάτων

Το παρόν κεφάλαιο επικεντρώνεται στην *ανάλυση* και *αξιολόγηση* της ποιότητας των συστημάτων παραγωγής επεξηγήσεων μέσω αντιπαραδειγμάτων (editors). Επίσης, υπογραμμίζεται η ανάγκη ύπαρξης μεθοδικών μετρικών για την εκτίμηση του κατά πόσο οι editors μπορούν να εξάγουν τις ελάχιστες επεξηγήσεις. Οι αντιπαραδειγματικές εξηγήσεις—παρεμβάσεις που αντιστρέφουν την αρχική πρόβλεψη ενός μοντέλου με τις ελάχιστες αλλαγές—αναδεικνύονται ως κεντρικό εργαλείο επεξηγησιμότητας σε μοντέλα μηχανικής μάθησης, καθώς επιτρέπουν σε χρήστες και ειδικούς να αντιληφθούν *ποιο στοιχείο* του εισαγωγικού δείγματος ευθύνεται για την προβλεπόμενη κλάση. Ωστόσο, η έλλειψη των βέλτιστων λύσεων (gold standard) για τέτοιου είδους επεξηγήσεις καθιστά την αξιολόγηση περίπλοκη: πώς μπορεί κάποιος να κρίνει ότι μια προτεινόμενη επεξήγηση αποτελεί πράγματι την *καλύτερη* λύση, όταν δεν υφίσταται συγκεκριμένη βέλτιστη αναφορά, οπότε δεν δύναται η δυνατότητα άμεσης σύγκρισης μαζί του;

Προκειμένου να αντιμετωπιστεί αυτό το κενό, στο κεφάλαιο αυτό προτείνεται κεφάλαιο μια **επαναληπτική προσέγγιση τύπου feedback loop**, μέσω της οποίας δύνεται η δυνατότητα εξαγωγής συμπερασμάτων που αφορούν την βελτιστότητα του editor. Σε αυτή την επαναληπτική διαδικασία η παραγόμενη έξοδος ενός editor f για μια είσοδο x, η όποια συμβολίζεται ως $f(x)$, τροφοδοτείται εκ νέου στον ίδιο editor f, παράγοντας μια δεύτερη εκδοχή $f(f(x))$. Η παραπάνω διαδικασία επαναλαμβάνεται αναδρωμικά για πολλαπλά βήματα. Η ιδέα αυτή, εμπνευσμένη από την back-translation τεχνική για την αξιολόγηση της μετάφραση κειμένου, επιτρέπει τη μελέτη του κατά πόσο ο editor εξακολουθεί να τηρεί την αρχή της *ελαχιστοποίησης* ή παρουσιάζει ασυνέπειες σε διαδοχικά στάδια. Για αυτό προτείνεται και μια νεα μετρική για την αυτόματη αξιολόγηση τέτοιων φαινομένων η όποια ονομάζεται ασυνέπεια και υπολογίζεται σε διάφορα βήματα της παραπάνω αναδρομικής διαδικασίας - `inc@n`, εκτιμά την *αυξημένη απόσταση* που προκύπτει από διαδοχικά edits και εντοπίζει περίπτωση ύπαρξης μονοπατιών που ο editor παρέβλεψε ενώ δεν θα έπρεπε.

1.4.1 Γιατί η Αξιολόγηση των Αντιπαραδειγμάτων είναι Δύσκολη

Το κεντρικό ζήτημα στην αξιολόγηση αντιπαραδειγματικών εξηγήσεων συνίσταται στο ότι δεν υπάρχει ένα σαφές "σωστό" αντιπαράδειγμα προς σύγκριση. Αντ΄ αυτού χρησιμοποιούνται διάφορες μετρικές, άλλοτε ανεξάρτητες του πεδίου εφαρμογής (flip rate, minimality/proximity, sparsity, coverage, feasibility, actionability) και άλλοτε εξειδικευμένες στην περίπτωση επεξεργασίας κειμένου (π.χ. *Levenshtein* για minimality, perplexity για *fluency*, κ.λπ.). Ωστόσο, η μέτρηση μόνο από μια οπτική δεν επαρκεί για να τεκμηριώσει ότι ένας editor βρίσκει μια όντως *βέλτιστη* λύση ούτε για να αναδείξει κρυφούς περιορισμούς (π.χ. το φαινόμενο της καθολικής αλλαγής του κειμένου).

1.4.2 Μια Επαναληπτική Προσέγγιση

Ορίζουμε το πρόβλημά μας ως εξής: Έχουμε έναν ταξινομητή g με $g : \mathcal{L} \to [0,1]^C$, όπου $\mathcal{L}$ είναι το σύνολο κειμένων μιας συγκεκριμένης γλώσσας και C ο αριθμός των κλάσεων. Θεωρούμε τους counterfactual editors ως συναρτήσεις $f : \mathcal{L} \to \mathcal{L}$, με σκοπό:

1. Το τροποποιημένο κείμενο να ταξινομείται σε διαφορετική κλάση: $\arg\max g(f(x)) \neq \arg\max g(x)$.
2. Οι αλλαγές να είναι ελάχιστες βάσει κάποιας μετρικής απόστασης d:

$$f = \arg\min_{h \in \mathcal{F}} d(x, h(x))$$

   όπου $\mathcal{F}$ είναι το σύνολο των συναρτήσεων για τις οποίες $\arg\max g(f(x)) \neq \arg\max g(x)$.

3. Το τροποποιημένο κείμενο $f(x)$ να είναι ευανάγνωστο και εντός της κατανομής της γλώσσας $\mathcal{L}$.

Για την αξιολόγηση της συμμόρφωσης με τα παραπάνω κριτήρια, αναλύεται η συμπεριφορά των editors υπό συνθήκες επαναληπτικής ανάδρασης, εξετάζοντας τη συνάρτηση $f(f(\ldots f(x)))$ για n επαναλήψεις. Καθορίζεται ένας νέος μετρικός δείκτης για την ποσοτικοποίηση του δεύτερου κριτηρίου μέσω της επαναληπτικής διαδικασίας, ενώ τα πρώτα και τρίτα κριτήρια ελέγχονται μέσω μετρικών απόδοσης στο n-οστό βήμα ανάδρασης, δηλαδή metric@n. Η ανάλυση επικεντρώνεται στις τιμές των μετρικών μετά από n εφαρμογές του f, ενώ οι επόμενες ενότητες εξειδικεύουν τις μετρικές αξιολόγησης και τις σχετικές υποθέσεις.

Συγκεκριμένα για την αυτόματη αξιολόγηση της βελτιστότητας της παραγωγής αντιπαραδειγμάτων, εισάγεται μια μέθοδος που ξεπερνά την απλή χρήση μετρικών για ένα βήμα της εξόδου ενός editor. Αντί να μελετάται μόνο ένας μετασχηματισμός $x \mapsto f(x)$, η ιδέα έγκειται στην ανατροφοδότηση του κείμενου *f(x)* στον ίδιο editor, ώστε να προκύψει $f(f(x))$, και ούτω καθεξής έως n φορές. Η νεοεισαγόμενη μετρική inc@n (inconsistency) μετρά πόσο μεγαλώνει η απόσταση (π.χ. μέσω *Levenshtein*) μεταξύ διαδοχικών βημάτων. Αν ο editor f ήταν πραγματικά "ελάχιστος" στη συμπεριφορά του, τότε δε θα έπρεπε σε επόμενο βήμα να εκτελεί αλλαγές με μεγαλύτερο "κόστος" (π.χ. 10 αλλαγές) από αυτό που χρησιμοποιήθηκε αρχικά (π.χ. 8 αλλαγές). Έτσι, αν τελικά υφίσταται κάποια ανώτερη τιμή, συνάγεται ότι υπήρχε ένα καλύτερο μονοπάτι (με μικρότερη απόσταση) που ο editor αγνόησε, άρα αυτός παρουσιάζει **ασυνέπειες** (inconsistencies) σχετικά με τη βελτιστότητά του. Με βάση τη παραπάνω ανάλυση, ορίστηκε ένα πλαίσιο μετρικων για την αυτόματη αξιολόγηση της βελτιστότητας των αντιπαραδειγμάτων η οποία ονομάζεται **ασυνέπεια**. Η ασυνέπεια μπορεί να οριστεί με βάση διάφορες μετρικές, άλλα στο κεφάλαια αυτό αναλύεται η πιο βασική και ευρέως χρησιμοποιούμενη μετρική της ελαχιστότητας η όποια πρακτικά μετράει την απόσταση (σε αριθμό λέξεων ή χαρακτήρων) μεταξύ δυο δειγμάτων d. Η μετρική αυτή που εξετάστηκε ονομάζεται **ασυνέπεια ελαχιστότητας** και δίνεται από τον παρακάτω μαθηματικό τύπο.

$$\mathsf{inc@}n(f,x) = \frac{1}{n}\sum_{i=0}^{n-1} \mathsf{inc}(f_{i+1}(x), f_i(x)), \tag{1.4.1}$$

όπου $f_0(x) = x$ και $f_i(x) = f(f_{i-1}(x))$.

1.4.3 Πειράματα

Το κεφάλαιο επιβεβαιώνει τις ιδέες του δοκιμάζοντας τρεις γνωστούς editors: **MiCE, Polyjuice** και **TextFooler**, πάνω σε δύο συνήθη σύνολα δεδομένων: IMDb (ταξινόμηση κειμένου για συναίσθημα) και Newsgroups (20 κλάσεις θεματολογίας). Οι editors αυτοί αντιπροσωπεύουν διαφορετικές πρακτικές:

- *MiCE*, επιλέγει τοποθεσίες στο κείμενο και τις συμπληρώνει με ένα γλωσσικό μοντέλο, και συχνά απαιτεί white-box πρόσβαση στον ταξινομητή.
- *Polyjuice*, προτείνει αλλαγές με βάση ένα μεγάλο γλωσσικό μοντέλο εκπαιδευμένο σε διάφορα σύνολα δεδομένων ώστε να καλύψει όσο μεγαλύτερο φάσμα εφορμογών είναι δυνατόν. Για αυτό ο editor αυτός θεωρείται γενικού-σκοπού, μιας και δεν εξαρτάται από την έξοδο ή την task ενός συγκεκριμένου ταξινομητή.
- *TextFooler*, επιδιώκει να εντοπίσει λέξεις κλειδια σε ένα κείμενο και τις αντικαθιστά με συνώνυμα. Όπως και ο Polyjuice λειτουργεί χωρίς να απαιτεί πρόσβαση στην εσωτερική δομή του ταξινομητή, αλλά σε αντίθεση με το Polyjuice τον χρειάζεται σαν μαύρο κουτί για την παραγωγή (black-box adversarial).

Η έξοδος ενός editor για ένα μόνο βήμα (π.χ. @1 edit), πολλές φορές μπορεί να οδηγήσει στην εξαγωγή λανθασμένων συμπερασμάτων τα οποία αφορούν την λειουργία του. Για παράδειγμα ορισμένοι editors στο πρώτο βήμα φαίνεται να έχουν υψηλό flip rate ή μικρό minimality. Ωστόσο, με την εισαγωγή του feedback loop—δηλαδή με την εισαγωγή της εξόδου πίσω στην είσοδο του—παρατηρούνται διάφορα ενδιαφέροντα φαινόμενα που μπορεί να επηρεάζουν σημαντικά τα αποτελέσματά του όταν αυτός εφαρμοστεί σε πραγματικές συνθήκες.

1.4.4 Κύριες Διαπιστώσεις και Εφαρμογές

Τα ευρήματα υποδηλώνουν ότι μια αξιολόγηση χωρίς το feedback loop των editors δεν αρκεί για να αποκαλυφθούν *κρυφές* αδυναμίες ή πλεονεκτήματα. Για παράδειγμα ο editor να φαίνεται πολύ αποτελεσματικός στο να βρίσκει δείγματα που ανήκουν σε άλλη κλάση και με λίγες αλλαγές, αλλά να μην είναι καθόλου ανεκτικός σε δεδομένα εκτός κατανομής. Επιπλέον, αναδεικνύεται ότι:

- Η στόχευση συγκεκριμένης κλάσης (π.χ. MiCE) ενδέχεται να δυσκολεύει περισσότερο την παραγωγή διαδοχικών έγκυρων αντιπαραδειγμάτων σε περιβάλλον με πολλές κλάσεις στόχους (π.χ. στο σύνολο δεδομένων Newsgroups).
- Οι editors που στηρίζουν την λειτουργία τους σε δημιουργικά συστήαμτα (π.χ. Polyjuice, MiCE) κάνουν γενικά πιο "ευφυείς" αλλά και πιο *απρόβλεπτες* αλλαγές. Αυτό καταγράφεται και σε μετρικές τύπου

perplexity ή grammatical errors: όπου σε επόμενα βήματα του feedback loop, είτε βελτιώνουν το κείμενο είτε χειροτερεύουν απρόσμενα.

- Το feedback loop (`inc@n` κ.λπ.) καθιστά εφικτή τη "διάγνωση" του αν ο editor χάνει ορισμένες ελάχιστες λύσεις. Μια αυξημένη τιμή inc@n σημαίνει ότι ο editor αναγκάστηκε να κάνει μεγαλύτερο αριθμό αλλαγών από ότι θα έπρεπε, άρα υποδεικνύει περιπτώσεις μη βέλτιστης συμπεριφοράς του.
- Σε μια πραγματική χρήση, λ.χ. επαναλαμβανόμενης διερεύνησης ενός κειμένου, οι editors που αντέχουν σε πολλούς επαναληπτικούς κύκλους (δηλαδή επιτυγχάνουν χαμηλό inc@n) αποδεικνύονται μακροπρόθεσμα προτιμητέοι.

1.4.5 Συμπεράσματα και Παρατηρήσεις

Καταλήγοντας, στο κεφάλαιο αυτό παρουσιάζεται μια νεα τεχνική μαζί με μια μετρική για την αυτόματη αξιολόγηση των συστημάτων παραγωγής αντιπαραδειγματικών (editors). Η μετρική αυτή, η οποία ονομάζεται inconsistency, στηρίζεται στην πραγματοποίηση ενός feedback loop δύνοντας την έξοδο του editor πίσω στην είσοδο του. Πειραματικά φαίνεται ότι το inconsistency μπορεί να εντοπίσει περιπτώσεις στις οποίες ο editor συμπεριφέρεται υπο-βέλτιστα ή ασυνεπώς, αναδεικνύοντας την αναγκαιότητα για βαθύτερη αξιολόγηση. Οι editors που βάσει τυπικών μετρικών μοιάζουν ιδανικοί με χρήση των καθιερωμένων τεχνικών αξιολόγησης μπορεί να αποδειχθούν *ασταθείς* όταν αξιολογούνται με την προτεινόμενη τεχνική, ενώ κάποιοι υπό-εκτιμημένοι σε μία μόνο εκτέλεση εμφανίζονται *συνεπέστεροι* σε επανειλημμένες εφαρμογές.

Η συμβολή της παρούσας ανάλυσης είναι πως επιτρέπει στους χρήστες και ερευνητές να επιλέγουν τον καταλληλότερο editor ανάλογα με τη χρήση, αναλύοντας ποιοτικότερα τα "προφίλ" πλεονεκτημάτων/μειονεκτημάτων. Επίσης, η μεθοδολογία *επανατροφοδότησης* μπορεί να αξιοποιηθεί για την περαιτέρω εκπαίδευση και βελτίωση των ίδιων των editors, εφόσον τα δεδομένα της ανατροφοδότησης μπορούν να χρησιμεύσουν ως δεδομένα για επιπλέον εκπαίδευση του editor Στο άμεσο μέλλον, προβλέπεται η επέκταση των παραπάνω ιδεών και σε άλλα πεδία (π.χ. εικόνες, time-series), καθώς και την εμπλοκή "ανθρώπων" για την αξιολόγηση των αποτελεσμάτων τόσο της προτεινόμενης μετρικής όσο και των ίδιων των editor, ώστε να αναδειχθούν τυχόν συσχετίσεις της μετρικής με την προτίμηση των ανθρώπων.

1.5 Επεξηγήσιμη Μετρική για την Οπτικοποίηση Ιστοριών μέσω Αντιπαραδειγματικών Εξηγήσεων

1.5.1 Εισαγωγή

Τα σύγχρονα παραγωγικά μοντέλα εικόνας, όπως τα Generative Adversarial Networks (GANs), τα μοντέλα διάχυσης (diffusion models) και οι αρχιτεκτονικές που βασίζονται σε μετασχηματιστές (transformers), έχουν επιτύχει εντυπωσιακή πρόοδο στην παραγωγή ρεαλιστικών και υψηλής ποιότητας εικόνων. Παρ' όλα αυτά, η *αξιολόγηση* των αποτελεσμάτων τους παραμένει δύσκολη. Οι δημοφιλείς μετρικές, όπως το Inception Score (IS) και το Fréchet Inception Distance (FID), επικεντρώνονται κατά κύριο λόγο σε χαρακτηριστικά χαμηλού επιπέδου (pixel-level ή στατιστικούς δείκτες), με αποτέλεσμα να παραβλέπουν σημαντικές *εννοιολογικές* πληροφορίες στις παραγόμενες εικόνες. Παράλληλα, η ερμηνευσιμότητα (explainability) στα γεννητικά μοντέλα είναι λιγότερο ανεπτυγμένη σε σύγκριση με τα μοντέλα ταξινόμησης, όπου τεχνικές όπως οι χάρτες προσοχής (saliency maps) και τα τοπικά μοντέλα αντικατάστασης είναι από καιρό καθιερωμένες.

Το κεφάλαιο αυτό επιχειρεί να καλύψει αυτά τα κενά, προτείνοντας ένα *μοντέλο αξιολόγησης που βασίζεται σε έννοιες (concept-based)*, το οποίο είναι *ανεξάρτητο* από τη δομή του εκάστοτε γεννητικού μοντέλου (model-agnostic). Το κλειδί είναι η αποτύπωση *ποια* εννοιολογικά στοιχεία (αντικείμενα, ιδιότητες, σχέσεις) εμπεριέχονται στις παραγόμενες εικόνες, συγκριτικά με τα στοιχεία που *αναμένονταν* βάσει της προτροπής (prompt) ή των δεδομένων αληθείας (ground truth). Η σύγκριση γίνεται με έναν σαφή μηχανισμό εμπνευσμένο από την μέθοδος παραγωγής *αντιπαραδειγμάτων (counterfactual explanations)* που έχει παρουσιαστεί. Ο μηχανισμός αυτός λειτουργεί προσδιορίζοντας τις ελάχιστες εννοιολογικές αλλαγές που χρειάζονται ώστε οι παραγόμενες εικόνες να ευθυγραμμιστούν εννοιολογικά με την εκάστοτε προτροπή.

1.5.2 Εξαγωγή Εννοιών και Σύγκριση

Σε αντίθεση με τις κλασικές αξιολογήσεις που πραγματοποιούνται σε επίπεδο εικονοστοιχείων (pixels), η προτεινόμενη προσέγγιση χαρτογραφεί κάθε παραγόμενη εικόνα σε ένα *σύνολο εννοιών*. Στην περίπτωση που ο χρήστης δίνει κείμενο ως προτροπή, αυτό μετατρέπεται σε ένα *σύνολο-στόχο εννοιών*, το οποίο συμβολίζεται ως T. Στη συνέχεια, ένα μοντέλο ανίχνευσης αντικειμένων ή χαρακτηριστικών εφαρμόζεται στην παραγόμενη εικόνα, παράγοντας μια *σύνολο-πηγή εννοιών*, το οποίο συμβολίζουμε ως S. Οι έννοιες αυτές μπορεί να αφορούν αντικείμενα (π.χ. “σκύλος”, “αυτοκίνητο”) ή ιδιότητες (“μπλε”, “κυκλικό”) ή ακόμα και πιο σύνθετες οντότητες (“μεγάλη μπλε σφαίρα”).

1.5.3 Αντιπαραδείγματα και Συντακτικό των Επεξεργασιών Εννοιών

Για να μετρήσουμε πόσο κοντά ή πόσο μακριά βρίσκεται το S από το T, ορίζουμε ένα *ελάχιστο σύνολο πράξεων επεξεργασίας* που μπορούν να μετασχηματίσουν την σύνολο-πηγή στο σύνολο-στόχο. Οι πράξεις αυτές είναι (ομοίως με το αυτά που παρουσιάστηκαν στο Κεφάλαιο 1.1.3):

- **Εισαγωγή (Insertion, I)**: Εισαγωγή μίας έννοιας στο S, αν αυτή λείπει και υπάρχει στο T.
- **Διαγραφή (Deletion, D)**: Διαγραφή μίας επιπλέον έννοιας από το S, αν δεν υπάρχει στο T.
- **Αντικατάσταση (Replacement, R)**: Αντικατάσταση μίας λανθασμένης έννοιας του S με την σωστή, ώστε να ταιριάξει στο T.

Για καθεμία από αυτές τις ενέργειες ορίζεται ένα *κόστος*, συχνά βασισμένο σε κάποια μετρική απόστασης $d(\cdot,\cdot)$ ανάμεσα σε δύο έννοιες. Για παράδειγμα, η αντικατάσταση της έννοιας “γάτα” με την έννοια “σκύλος” μπορεί να είναι φθηνότερη από την αντικατάσταση “γάτα” με “αυτοκίνητο”, εφόσον οι δύο πρώτες ανήκουν στην ευρύτερη κατηγορία “ζώα”.

Ορίζουμε τη *Συντακτική Απόσταση Συνόλου Εννοιών* (Concept Set Edit Distance, CSED) με τον παρακάτω τρόπο:

$$\mathrm{CSED} = D(S \to T) \; = \; \min \sum_{\substack{s \in S \\ t \in T}} \; \sum_{\mathrm{ops} \in \{I, D, R\}} d(s, t), \tag{1.5.1}$$

όπου το $d(s,t)$ εκφράζει το κόστος της απαραίτητης αλλαγής για να μεταβούμε από την έννοια s στην έννοια t. Στις περιπτώσεις εισαγωγής ή διαγραφής, μπορεί να ληφθεί υπόψη και η απόσταση από μία γενική ρίζα (π.χ. "entity" σε μια βάση γνώσης όπως το WordNet).

1.5.4 Οπτικοποίηση Ιστοριών (Story Visualization)

Στο πρόβλημα της οπτικοποίησης ιστοριών (SV), καλούμαστε να παράγουμε μια *ακολουθία* εικόνων $\{I_k\}_{k=1}^{L}$, καθεμία από τις οποίες αντιστοιχεί σε ένα τμήμα της ιστορίας $\{c_k\}_{k=1}^{L}$. Αναδύονται δύο βασικές μετρικές:

Story Loss (SL). Για την k-οστή εικόνα, ορίζουμε το σύνολο εννοιών S_k (όπως ανιχνεύεται αυτόματα) και το σύνολο-στόχο T_k (από το k-οστό τμήμα της αφήγησης). Ο υπολογισμός της απόστασης επεξεργασίας (CSED) ανάμεσα στα δύο αυτά σύνολα είναι:

$$\text{CSED}_k = D(S_k,\, T_k).$$

Το άθροισμα των CSED σε όλα τα καρέ δίνει τον ορισμό του *Story Loss*:

$$\text{SL} = \sum_{k=1}^{L} \text{CSED}_k.$$

Όσο μεγαλύτερη είναι η τιμή του SL, τόσο πιο ελλειμματική είναι η απόδοση του μοντέλου στην πιστή αναπαράσταση του σεναρίου σε κάθε καρέ.

Consistency Loss (CL). Εκτός από την ορθή απόδοση κάθε καρέ χωριστά, το μοντέλο πρέπει να διατηρεί *συνέπεια* (consistency) σε αντικείμενα ή ιδιότητες που ήδη εμφανίστηκαν. Ορίζουμε τις έννοιες που ανιχνεύονται στο k-οστό καρέ ως S_k. Τότε η μετρική CL υπολογίζει ποινές για ανεπιθύμητες αλλαγές μεταξύ διαδοχικών καρέ:

$$\text{CL} = \sum_{k=2}^{L} D(S_k,\, S_{k-1}),$$

όπου $D(\cdot,\cdot)$ είναι η ίδια λογική απόστασης επεξεργασίας εννοιών. Μεγάλο CL υποδεικνύει ασυνέχειες, π.χ. αφαίρεση ενός αντικειμένου που θα έπρεπε να παραμείνει ή προσθήκη/αλλαγή γνωρισμάτων χωρίς να το υπαγορεύει η αφήγηση.

1.5.5 Παραγωγή Σκηνών (Scene Generation)

Σε μια απλούστερη περίπτωση, ζητούμε από το μοντέλο να παράγει *μία* εικόνα βάσει ενός κειμένου c. Από το κείμενο εξάγεται το σύνολο εννοιών-στόχος T, ενώ από την παραγόμενη εικόνα ένα σύνολο-πηγή S. Ο τελικός στόχος είναι η CSED, δηλαδή η απόσταση $D(S \to T)$. Η διαφορά ανάμεσα στα δύο σύνολα φανερώνει ποιες εισαγωγές, διαγραφές ή αντικαταστάσεις χρειάζονται ώστε να ταιριάξει η σκηνή στο ζητούμενο περιεχόμενο.

1.5.6 Τοπικές Επεξηγήσεις (Local Explanations)

Κάθε εικόνα που παράγεται (ή καρέ σε μια ιστορία) δίνει ένα ελάχιστο μονοπάτι μετασχηματισμού από S σε T. Για παράδειγμα, "αντικατάστησε το 'rubber' με 'metallic' " ή "διάγραψε το 'car"' εάν η ιστορία δεν ανέφερε κανένα αυτοκίνητο. Το συνολικό *κόστος* αποτυπώνει την ποιότητα της παραγόμενης εικόνας ως προς την πιστότητα στην προτροπή/κείμενο. Συνεπώς, η τοπική επεξήγηση αναδεικνύει με ακρίβεια *που* ακριβώς αποτυγχάνει το μοντέλο και *πώς* μπορεί να διορθωθεί.

1.5.7 Γενικές Επεξηγήσεις (Global Explanations)

Η ανάλυση όλων των τοπικών επεξηγήσεων σε ένα σύνολο δεδομένων επιτρέπει την εξαγωγή ευρέων *κανόνων*. Για παράδειγμα, με κανόνες συσχέτισης (π.χ. αλγόριθμος Apriori), μπορούμε να δούμε ότι το μοντέλο *συχνά* μπερδεύει κάποιο σχήμα (π.χ. "cylinder" αντί για "sphere") ή *αποτυγχάνει* συστηματικά σε μια κατηγορία όπως το "rubber" vs "metallic". Αυτή η γενικευμένη εξήγηση βοηθάει στο να εντοπίσουμε πιθανές προκαταλήψεις ή "τυφλά σημεία" του παραγωγικού μοντέλου.

1.5.8 Πειραματικά Αποτελέσματα

Στην παρούσα ενότητα περιγράφουμε συνοπτικά τα βασικά πειράματα που πραγματοποιήθηκαν, τα σύνολα δεδομένων που επιλέχθηκαν, τα μοντέλα παραγωγής εικόνων που χρησιμοποιήθηκαν και ορισμένα από τα κυριότερα εμπειρικά αποτελέσματα. Καταρχάς, για τη διερεύνηση της απόδοσης στη μετατροπή κειμένων σε αλληλουχίες εικόνων (*Story Visualization*), χρησιμοποιήθηκε μια παραλλαγή του γνωστού συνόλου δεδομένων CLEVR (ονομάζεται CLEVR-SV). Στη μορφή αυτή, το CLEVR περιλαμβάνει διαδοχικά καρέ (frames), όπου κάθε καρέ αντιστοιχεί σε μια αφήγηση-πρόταση και προστίθενται σταδιακά νέα αντικείμενα με συγκεκριμένα χαρακτηριστικά (π.χ. χρώμα, σχήμα, μέγεθος, υλικό). Κάθε πείραμα στο CLEVR-SV διερευνά το κατά πόσο ένα μοντέλο μπορεί να αποτυπώσει σωστά όχι μόνο τις έννοιες ενός μεμονωμένου καρέ αλλά και τη συνέχειά τους ανάμεσα σε διαφορετικά καρέ.

Για να αξιολογηθεί η αποδοτικότητα των μοντέλων στη δημιουργία μεμονωμένων εικόνων από περιγραφικές λεζάντες (*Scene Generation*), έγινε χρήση ενός μεγάλου συνόλου δεδομένων που προέρχεται από το MS-COCO. Στο COCO, κάθε εικόνα συνοδεύεται από διάφορες κειμενικές περιγραφές (captions), που αναφέρουν τα αντικείμενα, τη δράση ή το πλαίσιο της σκηνής. Στην πράξη, για τα πειράματά μας, εστιάσαμε στην περίπτωση όπου δίνεται μια περιγραφή και καλείται το μοντέλο να παράγει μια εικόνα συμβατή με αυτήν. Η περαιτέρω ανάλυση στηρίχθηκε στην εξαγωγή εννοιών από το κείμενο (π.χ. «car», «street», «person») και στον συσχετισμό τους με τις έννοιες που πραγματικά εμφανίζονται στην παραγόμενη εικόνα.

Σε ό,τι αφορά τα παραγωγικά μοντέλα, χρησιμοποιήθηκαν τόσο παραδοσιακές αρχιτεκτονικές GAN όσο και πιο πρόσφατα diffusion models (όπως Stable Diffusion και παραλλαγές Protogen). Τα GAN εκπαιδεύονται χάρη στην αλληλεπίδραση ενός γεννήτορα (generator) και ενός διαχωριστή-κριτή (discriminator), ενώ τα diffusion models βασίζονται σε μια διαδικασία αντίστροφης διάχυσης θορύβου. Στα πειράματα φάνηκε ότι τα diffusion models έχουν τη δυνατότητα να παράγουν πιο ρεαλιστικές και συνεκτικές εικόνες σε πολλές περιπτώσεις, αν και κάθε προσέγγιση παρουσιάζει διαφορετικές αδυναμίες ή μεροληψίες.

Το επόμενο βήμα ήταν η εφαρμογή ανιχνευτών αντικειμένων (object detectors) όπως το YOLO-v8 και το YOLOS στις παραγόμενες εικόνες, ώστε να εντοπιστούν τυχόν παρουσιαζόμενα αντικείμενα ή χαρακτηριστικά. Παράλληλα, από τις περιγραφές-στόχους εξάγονται επίσης λίστες εννοιών. Σε αυτό το πλαίσιο, η αξιολόγηση γίνεται συγκρίνοντας τις δύο λίστες, δηλαδή το «παραγμένο σύνολο εννοιών» με το «στόχο», και υπολογίζοντας πόσες εισαγωγές, διαγραφές ή αντικαταστάσεις χρειάζονται (Concept Set Edit Distance) για να υπάρξει πλήρης ταύτιση.

Τα αποτελέσματα έδειξαν ότι σε απλές περιπτώσεις το μοντέλο μπορεί να είναι αρκετά ακριβές, ειδικά αν η περιγραφή περιλαμβάνει αντικείμενα και χαρακτηριστικά που έχουν εκπροσωπηθεί επαρκώς στην εκπαίδευση. Ωστόσο, σε πιο σύνθετες σκηνές, συχνά εντοπίζεται ανάγκη να «διορθωθούν» πολλές έννοιες. Για παράδειγμα, σε ορισμένα diffusion models παρατηρείται η συστηματική εισαγωγή επιπλέον αντικειμένων (π.χ. περισσότερα άτομα απ' ό,τι περιγράφονται στην πραγματικότητα), ενώ σε άλλες περιπτώσεις οι εικόνες δείχνουν λανθασμένους τύπους αντικειμένων (π.χ. «car» αντί για «bus»). Στο CLEVR-SV, όπου απαιτείται και συνέπεια από καρέ σε καρέ, παρατηρήθηκε ότι μερικές αρχιτεκτονικές GAN ξεχνούν ή αντικαθιστούν προηγούμενα αντικείμενα, με αποτέλεσμα υψηλότερη τιμή Consistency Loss.

Συνολικά, τα πειράματα αναδεικνύουν τα ιδιαίτερα πλεονεκτήματα της μεθόδου που βασίζεται σε ανάλυση εννοιών: δεν μετρά μόνο την οπτική ποιότητα (όπως γίνεται με το FID), αλλά εστιάζει στο αν το σύστημα «κατανόησε» τα ζητούμενα αντικείμενα και τα απεικόνισε σωστά. Παράλληλα, η επεξήγηση των αποτελεσμάτων μέσω των αντιπαραδειγμάτων (π.χ. «αντί να υπάρχει αυτό το αντικείμενο, θα έπρεπε να εμφάνιζε κάτι άλλο») καθιστά εμφανή τα σημεία όπου κάθε μοντέλο δυσκολεύεται σταθερά — ένα στοιχείο ιδιαίτερα χρήσιμο για μελλοντικές βελτιώσεις και αποφυγή επαναλαμβανόμενων λαθών.

1.5.9 Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Το κεφάλαιο παρουσιάζει ένα *μοντέλο αξιολόγησης βασισμένο στις έννοιες*, αξιοποιώντας **αντιπαραδείγματα** (counterfactuals) για να μετρήσει *και* να επεξηγήσει την απόδοση των γεννητικών μοντέλων εικόνας. Προς αντικατάσταση της καθαρά εικονοστοιχειωτής προσέγγισης, η μέθοδος ελέγχει ρητά αν τα παραγόμενα αντικείμενα και οι ιδιότητές τους ανταποκρίνονται στις απαιτήσεις. Επιπλέον, δημιουργεί εύκολα ερμηνεύσιμους δείκτες, τόσο τοπικά ανά δείγμα όσο και γενικευμένα ανά σύνολο δεδομένων, φωτίζοντας «αδύναμα σημεία» ή προκαταλήψεις του μοντέλου.

Μελλοντικά, δυνατότητες βελτίωσης περιλαμβάνουν:

- Χρήση *πλουσιότερων βάσεων γνώσης* για μεγαλύτερη ακρίβεια στην εννοιολογική ιεραρχία και την ανάλυση αποστάσεων.
- Επέκταση στην ανίχνευση *σχέσεων* (π.χ. “αντικείμενο A πίσω από αντικείμενο B”) για ακριβέστερη αξιολόγηση πολυπλοκότερων σκηνών.
- Εφαρμογή των ίδιων αρχών σε ακόμη πιο σύνθετες γεννητικές εργασίες (όπως παραγωγή βίντεο), ώστε να καλύπτεται και η διάσταση του χρόνου.

Συνολικά, η μεθοδολογία αυτή συνιστά ένα βήμα προς πιο *ερμηνεύσιμη*, *ανιχνεύσιμη* και *επεξηγήσιμη* αξιολόγηση των γεννητικών συστημάτων, αναδεικνύοντας όχι μόνο την *ποιότητα* της τελικής εικόνας αλλά και *τους λόγους* για τους οποίους το μοντέλο παρεκκλίνει ή ευθυγραμμίζεται με τις εκάστοτε προδιαγραφές.

1.6 Επεξηγήσιμη Μετρική για την Ανίχνευση Ψευδαισθήσεων στην Αυτόματη Περιγραφή Εικόνων

Η σύγκλιση όρασης υπολογιστών και επεξεργασίας φυσικής γλώσσας (NLP) έχει οδηγήσει στη δημιουργία μοντέλων *Vision-Language (VL)* ικανά να παράγουν λεζάντες (captions) για εικόνες. Παρά τις εντυπωσιακές επιδόσεις τους, τα μοντέλα αυτά πάσχουν συχνά από το φαινόμενο των *"hallucination"* (**παραισθήσεων**), όπου το παραγόμενο κείμενο περιέχει αναφορές σε ανύπαρκτα αντικείμενα ή σε λανθασμένες σχέσεις μεταξύ πραγματικών αντικειμένων. Το φαινόμενο αυτό έχει σοβαρό αντίκτυπο στην αξιοπιστία αυτών των συστημάτων, ειδικά σε ευαίσθητα σενάρια όπως η ιατρική απεικόνιση ή οι βοηθητικές τεχνολογίες για άτομα με προβλήματα όρασης.

Σε αυτό το κεφάλαιο, παρουσιάζουμε ένα *επεξηγήσιμο* πλαίσιο αξιολόγησης των παραισθήσεων στη αυτόματη περιγραφή εικόνων, προσαρμόζοντας τις μεθόδους **παραγωγής αντιπαραδειγμάτων επεξηγήσεων (counterfactual explanations)** που εισήχθησαν στα προηγούμενα κεφάλαια (Κεφ. 4 και 5). Η μεθοδολογία μας εντοπίζει *πώς* και *πού* προκύπτουν οι παραισθήσεις σε ένα κείμενο, και προτείνει τις **ελάχιστες** δυνατές τροποποιήσεις για την αφαίρεση ή διόρθωσή τους. Χάρη στην ιεραρχική γνώση (π.χ. WordNet), διασφαλίζουμε ότι οι διορθώσεις δεν είναι αυθαίρετες, αλλά *σημασιολογικά κοντινές* και λογικά αποδεκτές.

1.6.1 Μεθοδολογία

Οι παραισθήσεις στην έξοδο μοντέλων τεχνητής νοημοσύνης έχουν απασχολήσει κυρίως την κοινότητα του NLP (π.χ. σε Μεγάλα Γλωσσικά Μοντέλα). Ωστόσο, **η μελέτη τους στο πεδίο της πολυτροπικής πληροφορίας** (εικόνα+κείμενο) βρίσκεται ακόμα σε πρώιμο στάδιο. Στα συστήματα περιγραφής εικόνων (image captioning), οι παραισθήσεις εμφανίζονται ως ψευδείς αναφορές σε αντικείμενα που δεν υπάρχουν ή ως ανακρίβειες στις σχέσεις ανάμεσα σε υπαρκτά αντικείμενα.

Τα τρέχοντα μοντέλα παραγωγής λεζάντας, όπως τα *BLIP*, *BLIP-2*, *GiT* κ.ά., πετυχαίνουν υψηλές βαθμολογίες σε παραδοσιακούς γλωσσικούς δείκτες (BLEU, ROUGE, CIDEr), αλλά μπορεί να **παραβλέπουν** τον κρίσιμο άξονα της *πιστότητας προς το οπτικό περιεχόμενο.* Αυτό το κενό δημιουργεί την ανάγκη για μετρικές που εστιάζουν ειδικά στις παραισθήσεις και είναι *επεξηγήσιμες*.

Η βασική ιδέα της προτεινόμενης μεθόδου έγκειται στη χρήση *συνόλων εννοιών* (concept sets) τόσο από το κείμενο όσο και από την εικόνα:

- **Σετ πηγής** S**:** Αντιστοιχεί σε έννοιες (αντικείμενα ή σχέσεις) που εξάγουμε από την παραγόμενη λεζάντα.
- **Σετ στόχου** T**:** Αντιστοιχεί σε έννοιες ή σχέσεις που *όντως* υπάρχουν στη σκηνή της εικόνας, όπως προκύπτουν από δεδομένα εδάφους αλήθειας (annotations).

Για παράδειγμα, αν η λεζάντα αναφέρει *"a dog next to a man"*, τότε $S = \{\text{dog}, \text{man}, (\textit{dog}\text{-next\_to-man})\}$. Εάν η πραγματική εικόνα δείχνει έναν άνδρα με ένα λάπτοπ (laptop) πάνω στα γόνατά του, τότε T θα είναι $\{\text{man}, \text{laptop}, (\textit{laptop}\text{-on-man})\}$.

1.6.2 Βασικές Τροποποιήσεις (Edit Operations)

Ο τροποποιήσεις για τη μετατροπή του S σε T ορίζονται ως εξής (ομοίως με την συλλογιστική που ακολουθήθηκε στο Κεφάλαιο 1.1.3):

- **Αντικατάσταση (Replacement, R):** Αντικατάσταση μιας έννοιας $s \in S$ από μια σωστή έννοια $t \in T$.
- **Διαγραφή (Deletion, D):** Αφαίρεση μιας έννοιας s που δεν έχει αντίστοιχο στην εικόνα (ψευδής αναφορά).
- **Εισαγωγή (Insertion, I):** Προσθήκη μιας έννοιας t που υπάρχει στην εικόνα αλλά λείπει από τη λεζάντα.

Η συνολική *απόσταση εννοιολογικών σετ* (Concept Set Edit Distance, CSED) ορίζεται ως το άθροισμα των *ελαχίστων* κόστους τροποποιήσεων για την πλήρη μετατροπή του S στο T:

$$\text{CSED}(S \to T) \; = \; \min \sum_{\text{edits}} d(s,t), \tag{1.6.1}$$

όπου $d(\cdot,\cdot)$ μια σημασιολογική απόσταση, π.χ. η ελάχιστη διαδρομή σε ιεραρχία όπως το WordNet.

1.6.3 Εντοπισμός Παραισθήσεων σε Αντικείμενα

Έστω ότι η λεζάντα περιλαμβάνει $|S|$ αντικείμενα. Εάν ένα αντικείμενο *δεν* υπάρχει στην εικόνα, πρέπει να διαγραφεί (φαινόμενο *hallucination*). Παράλληλα, εάν ένα αντικείμενο στην πραγματικότητα είναι άλλο (π.χ. “dog” αντί “laptop”), απαιτείται *αντικατάσταση (R)*. Ορίζουμε:

$$\text{Hallucinations}(S,T) \; = \; |\mathbf{D}(S,T)| + |\mathbf{R}(S,T)| + |\mathbf{O}(S,T)|,$$

όπου **O** (Over-specialization) ανταναклά την περίπτωση που το μοντέλο *βλέπει* ένα πιο συγκεκριμένο αντικείμενο από αυτό που υπάρχει (π.χ. λέει “woman” ενώ είναι “girl”), και το T δείχνει άλλο σημασιολογικό κόμβο.

Η αναλογία:

$$\text{HalRate}(S,T) \; = \; \frac{\text{Hallucinations}(S,T)}{|S|}$$

εκφράζει το ποσοστό αντικειμένων στη λεζάντα που είναι παραισθήσεις.

1.6.4 Εντοπισμός Παραισθήσεων σε Σχέσεις (Roles)

Εκτός από τα ίδια τα αντικείμενα, εξετάζουμε *πώς* συσχετίζονται μεταξύ τους στο κείμενο. Παρουσιάζονται σε μορφή τριπλών: $(s_i,\ r,\ s_j)$. Ορίζουμε αντίστοιχα ένα σετ σχέσεων S^r από τη λεζάντα και T^r από τα annotations της εικόνας. Οι ίδιες τροποποιήσεις (**I**, **D**, **R**) εφαρμόζονται, με βασικό μέλημα τώρα την *αλλαγή του ρήματος ή της πρόθεσης* (π.χ. “next to” αντί “on”).

Το *Graph Edit Distance (GED)* επί των S^r, T^r καταγράφει ποιες (και πόσες) σχέσεις είναι *λανθασμένες* (hallucinated) ή *παραλειπόμενες*.

1.6.5 Πειράματα

Χρησιμοποιούμε δεδομένα από το *Microsoft COCO* και από το *Visual Genome (VG)*. Τα αντίστοιχα μοντέλα λεζάντας (BLIP, GiT κ.λπ.) παράγουν λεζάντες για τις εικόνες, ενώ τα ground-truth annotations προσφέρουν “αληθινά” αντικείμενα/σχέσεις. Ακολούθως, συγκρίνουμε το *σετ πηγής* (S, S^r) με το *σετ στόχου* (T, T^r).

Παραδείγματα Παραισθήσεων

Συχνά εμφανίζεται το φαινόμενο να “βλέπει” το μοντέλο επιπλέον αντικείμενα (**Deletion needed**) ή να μπερδεύει τύπους αντικειμένων (**Replacement**). Σε κάποιο παράδειγμα, η λεζάντα μιλάει για “dog next to a man”, ενώ υπάρχει “laptop on man’s lap”. Η ανάλυση υποδεικνύει:

$$\mathbf{R}(\text{“dog”} \;\to\; \text{“laptop”}), \quad \mathbf{R}(\text{“next\_to”} \;\to\; \text{“on”}),$$

ως τις ελάχιστες απαραίτητες τροποποιήσεις.

Ευρήματα

1. **Επίπεδο ψευδών αντικειμένων:** Σε ορισμένες περιπτώσεις, έως και 30% των αντικειμένων μιας λεζάντας προκύπτουν ψευδή.
2. **Σχέσεις-ρόλοι (role hallucinations):** Άνω του 50% των προτάσεων περιέχει λάθος σχέσεις, ιδιαίτερα όταν υπάρχουν περίπλοκες σκηνές.
3. **Ασυμφωνία με γλωσσικούς δείκτες:** Οι καθιερωμένες μετρικές (BLEU, ROUGE) δεν συσχετίζονται απαραίτητα με χαμηλότερα ποσοστά παραισθήσεων. Μπορεί ένα μοντέλο να έχει υψηλό BLEU αλλά να “εφευρίσκει” αντικείμενα.

1.6.6 Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Με την προτεινόμενη μεθοδολογία *αντεπικουρικών επεξηγήσεων*, επιτυγχάνουμε:

- **Εντοπισμό συγκεκριμένων λαθών**: ποια αντικείμενα/ρόλοι δεν συμφωνούν με την πραγματική εικόνα.
- **Πρόταση ελάχιστων διορθώσεων**: ποια έννοια πρέπει να αφαιρεθεί, αντικατασταθεί κτλ.
- **Μοντέλο-αγνωστική προσέγγιση**: Δεν απαιτείται εσωτερική πρόσβαση (white-box) στο μοντέλο λεζάντας, άρα εφαρμόζεται ενιαία σε διάφορα VL μοντέλα.

Η μελέτη των παραισθήσεων στην περιγραφή εικόνων αναδεικνύει την ανάγκη για πιο *εξηγήσιμες* μετρικές και αλγορίθμους. Στο μέλλον, ενδείκνυται:

- **Επέκταση σε επιπλέον πόρους γνώσης**: Ενσωμάτωση ConceptNet, word embeddings (Word2Vec, BERT) για εντοπισμό πιο λεπτών σημασιολογικών λαθών.
- **Εφαρμογή σε πολυπλαίσια δεδομένα**: Video captioning ή *διαλογικά* σενάρια, όπου οι παραισθήσεις μπορεί να εμφανιστούν διαδοχικά.
- **Ρυθμίσεις εκπαίδευσης**: Χρήση του CSED ή GED ως πρόσθετο loss term, ώστε το μοντέλο να τιμωρείται όταν επινοεί μη υπαρκτά αντικείμενα.

Συνολικά, το παρόν κεφάλαιο σκιαγραφεί έναν νέο *εξηγήσιμο* τρόπο για να μετράμε την ποιότητα και αληθοφάνεια σε συστήματα περιγραφής εικόνων, αντιμετωπίζοντας το πρόβλημα των παραισθήσεων σε επίπεδο εννοιών και συσχετίσεων. Η δυνατότητα *σαφούς διόρθωσης* των λαθών (το “πώς” διορθώνεται κάτι) υπερβαίνει την απλή βαθμολόγηση και προωθεί τη διαφάνεια και την εμπιστοσύνη στα μοντέλα της τεχνητής νοημοσύνης.

1.7 Χρήση Αντιπαραδειγμάτων για τη Βελτίωση των Ικανοτήτων Συλλογισμού των Μοντέλων Μεγάλων Γλωσσών

Οι πρόσφατες εξελίξεις στα Μεγάλα Γλωσσικά Μοντέλα (LLMs) όπως το GPT-3 και το GPT-4 έχουν αποκαλύψει σημαντικές δυνατότητες συλλογισμού σε ευρύ φάσμα πεδίων. Αν και τα μοντέλα αυτά έχουν επιτύχει αξιοσημείωτες προόδους στον παραγωγικό (deductive) συλλογισμό, **δυσκολεύονται** σε περιπτώσεις όπου απαιτούνται *inductive* λογικές δεξιότητες.

Στο Κεφάλαιο αυτό, παρουσιάζεται ένα σχήμα ταξινόμησης το οποίο εστιάζει κυρίως στη γνωστική διαδικασία και τις απαιτούμενες δεξιότητες για την επίλυση «puzzle»-τύπου προβλημάτων, αντί να επικεντρώνεται αποκλειστικά στην τυπική κατηγοριοποίηση με βάση τη μορφή της ερώτησης (π.., πολλαπλής επιλογής, σύντομης απάντησης) ή το είδος του συλλογισμού (παραγωγικός, επαγωγικός, παραγωγικός με εξαίρεση κ.λπ.). Για παράδειγμα, γρίφοι όπως τα Sudoku ή τα σταυρόλεξα στηρίζονται σε *κανόνες* και απαιτούν στρατηγικές που αξιοποιούν συγκεκριμένες κινήσεις μέσα σε *αυστηρά ορισμένο* χώρο κατάστασης. Από την άλλη, οι προγραμματιστικοί γρίφοι (*Programming Puzzles*) ή προβλήματα που αξιοποιούν ευρύτερη «κοινή γνώση» (commonsense) και επιπλέον λογικές διεργασίες δεν βασίζονται σε προκαθορισμένους κανόνες αλλά στην ικανότητα του μοντέλου να εφαρμόζει ευρύτερες γνωστικές δεξιότητες.

Για αυτό, παρουσιάζεται μια ταξινόμηση παζλ, χωρισμένων ανάλογα με το αν προϋποθέτουν *αυστηρά formal rules* ή περισσότερο *ευέλικτη-ελεύθερη* (rule-less) σκέψη, δείχνοντας τη διαφορετική φύση των λογικών προκλήσεων που προκύπτουν. Το παρόν κεφάλαιο πραγματεύεται **πώς** οι γρίφοι που δημιουργούνται ως αντιπαραδείγματα (*counterfactual*) μπορούν να βελτιώσουν περαιτέρω τις ικανότητες συλλογισμού των LLMs, εστιάζοντας κυρίως σε *riddle-solving tasks*. Με έμφαση στην παραγωγή "εναλλακτικών" γρίφων με ίδιο συλλογιστικό πυρήνα αλλά αλλαγμένο πλαίσιο (context), αναδεικνύεται η αξία του να αποκτούν τα μοντέλα *διαφορετικές οπτικές* του ίδιου νοητικού μοτίβου, βελτιώνοντας τις γενικές τους ικανότητες λογικής και προσαρμοστικότητας.

1.7.1 Χρήση LLMs για την Επίλυση Προβλημάτων Γρίφων

Με την ενσωμάτωση των LLMs σε προβλήματα επίλυσης γρίφων, η ερευνητική κοινότητα έχει αναπτύξει τεχνικές που βελτιώνουν τις ικανότητες λογικής, π.χ. μέσω *prompting*, *neuro-symbolic* προσεγγίσεων, καθώς και με τη χρήση *fine-tuning* σε εξειδικευμένα σετ δεδομένων. Διάφορες μέθοδοι prompting (π.. Chain-of-Thought, Self-Consistency, Tree-of-Thought) διατυπώνουν *διάμεσες επεξηγήσεις* (reasoning steps), αποδεικνύοντας βελτιωμένη ακρίβεια σε λογικές διεργασίες. Ωστόσο, ακόμα και οι καλύτερες υλοποιήσεις παραμένουν ευαίσθητες στην *ποιότητα* και την *ποικιλία* των παραδειγμάτων που παρέχονται ως in-context (few-shot) ενδείξεις.

1.7.2 Δημιουργία Γρίφων μέσω Αντιπαραδειγμάτων

Η χρήση (*counterfactual*) για την κατανόηση και την ενίσχυση των δυνατοτήτων των συστημάτων ΤΝ, αποτελεί βασική ιδέα στη βιβλιογραφία XAI (Explainable AI). Με παρόμοιο σκεπτικό, η παραγωγή *counterfactual riddles* στοχεύει στη δημιουργία *εναλλακτικών γρίφων* που απαιτούν την ίδια λογική διαδρομή αλλά θέτουν ένα *διαφορετικό πλαίσιο* (*context*). Παρόμοιοι όροι, όπως *context-reconstructed riddles* ή *alternative puzzles*, χρησιμοποιούνται στη βιβλιογραφία. Όταν τα μοντέλα εκτίθενται τόσο στο αυθεντικό όσο και στο ανακατασκευασμένο παράδειγμα, ενισχύεται η ικανότητά τους να κατανοούν βαθύτερα το *reasoning pattern* — αντί να απλώς προσαρμόζονται στα επιφανειακά (*semantic*) χαρακτηριστικά του πρωτότυπου δείγματος.

1.7.3 Μεθοδολογία

Έστω ένας πρωτότυπος γρίφος *"Ξυρίζεται κάθε μέρα, αλλά τα γένια του παραμένουν μακριά"*, με απάντηση *"κουρέας" (barber)*. Για να δημιουργήσουμε ένα ανακατασκευασμένο (*context-reconstructed*) γρίφο, μπορούμε να αλλάξουμε το πλαίσιο διατηρώντας την ίδια *λογική* που οδηγεί στην απάντηση: *"Παίρνει συνεχώς μέτρα ρούχων, μα δεν έχει ποτέ δικά του..."* (*τύπου "ράφτης"*). Σημασία έχει ότι η *συλλογιστική διαδρομή* —η ειρωνική ή παράδοξη πτυχή— παραμένει η ίδια, ενώ το θέμα (*context*) μεταβάλλεται.

1.7.4 Η Μέθοδος RISCORE

Η μέθοδος **RISCORE** (*RIddle Solving with CO-ntext RE-construction*) εισάγει *ανακατασκευασμένους γρίφους* στη διαδικασία few-shot prompting. Συγκεκριμένα:

- **Αρχική επιλογή παραδειγμάτων (exemplars):** Γίνεται συνήθως με όμοιες τεχνικές (π.. semantic similarity).
- **Ανακατασκευή ερώτησης και σωστής απάντησης:** Για κάθε επιλεγμένο παράδειγμα, παράγεται ένας *context-reconstructed* γρίφος που αξιοποιεί την ίδια λογική, αλλά σε εναλλακτικό πλαίσιο (context).
- **Δημιουργία λανθασμένων επιλογών:** Προστίθενται παραπλανητικές επιλογές (distractors) που είναι *λανθασμένες* αλλά *πειστικές*, ώστε να ενισχυθεί η δυσκολία.
- **Διάταξη (prompting):** Τελικά, στον prompt παρέχονται τόσο το αρχικό όσο και το ανακατασκευασμένο παράδειγμα, με στόχο να αναδειχθεί η κοινή συλλογιστική διαδρομή και να βελτιωθεί η γενίκευση.

Η διαδικασία παραγωγής έχει δύο βήματα: πρώτον φτιάχνεται ο νέος γρίφος-απάντηση (χωρίς distractors), ύστερα δημιουργούνται κατάλληλα distractors (λανθασμένες επιλογές) σε σχήμα πολλαπλής επιλογής. Για dataset με *παραγωγικές* (creative) απαντήσεις, αξιοποιείται η ικανότητα του LLM να μεταφέρει την ίδια ιδέα σε *διαφορετικά context*, ενώ για datasets με μονολεκτικές απαντήσεις (*π.χ. RiddleSense*) υιοθετούνται διαφορετικές κατηγορίες τύπου *(food, person, object, animal, nature, time, place, concept)* για να εξασφαλίσουμε ότι οι *distractors* ανήκουν σε άλλη κατηγορία από τη σωστή.

1.7.5 Πειράματα

Για να αξιολογηθεί η αποτελεσματικότητα της μεθόδου RISCORE, επιλέχθηκαν δύο σύνολα δεδομένων το BrainTeaser και το RiddleSense.

- **BrainTeaser:** Επικεντρώνεται σε *lateral (πλευρική) σκέψη* όπου οι γρίφοι απαιτούν δημιουργικότητα και «άλματα λογικής», με 4 επιλογές απάντησης (η τελευταία “None of the above”). Επιπλέον, περιλαμβάνει *manually crafted* αντίστοιχα context-reconstructed δείγματα, τα οποία χρησιμοποιήθηκαν ως **ανώτατο όριο σύγκρισης** (upper bound) ποιότητας.
- **RiddleSense:** Αντίθετα, εστιάζει κυρίως σε *vertical reasoning* γρίφους (*αλληλουχίες λογικών βημάτων*). Δεν διαθέτει reconstructions, οπότε εφαρμόστηκε αποκλειστικά η αυτοματοποιημένη μέθοδος παραγωγής ανακατασκευασμένων παραδειγμάτων.

Επιπλέον, δοκιμάστηκαν πολλαπλές τεχνικές prompting:

1. **Zero-shot (ZS):** Με ή χωρίς παρότρυνση “Let’s think step-by-step” (*CoT ZS*).
2. **Few-shot (FS):** Με *2, 4, 8* παραδείγματα, επιλεγμένα τυχαία (*Rand*) ή με βάση semantic similarity (*Sim*).
3. **CoT FS:** Όπου τα παραδείγματα συνοδεύονται από αναλυτικές επεξηγήσεις (chain-of-thought) οι όποιες έχουν δημιουργηθεί χειροκίνητα.
4. **RISCORE (automated & manual)**: Συνδυάζει $N/2$ αυθεντικά παραδείγματα + $N/2$ context-reconstructed, διατηρώντας το ίδιο συνολικό πλήθος N. Όπου υπάρχουν διαθέσιμα ανακατασκευασμένα από ανθρώπους (manual reconstructions), αυτά προσφέρουν upper bound αποτελέσματα ($RISCORE_m$).

Χρησιμοποιήθηκαν διάφορα μοντέλα, όπως Llama3 (8B, 70B), Mistral (7B, 8x7B), Qwen2-7B, σε *black-box* συνθήκες.

Αποτελέσματα

BrainTeaser Οι δοκιμές έδειξαν ότι **RISCORE$_\mathbf{m}$** (χειροκίνητες reconstructions) συστηματικά βελτιώνει τις επιδόσεις έναντι της βασικής μεθόδου few-shot. Ακόμα και όταν τα 2/4 επιπλέον παραδείγματα επιλέγονται όχι βέλτιστα, τα *contextual reconstructions* «διορθώνουν» τη συνήθη αστάθεια που εμφανίζεται στην επιλογή παραδειγμάτων. Επιπλέον, ακόμα και η αυτόματη παραγωγή ανακατασκευασμένων γρίφων μέσω της RISCORE, εμφανίζει *σημαντικές βελτιώσεις* σε σχέση με τυπικά 4-shot ή 8-shot prompts. Για παράδειγμα, με Llama3-70B, από 0.783 (8-shot *FS Sim*) φτάνουμε 0.808 (8-shot *RISCORE*), δείχνοντας τη χρησιμότητα των ανακατασκευασμένων γρίφων σε γρίφους που απαιτούν lateral thinking ικανότητες για την επίλυσή τους.

RiddleSense Καθώς το RiddleSense δεν περιείχε ανακατακσευασμένους γρίφους από ανθρώπους, χρησιμοποιήθηκε αποκλειστικά η αυτοματοποιημένη εκδοχή *RISCORE*. Και εδώ παρατηρήθηκε βελτίωση σε σχέση με το κλασικό 8-shot *FS Sim*, παρά το γεγονός ότι η μεθοδολογία βασίστηκε σε μικρότερο (*N/2*) πλήθος κανονικών παραδειγμάτων. Επιπλέον, διαπιστώθηκε ότι τα *vertical reasoning* tasks είναι ευκολότερα για μικρότερα μοντέλα (π.. Llama3-8B), τα οποία απέδιδαν επαρκώς στο στάδιο της παραγωγής (context-reconstruction) και βελτίωσαν τα τελικά σκορ τους.

Ποιότητα Παραγόμενων Γρίφων Ορισμένα ζητήματα προέκυψαν κυρίως στο BrainTeaser, όπου ο μικρότερος Llama3-8B *αδυνατούσε* να παράγει ποιοτικές Q-A pairs σε lateral puzzles. Ωστόσο, η ύπαρξη φιλτραρίσματος και κανόνων ποιότητας *εξαίρεσαν* χαμηλής ποιότητας δείγματα, εξασφαλίζοντας ότι τελικά χρησιμοποιούμε μόνο *υψηλής ποιότητας ανακατασκευασμένους γρίφους*. Στο RiddleSense (vertical reasoning), αντιθέτως, ο ίδιος μικρότερος μοντέλο απέδωσε ικανοποιητικά.

1.7.6 Συμπεράσματα

Το παρόν κεφάλαιο ανέλυσε μια μέθοδο (**RISCORE**) για την παραγωγή ανακατασκευασμένων γρίφων, δηλαδή *επανασχηματισμένων παραδειγμάτων* που διατηρούν την ίδια λογική-συλλογιστική πορεία σε διαφορετικό *context*. Η μέθοδος αυτή εντάσσεται σε *few-shot prompting* ρυθμίσεις και στοχεύει στη βελτίωση των ικανοτήτων συλλογισμού των Μεγάλων Γλωσσικών Μοντέλων.

- Σε datasets όπως το *BrainTeaser*, όπου υπάρχουν *manually crafted* reconstructions, η **RISCORE$_\mathbf{m}$** προσεγγίζει υψηλές επιδόσεις και συχνά υπερέχει των παραδοσιακών strategies (π.. 8-shot *FS Sim*).
- Όπου δεν διατίθενται ανακατασκευασμένοι γρίφοι από ανθρώπους (π.. RiddleSense), η αυτόματη μέθοδος παραγωγής τους **RISCORE** διαμορφώνει επιπλέον παραδείγματα *counterfactual riddles* δημιουργώντας αξιόλογη βελτίωση.
- Η ποιότητα των παραγόμενων γρίφων είναι καθοριστική. Απαιτείται κατάλληλο φιλτράρισμα και τεχνικές επιλογής παραδειγμάτων για να αποφευχθεί η εισαγωγή θορύβου.
- Τέλος, επισημαίνεται ότι το *context reconstruction* ενισχύει τη *γενίκευση* των LLMs σε γρίφους με παρόμοιο συλλογιστικό πυρήνα αλλά διαφοροποιημένο γλωσσικό περιβάλλον.

Συνολικά, η παραγωγή και η χρήση ανακατασκευασμένων γρίφων (*counterfactual riddles*) στην είσοδο συνιστά μια αποτελεσματική προσέγγιση για την περαιτέρω ανάπτυξη της *συλλογιστικής ικανότητας* των LLMs, προσφέροντας *εναλλακτικές οπτικές* του ίδιου λογικού μοτίβου και οδηγώντας σε βελτιωμένες επιδόσεις σε lateral και vertical reasoning tasks. Οι μελλοντικές επεκτάσεις μπορούν να διερευνήσουν το πώς η μέθοδος αυτή εφαρμόζεται σε ακόμη πιο πολύπλοκα puzzles (π.. *stochastic* ή *multi-step* games).

1.8 Επεξηγήσεις μέσω Αντιπαραδειγμάτων για τη Σύσταση Προιόντων μέσω Μεγάλων Γλωσσικών Μοντέλων

Το κεφάλαιο ξεκινάει με την επισκόπηση της παραδοσιακής χρήσης των αντιπαραδειγματικών (counterfactual) επεξηγήσεων στη μηχανική μάθηση. Συνήθως, οι τεχνικές αυτές επιστρατεύονται για να καταδείξουν πώς θα μπορούσε να αλλάξει η έξοδος ενός μοντέλου ταξινόμησης (π.χ. έγκριση ή απόρριψη δανείου) εάν μεταβάλλονταν συγκεκριμένα χαρακτηριστικά εισόδου. Ωστόσο, η συγγραφική ομάδα προτείνει ότι η ίδια λογική επεξηγηματικής παρέμβασης μπορεί να προσαρμοστεί και σε Μεγάλα Γλωσσικά Μοντέλα (LLMs), όχι απλώς για να *ερμηνεύσουμε* τις προτάσεις προϊόντων που παράγουν, αλλά και για να τις *επηρεάσουμε* με στοχευμένο τρόπο.

Σε αντίθεση με τα περισσότερα συστήματα προτάσεων που στοχεύουν απλώς στην ικανοποίηση του χρήστη, η συζήτηση εστιάζεται στη μελέτη του πώς μικρές τροποποιήσεις στα κείμενα περιγραφής προϊόντων μπορούν να αλλοιώσουν την ορατότητα (visibility) και τη σειρά κατάταξης αυτών των προϊόντων στις προτάσεις ενός LLM. Η έμφαση δίνεται στην ερμηνεία των αποτελεσμάτων αυτών, με το επιπρόσθετο όφελος ότι παρατηρείται και το *πώς* αντιδρά το ίδιο το μοντέλο στις λεκτικές παρεμβάσεις.

1.8.1 Γνωστικές Προκαταλήψεις ως Στρατηγικές Επίθεσης σε LLMs

Η κεντρική σύλληψη του κεφαλαίου βασίζεται στην αξιοποίηση γνωστικών προκαταλήψεων (cognitive biases), βαθιά ριζωμένων στην ανθρώπινη ψυχολογία, ώστε να ενεργήσουν ως «αθόρυβες» επιθετικές τεχνικές για την παραπλάνηση των LLM. Εδώ, απλές φράσεις όπως «Περισσότεροι από 10.000 αγοραστές επέλεξαν αυτό το προϊόν τον τελευταίο μήνα» (*social proof*) ή «Αποκλειστικά για απαιτητικούς χρήστες» (*exclusivity*) εισάγονται φυσιολογικά στο κείμενο των περιγραφών. Σε αντίθεση με τις παραδοσιακές επιθέσεις που εισάγουν ακατανόητους ή τυχαίους χαρακτήρες, οι συγκεκριμένες παρεμβάσεις μοιάζουν με συνηθισμένες προωθητικές φράσεις του μάρκετινγκ.

Το κεφάλαιο περιγράφει δύο βασικές μεθόδους:

1. **Επεξεργασίες από Ειδικούς (Expert-Crafted Edits):** Συνοπτικές, στοχευμένες προτάσεις προστίθενται χειροκίνητα από επαγγελματίες του μάρκετινγκ (π.χ. «Προϊόν με τη μεγαλύτερη δημοφιλία στην κατηγορία του»).
2. **Αυτόματες Γεννήσεις Περιγραφών (Generated Edits):** Ολόκληρη η περιγραφή ανασυντάσσεται από ένα LLM (όπως ο Claude 3.5 Sonnet) ώστε η προκατάληψη να ενσωματωθεί αβίαστα στο κείμενο. Έτσι, το τελικό αποτέλεσμα εμφανίζεται πιο φυσικό, ελαχιστοποιώντας τον κίνδυνο εντοπισμού.

Αμφότερες οι προσεγγίσεις στοχεύουν στην αναβάθμιση ή υποβάθμιση της θέσης ενός προϊόντος στις προτάσεις, εκμεταλλευόμενες την ενδεχόμενη «προδιάθεση» ενός LLM σε συγκεκριμένες λεκτικές διατυπώσεις.

1.9 Πειραματική Διάταξη και Δεδομένα

Συνθετικά Δεδομένα Αρχικά, η ανάλυση βασίστηκε σε μικρά, ελεγχόμενα σύνολα προϊόντων (π.χ. 10 καφετιέρες, 10 κάμερες, 10 βιβλία), ώστε να τεκμηριωθεί η επίδραση κάθε πλαγιορμήσης σε συνθήκες χωρίς «θόρυβο». Χρησιμοποιούνται ελάχιστες μεταβλητές —όνομα, τιμή, βαθμολογία, περιγραφή— έτσι ώστε να είναι ξεκάθαρο ότι τυχόν αλλαγές στο LLM προκαλούνται αμιγώς από τις λεκτικές τροποποιήσεις.

Δεδομένα Από Amazon Για να αποδειχθεί ότι οι ίδιες τεχνικές επιδρούν και σε αληθινά περιβάλλοντα, το κεφάλαιο μεταβαίνει σε πραγματικές λίστες προϊόντων από το Amazon Reviews. Εδώ, οι περιγραφές είναι συχνά εκτενέστερες και εμπλουτίζονται με τεχνικά χαρακτηριστικά, ήδη ενσωματωμένα διαφημιστικά στοιχεία, επισημάνσεις αξιολόγησης κ.λπ. Παρ' όλ' αυτά, η εισαγωγή γνωστικών πλαγιορμήσεων διατηρεί αισθητό αντίκτυπο στη συχνότητα εμφάνισης και τη θέση που λαμβάνει κάθε προϊόν στις προτάσεις.

LLMs Τα πειράματα διεξάγονται σε μια ποικιλία μοντέλων:

- *Open-source:* Διάφορες εκδόσεις Llama (8B, 70B, 405B).
- *Κλειστού κώδικα:* Claude 3.5 Sonnet και Mistral 2 large.

Η διαφοροποίηση σε κλίμακα, αρχιτεκτονική και εκπαίδευση βοηθά να διαπιστωθεί κατά πόσο η ευπάθεια στα λεκτικά «τρικ» είναι κοινή σε όλα τα LLMs.

Μετρικές και Μέθοδοι Αξιολόγησης Δύο βασικές μετρικές ξεχωρίζουν:

1. **Συχνότητα Σύστασης (Recommendation Frequency):** Πόσο συχνά προτείνεται ένα συγκεκριμένο προϊόν (σε πολλαπλές εκτελέσεις).
2. **Θέση στη Λίστα (Ranking Position) & MRR:** Ο μέσος όρος της κατάταξης (με έμφαση στην επάνω θέση), καθώς και πόσο βελτιώνεται ή χειροτερεύει σε σχέση με τα αρχικά δεδομένα.

Έτσι, αν η παρεμβολή μιας φράσης τύπου *discount framing* (π.χ. «Προσφορά 25%!») κάνει το προϊόν να εμφανίζεται από την πέμπτη στη δεύτερη θέση, καταγράφεται σημαντική θετική αλλαγή.

1.10 Αποτελέσματα και Συμπεράσματα

Ισχυρές Θετικές Προκαταλήψεις **Social Proof** και **Discount Framing** αναδεικνύονται ως οι δύο ισχυρότερες στρατηγικές για την ενίσχυση ορατότητας ενός προϊόντος. Όταν μια περιγραφή περιέχει ισχυρή ένδειξη κοινωνικής απήχησης («Χιλιάδες αγοραστές εμπιστεύτηκαν το προϊόν») ή κάποια μορφή εκπτώσεων («Αρχική τιμή 100€, τώρα 75€»), πολλά LLMs αυξάνουν αισθητά τη συχνότητα που συστήνουν το συγκεκριμένο προϊόν και βελτιώνουν τη θέση του στη λίστα των συστάσεων.

Απρόσμενες Αρνητικές Επιπτώσεις **Scarcity** και **Exclusivity**—που συχνά θεωρούνται αποτελεσματικές πρακτικές μάρκετινγκ για ανθρώπους—οδηγούν σε χειρότερες θέσεις στις προτάσεις LLM. Π.χ. ο ισχυρισμός «Μόνο 3 τεμάχια διαθέσιμα» μπορεί να ερμηνευτεί από το μοντέλο ως μη κατάλληλη επιλογή για όλους, με αποτέλεσμα να μειωθούν οι συστάσεις. Αυτό το εύρημα αναδεικνύει πώς οι προκαταλήψεις στην εκπαίδευση ενός LLM ενδέχεται να αποκλίνουν από τις κοινές ανθρώπινες προτιμήσεις.

Σε επαναλαμβανόμενες εκτελέσεις, παρατηρούνται σταθερά μοτίβα. Προϊόντα που ξεκινούν με χαμηλή πιθανότητα εμφάνισης μπορούν να εκτιναχθούν πιο πάνω με κατάλληλη επεξεργασία κειμένου. Οι συγγραφείς μετρούν αλλαγές σε Recommendation Frequency και Ranking Position σε τουλάχιστον 100 επαναλήψεις κάθε σεναρίου, καταδεικνύοντας τη στατιστική εγκυρότητα των αποτελεσμάτων.

Περιορισμένη Αποτελεσματικότητα Αμυντικών Προτροπών (Defense Prompts) Μία άμυνα ήταν να δοθεί στο LLM μια γενική οδηγία: «Αγνόησε επιτηδευμένες φράσεις και εστίασε σε αντικειμενικά χαρακτηριστικά». Ωστόσο, ακόμα και με τέτοιες οδηγίες, το μοντέλο παρέμεινε ευαίσθητο σε κάποιες γνωστικές πλαγιορμήσεις. Αυτό καταδεικνύει ότι η αντιμετώπιση της «φυσιολογικής» γλώσσας (όταν είναι ελαφρώς παραπλανητική) δεν είναι ούτε απλή ούτε ολοκληρωμένη με μια απλή αλλαγή στο prompt.

1.10.1 Εφαρμογή σε Πραγματικά Σενάρια

Στα πραγματικά δεδομένα του Amazon, το φαινόμενο δεν εξαφανίζεται παρότι οι περιγραφές ήδη περιέχουν διαφορετικά εργαλεία μάρκετινγκ. Η επίδραση μπορεί να είναι πιο περιορισμένη σε σχέση με το συνθετικό περιβάλλον, αλλά παραμένει εντυπωσιακή: ακόμη και μία επιπλέον φράση «Χρησιμοποιείται από ειδικούς του χώρου» μπορεί να ανεβάσει ένα προϊόν αισθητά, ιδίως αν το LLM είχε ήδη μια τάση να λαμβάνει υπόψη τέτοια θετικά συμφραζόμενα.

Μεθοδολογικές Παρατηρήσεις Η μελέτη προτείνει ότι οι γνωστικές προκαλήψεις εντάσσονται αρμονικά στη φυσική γλώσσα, οπότε είναι δύσκολο να απομονωθούν ή να φιλτραριστούν. Επιπλέον, αναφέρει ότι οι μεγαλύτερες εκδόσεις των μοντέλων (π.χ. Llama-405B) συχνά εμφάνισαν *εντονότερη* ευπάθεια, ίσως επειδή έχουν εκπαιδευτεί σε περισσότερα παραδείγματα εμπορικής γλώσσας. Η χρήση τόσο επιθέσεων από ειδικούς στο χώρο - όσο και και από LLMs φωτίζει το γεγονός ότι μια φαινομενικά κοινή φράση ή αφήγηση μπορεί να δημιουργήσει δυσανάλογη επίδραση στις συστάσεις.

Ευρύτερες Προεκτάσεις Το κεφάλαιο επισημαίνει ότι οι ευπάθειες αυτές δεν περιορίζονται μόνο στο ηλεκτρονικό εμπόριο. Κάθε χρήση μεγάλων γλωσσικών μοντέλων που επιβάλλει κατάταξη, εύρεση ή συμπερίληψη περιεχομένου—όπως ειδησεογραφικές συγκεντρώσεις ή αξιολογήσεις ερευνητικών άρθρων—μπορεί να διαστρεβλωθεί μέσω τέτοιων τεχνικών. Επίσης χρήζει μελέτης η πιθανή σύνδεση των προκαλήψεων με τη διασπορά παραπληροφόρησης ή της ενίσχυσης κείμενων που «κλίνουν» υπέρ μιας οπτικής, απλώς προσθέτοντας παρόμοιες, πειστικές φράσεις.

Σε πρακτικό επίπεδο, δίδεται έμφαση στη μελλοντική έρευνα για:

- *Εξεύρεση αμυντικών στρατηγικών* που μπορούν να διακρίνουν μεταξύ αληθούς πληροφορίας (π.χ. οντως μεγάλη βάση χρηστών) και στημένων δηλώσεων που στοχεύουν σε μοντέλα.
- *Διεύρυνση* των τύπων προκαταλήψεων (π.χ. storytelling, anchoring), έτσι ώστε να εντοπιστεί ένα ευρύτερο φάσμα δυνητικών παραβιάσεων.
- *Συμπεριφορική ανάλυση* μεγάλων γλωσσικών μοντέλων, ώστε να φανερωθούν οι ενδείξεις που τα κάνουν πιο ευάλωτα σε τέτοια λεκτικά ερεθίσματα.

1.11 Συμπέρασμα

Το κεφάλαιο καταδεικνύει πώς οι αντιπαραδειγματικές επεξηγήσεις μπορούν να επεκταθούν πέραν της απλής ερμηνείας και να χρησιμοποιηθούν ως «όπλα» για να μετακινήσουν τους αλγορίθμους προτάσεων LLM προς συγκεκριμένες κατευθύνσεις. Αξιοποιώντας ανθρώπινες γνωστικές πλαγιορμήσεις, οι συγγραφείς δείχνουν ότι ακόμη και ένας λόγος μάρκετινγκ που φαίνεται αθώος μπορεί να αποδειχθεί ισχυρός μοχλός διαμόρφωσης προτάσεων.

Οι πειραματισμοί αναδεικνύουν τη σημασία του να δημιουργηθούν πιο ανθεκτικά (robust) μοντέλα που δεν παρασύρονται εύκολα από παρόμοιες ρητορικές στρατηγικές. Καθώς ολοένα και περισσότερες εφαρμογές προχωρούν σε LLMs για εξατομικευμένες προτάσεις, η κατανόηση αυτών των τρωτών σημείων είναι εξαιρετικά σημαντική για να διαφυλαχθεί η δικαιοσύνη, η αξιοπιστία και η ακεραιότητα των συστημάτων αυτών.

Chapter 2

Introduction

Artificial intelligence (AI) has been undergoing a rapid transformation, evolving from conceptual prototypes to high-stakes applications across healthcare, finance, security, transportation, and more. Although AI systems have demonstrated remarkable performance in tasks such as image recognition, natural language understanding, and autonomous decision-making, their complexity often renders their internal decision-making processes opaque. This opaqueness can erode trust, hamper adoptability, and, in certain safety-critical contexts, introduce risk. The field of Explainable AI (XAI) has therefore emerged as a critical area of research, aimed at developing methods, frameworks, and tools to help users understand, trust, and effectively interact with these models.

Within the broader landscape of XAI, counterfactual explanations have risen to prominence because they offer actionable insights into how a model's output might change given slight adjustments to its input. Instead of merely stating why a specific outcome was produced, counterfactual explanations demonstrate how to arrive at an alternative outcome—often by modifying a small subset of input features. This is especially compelling in real-world applications (e.g., finance or medical domains), where end-users can interpret and act upon suggestions such as "If you reduce your credit card debt by X amount, your loan application might be approved," or "If a patient's blood pressure were lower, the predicted diagnosis would be different."

Despite the potential of counterfactual explanations, existing methods often work at a low level, focusing on raw input features like individual pixels in an image or the exact words in a text. This level of granularity, while precise, is not always aligned with how humans conceptualize and communicate ideas. The notion of a "red color" in an image, for example, is more intuitive than a specific numerical pixel value in a red color channel. In text processing, a user might reason about sentiment or topic, rather than about a specific word embedding or letter substitution. This gap between raw features and human-understandable concepts presents a central challenge in building explanations that are both accurate and interpretable.

A rapidly growing area in counterfactual research seeks to address this challenge by introducing conceptual counterfactual explanations. Conceptual counterfactuals operate at higher levels of abstraction—corresponding to semantically meaningful units such as attributes, categories, or symbolic concepts. These "concepts" might describe visual attributes (e.g., "presence of stripes," "round shape," "furry texture"), audio features (e.g., "high pitch," "rapid tempo"), or even textual properties (e.g.,"sentiment," "topic," "politeness")—all more naturally aligned with how humans reason about and describe the world. The overarching goal is to produce actionable insights that inform what needs to change conceptually to alter a model's decision, rather than just enumerating low-level perturbations that may be perplexing to a lay user.

This thesis aims to extend and deepen the theoretical and practical foundations of conceptual counterfactual explanations. We propose novel algorithms, frameworks, and metrics that, together, showcase the power, flexibility, and real-world applicability of conceptual counterfactuals across multiple data modalities (images, text, graphs, audio) and tasks (classification, generation, and more). We offer evidence that shifting to a concept-centric view not only boosts interpretability but can also streamline the computational process, leading to efficient and targeted edits that preserve semantic coherence.

2.1 Thesis Overview

This thesis is structured to guide the reader progressively from foundational ideas in XAI and counterfactual explanations to advanced, domain-specific methods for conceptual counterfactuals. We begin by setting the stage in Chapter 2 with a thorough review of the theoretical underpinnings of Explainable AI and relevant literature on counterfactual explanations. We then build upon these concepts step-by-step, delving into practical implementations of conceptual counterfactuals in various settings. Below is a high-level outline of the chapters and the core contributions they provide.

2.1.1 Framework for Computing Conceptual Counterfactual Explanations

The thesis begins its core contributions in **Chapter 3** by presenting a *unified framework* for conceptual counterfactual explanations. Traditional counterfactual explanation techniques in the literature typically operate at the raw-feature level (e.g., image pixels, numerical input features, or tokens in text). While this granular focus allows for precise optimization—like minimizing the total pixel change—it often suffers from low human interpretability. Users, especially those who are not domain experts in machine learning, can find it difficult to parse why a specific set of pixel intensities would need to be changed.

To address this gap, we introduce the notion of an explanation dataset, comprised of high-level concepts that align with human cognition and domain knowledge. These concepts might be manually curated or learned in a data-driven manner (for instance, by clustering feature embeddings). We show how working at the concept level allows for:

- **Actionable Edits**: Instead of telling a user to adjust a large matrix of raw data, we tell them to add or remove clearly defined attributes (e.g., "add more warmth in color" or "reduce background noise"), which is far more intuitive.
- **Semantic Coherence**: Concepts ensure that the set of changes remains internally consistent and meaningful. Editing a single concept, such as "color," will systematically affect a related set of pixels or audio frequencies.
- **Reduced Dimensionality**: In certain scenarios, conceptual editing significantly reduces the search space for valid counterfactuals, increasing computational efficiency and interpretability simultaneously.

Chapter 3 also offers a detailed algorithmic perspective, outlining how conceptual counterfactuals can be computed by systematically identifying which concepts to change, by how much, and in which combination to achieve a target outcome shift.

2.1.2 Counterfactual Explanations using Concepts

Chapter 3 introduces the framework; subsequent chapters—particularly the latter sections of Chapter 3 and onwards—show how these concepts translate into practical examples in a variety of domains. We describe in detail how high-level concepts can represent visual features in images, such as texture, shape, and color, as well as auditory characteristics, such as pitch range and harmonic structure. By abstracting away from raw data to these more interpretable building blocks, the entire model explanation pipeline becomes clearer and more aligned with human intuition.

For instance, if an image classification model labels an image of a cat as a "dog," a conceptual counterfactual might highlight changes in the "fur pattern" or "ear shape" concepts that would reclassify the image correctly. Similarly, in audio classification, the conceptual counterfactual might focus on altering "beat regularity" or "frequency band energy" to switch from one class to another. These high-level edits let end-users quickly understand why a misclassification occurred and what conceptual changes could rectify it.

In addition, we demonstrate how the principle of minimality—making the smallest conceptual edits necessary to achieve a different prediction—can ensure that the explanations are concise, meaningful, and easy to interpret. Collectively, these insights pave the way for a more user-friendly generation of counterfactual explanations in practical AI applications.

2.1.3 Conceptual Counterfactuals using Graphs

Following this, **Chapter 4** delves into the realm of *graph-structured data*. Graphs arise in numerous real-world contexts—ranging from social networks and knowledge graphs to biological networks (e.g., molecules, protein interactions) and sensor networks. In these domains, each node or subgraph can represent a concept or cluster of features (such as a functional group in a molecule), and edges may capture relationships between these concepts.

Within this chapter, we propose a method to decompose the graph into conceptual units that might reflect structural or semantic properties (e.g., presence of a specific substructure in a molecule, or a particular community structure in a social network). By adjusting these conceptual units—rather than just randomly removing or adding edges—our counterfactual explanations can be significantly more interpretable and relatable to domain experts. For instance, a chemist investigating why a molecule was predicted to be toxic might learn that the presence of a particular substructure or functional group was the crucial factor—and that removing or altering it leads to a non-toxic classification.

Central to this chapter is the use of *Graph Neural Networks (GNNs)*, which leverage graph topology and node/edge feature embeddings to make predictions. We provide a mechanism to search through relevant substructures and identify minimal conceptual changes within the graph that alter the GNN's output. We further show how domain experts can leverage these conceptual substructures to gain deeper insights into complex graph-based AI applications, including molecular property prediction, fraud detection in financial networks, or misinformation detection in social networks.

2.1.4 Optimal and Efficient Text Counterfactuals using GNN

Moving from images, audio, and graphs to the domain of natural language processing (NLP), **Chapter 5** details how to generate *optimal and efficient* text counterfactuals using specialized graph constructs. Text-based applications often require the ability to handle subtle changes, as a single word substitution can significantly alter the meaning or sentiment of a sentence. However, not all word substitutions are equal: some preserve semantic coherence, while others result in nonsensical or misleading statements.

Here, we introduce a bipartite graph structure that connects words in the original text with candidate synonyms or paraphrases. We then utilize GNN-based techniques to optimize a multi-objective criterion: (1) flipping the label of the classifier, (2) preserving semantic coherence, (3) maintaining grammatical correctness, and (4) ensuring minimal total edits. By using GNNs to propagate signals about plausible word substitutions, we prune the search space intelligently, identifying only those substitutions that are both valid and impactful.

This chapter highlights the importance of conceptual thinking even in text generation. Although we focus on word-level changes, these words can be seen as "concept placeholders," especially when they represent domain-specific jargon, sentiment-laden terms, or other high-level semantic indicators. Moreover, we discuss potential applications in fairness and bias mitigation, demonstrating how text counterfactuals can help identify and fix model behaviors that disproportionately affect particular demographic groups.

2.1.5 Evaluation of Counterfactual Explanations

The strength of any XAI method—and especially counterfactual explanations—ultimately hinges on robust evaluation metrics that can gauge the explanations' quality, consistency, and utility. **Chapter 6** addresses this crucial topic by:

- **Introducing domain-agnostic metrics** (e.g., proximity, sparsity, plausibility, actionability) that can be applied uniformly across multiple data modalities and tasks.
- **Highlighting domain-specific considerations**, such as ensuring grammatical correctness in text-based counterfactuals or preserving physically plausible changes in image- or audio-based ones.
- **Discussing the inconsistency problem** in counterfactual explanations, where multiple distinct sets of minimal changes may produce the same outcome shift. In some cases, this can be valuable—since it presents different "paths" to the same goal—but it can also generate confusion if the user lacks a clear ranking or prioritization of these alternatives.

We also introduce new metrics specific to conceptual counterfactuals, such as whether the chosen concepts align with human intuition and how effectively the user can implement or act upon the recommended conceptual changes. The chapter concludes with methodologies for systematically testing these metrics in large-scale experiments.

Chapters 7 and 8 then offer specialized evaluation methods, applying conceptual counterfactuals to generative tasks such as story visualization and image captioning. These chapters illustrate how conceptual edits can identify and rectify problems like "hallucinations" (when a model inserts details that are not present in the input) and other generative inaccuracies. By framing generative evaluation around conceptual counterfactuals, we show how developers and practitioners can gain deeper insights into why a generative model fails and how to steer it toward producing more coherent and trustworthy outputs.

Explanaible Metric for Story Visualization through Counterfactual Explanations

Chapter 8 applies conceptual counterfactuals to story visualization, where models generate image sequences from textual narratives. This task poses interpretability challenges because each textual segment can trigger changes in the generated scenes, risking hallucinations (fabricated details) or omissions of crucial elements. By framing these errors through conceptual edits—e.g., adding or removing objects, attributes, or characters—we reveal how the generative model responds to specific narrative concepts.

- **Conceptual Edits**: We define high-level visual concepts (e.g., "character X," "blue hat," "mountain background") based on the story text. By altering these concepts, we track how the model's outputs change, highlighting dependencies between textual cues and visual content.
- **Evaluation**: Through quantitative and qualitative analysis, we show how conceptual counterfactuals help diagnose generative inaccuracies, reduce unwarranted hallucinations, and improve coherence across sequential images. This approach offers a more precise and intuitive means to refine and control story visualization models compared to pixel-level or purely textual interventions.

Explainable Metric for Hallucination Detection in Image Captioning

Chapter 9 extends the idea of conceptual counterfactuals to image captioning, a domain where AI models risk describing nonexistent objects or mislabeling attributes. By focusing on a set of visual concepts (e.g., "dog," "table," "red color") derived from the image, we can remove or replace these concepts and observe whether the model's textual output changes accordingly. When the caption persists in mentioning a concept that was removed, we identify a hallucination.

- **Conceptual Consistency**: We introduce a new metric assessing alignment between detected image concepts and captioned concepts, providing clearer insights than traditional metrics (e.g., BLEU, CIDEr) for detecting fabricated elements.
- **Applications**: Beyond just detecting hallucinations, these conceptual edits enable interactive correction. By systematically identifying and removing unreliable concepts, we can fine-tune the model to generate more trustworthy and faithful captions.

Counterfactual Generation for Improving Reasoning Abilities of LLMs

Chapter 10 introduces a novel approach for enhancing the reasoning capabilities of LLMs) through the generation and usage of *counterfactual riddles.* By producing pairs of analogous riddles set in different contexts—yet tied together by the same logical pathways—this technique (dubbed RISCORE) improves generalization and performance across diverse puzzle-solving tasks. Notably, experimental findings on both lateral and vertical reasoning benchmarks indicate that providing these context-reconstructed exemplars in a few-shot setup significantly outperforms standard prompting methods, highlighting the value of focusing on consistent reasoning structures rather than superficial semantic cues.

Counterfactuals in LLM-Driven Product Recommendations

Lastly, in the field of LLMs, Chapter 11 demonstrates how counterfactual explanations can provide valuable insights into the decision-making processes of different LLMs when they function as product recommenders.

Specifically, this chapter delves into how subtle manipulations in product descriptions—particularly those employing psychological biases—can exert a surprisingly strong influence on how LLMs rank and recommend items. Traditionally, counterfactual explanations have focused on classification tasks, but here they are repurposed to introduce strategic edits in descriptions that exploit human cognitive tendencies such as social proof, discount framing, scarcity, and exclusivity. Through both expert-crafted and automatically generated text modifications, the authors show that these seemingly innocuous insertions can significantly elevate or diminish a product's visibility within LLM-based recommendation lists.

The experiments employ small synthetic datasets (e.g., coffee machines, cameras, books) alongside real-world Amazon product data to assess consistency of the effects. Across multiple LLMs—from open-source (Llama) to proprietary (Claude, Mistral)—biases like social proof and discount framing consistently boost a product's ranking, while scarcity or exclusivity often prove detrimental. Attempts to counteract these attacks by instructing models to ignore persuasive language only modestly reduce their influence. The findings underscore the need for more rigorous defenses: as LLMs become integral to personalized e-commerce and information systems, subtle adversarial wording can undercut fairness and reliability in AI-driven recommendations.

2.2 Structure of the Thesis

Bringing all these ideas together, the thesis is organized as follows:

- **Chapter 3** – *Background Material*: A comprehensive review of Explainable AI, the fundamentals of counterfactual explanations, and existing metrics for evaluating them. This chapter lays the theoretical groundwork for the subsequent chapters.
- **Chapter 4** – *Framework for Computing Conceptual Counterfactual Explanations*: The first core contribution, presenting our overarching framework for conceptual counterfactuals. We introduce the explanation dataset, define key concepts, and detail an algorithm for computing counterfactuals entirely at the concept level.
- **Chapter 5** – *Conceptual Counterfactuals using Graphs*: We adapt the conceptual counterfactual paradigm to graph-structured data, proposing methods to leverage GNNs for interpretable edits on subgraph concepts.
- **Chapter 6** – *Optimal and Efficient Text Counterfactuals using GNN*: A foray into natural language processing, showing how bipartite graphs and GNNs can be used to generate minimal text edits that shift a classifier's prediction while preserving semantic and grammatical integrity.
- **Chapter 7** – *Evaluation of Counterfactual Explanations*: An in-depth look at metrics and methodologies for assessing the quality of counterfactual explanations, including domain-agnostic and domain-specific measures. We discuss the inconsistency of counterfactual methods and introduce novel approaches to mitigate and interpret multiple solution paths.
- **Chapter 8** – *Explainable Metric for Story Visualization through Counterfactual Explanations*: We demonstrate how conceptual counterfactuals can serve as an explanatory tool in generative tasks like story visualization, detecting and revising problematic generative outputs.
- **Chapter 9** – *Explainable Metric for Hallucination Detection in Image Captioning*: Extending the generative analysis further, we develop a conceptual counterfactual-based framework to identify, measure, and address hallucinations in image captioning systems.
- **Chapter 10** – *Counterfactual Generation for Improving Reasoning Abilities of LLMs*: Building on previous chapters, we propose a novel strategy that uses counterfactual riddles to improve large language models' consistency, adaptability, and overall reasoning performance.
- **Chapter 11** – *Counterfactuals in LLM-Driven Product Recommendations*: Counterfactual explanations, repurposed from their traditional role in classification, demonstrate in this chapter how minimal, bias-driven edits to product descriptions can significantly manipulate LLM-based recommendation rankings.

- **Chapter 12** – *Conclusion*: A concise reflection on the thesis contributions, insights gained, and an outlook on future research directions. We highlight how conceptual counterfactuals can be extended or combined with other emerging topics, including causal inference, reinforcement learning, and federated settings.

In sum, this thesis presents an thorough exploration of conceptual counterfactual explanations, elucidating how these methods enhance interpretability, actionability, and trust in AI systems. By consolidating theoretical foundations with practical algorithms and robust evaluation protocols, we illustrate the versatility and impact of conceptual thinking in the ongoing quest to make AI models more understandable. Our contributions signify a meaningful step toward bridging the gap between how models compute and how humans think, ultimately fostering more symbiotic and responsible relationships between AI technologies and the societies they serve.

Chapter 3

Background Material

In this chapter, the focus is on providing a comprehensive overview of the background material relating to eXplainable AI (XAI), especially focusing at the role of counterfactual explanations. The discussion will explore the relationship between semantics and counterfactual explanations, emphasizing how semantics enhance the interpretability of AI models. Furthermore, it addresses the challenges associated with evaluating these methods, with a focus on key concepts crucial to the framework. As required, any additional background material deemed necessary will be provided at the beginning of each subsequent chapter.

3.1 Explainable AI

Public concerns about biases in machine learning (ML) models have heightened the demand for transparent AI [107, 12]. End users are increasingly recognizing that reliable AI outputs must be accompanied by clear explanations to foster trust. This trust is crucial for organizations, governments, and professionals to confidently integrate AI tools into their workflows. However, traditional means of explaining and establishing trust in software, such as code inspection, understanding program logic, or thorough testing, are not feasible with complex AI systems, necessitating alternatives like self-explanatory AI systems or external AI audits [56, 60, 83]. The "black box" problem, which became widely recognized with the spread of ML tools to end-users, has long been a challenge for researchers in the deep learning field [4]. Large language models (LLMs) [340, 263, 7, 345] and Large Vision Languge Models [51, 403], for example, are often used as black boxes by many, emphasizing the necessity for increased transparency. This has made the need for explainable AI (XAI) crucial to render these processes transparent and understandable in human-AI interactions. Furthermore, explanations not only help in identifying errors but also provide opportunities for enhancing AI systems, leading to the emergence of Explanation Engineering, a field focused on integrating explainability systematically into AI designs.

Despite its importance, explainability remains a vague term without a formal, universally accepted definition, largely because it is investigated from various perspectives each time and across various domains. Authors often use the term broadly, leaving its specific interpretation to the reader's intended use. While satisfactory explanations vary by scenario, certain aspects consistently appear across most definitions [12].

Audience The explanation for any AI system must be tailored to its specific audience, which can vary widely. For instance, consider a cancer detection model that detects and classifies the stages of cancer in an X-ray [151]. Each audience requires a different type of explanation:

- **AI Engineer**: A "good explanation" for an AI engineer would involve technical details, such as a cancer being identified because an area in the image shows abnormal brightness compared to the surrounding areas.
- **Domain Specialist (Oncologist)**: For an oncologist, the explanation might focus on medical specifics, such as a cancer being identified due to "calcification," which refers to the accumulation of calcium deposits within tissues, appearing as a bright blob on the X-ray.

- **End User (Patient)**: Patients, who need to feel confident in the diagnosis, are generally not concerned with the technical workings of the model or the specific terminology used to describe features in the X-ray. Therefore, the explanation for them should focus on how accurately the system has performed in similar cases.

- **Lawmakers, Insurance Companies, and Hospitals**: Decision-makers in the healthcare system, such as lawmakers, insurance companies, and hospitals that are considering using the model clinically, will want to know if the AI system's decision-making aligns with that of doctors in general [336, 274].

Each explanation is crafted to meet the needs and understanding of different stakeholders involved in the use and impact of the AI system.

Level of Explanation Explanations in this context are generally divided into two main types: *global* and *local*. Global explanations are designed to illuminate the model's decision-making process as a whole. They provide a comprehensive view of the model's operations across the entire dataset and assess the relative importance of different variables. For example, global explanations can reveal how significantly each feature influences the model's predictions across all data, potentially uncovering unwanted biases. This approach is particularly valuable for understanding the model's overall behavior and for communicating insights to stakeholders. Conversely, local explanations focus on individual predictions. They explore the reasons behind a specific prediction made by the model. Local explanations are essential when it is important to understand particular outcomes, such as in scenarios involving medical diagnoses or loan approvals. In general, global explanations offer a macro-level perspective of the model's functionality, while local explanations provide a micro-level view of specific predictions. It is worth noting that these types of explanations are not mutually exclusive, as there are cases where the local behavior can be interpreted using global explanations and vice versa. Our proposed algorithms also provide a method for calculating global explanations using a set of local ones.

Access to model The differences in model access can lead to diverse methods of retrieving explanations. "White box" explainability methods necessitate access to the model's internals, such as the weights of the neural network [100, 348]. However, the applicability of such algorithms is somewhat restricted as they are typically confined to proprietary models. Moreover, these algorithms generally lack the flexibility to be transferable across different models, domains, or modalities. On the other hand, "black box" explainability methods aim to decipher the decision-making of a model by analyzing only the input-output relationships. This type of explanation is inherently adaptable and can be transferred across a variety of problems. Nevertheless, a significant drawback of "black box" explanations is that they might yield misleading insights, as they do not consider the model's internal mechanisms [50, 307]. From our perspective, accessing the weights of a proprietary model poses a formidable object, whereas enhancing "black box" explainability methods may offer a more feasible solution. Thus, our framework is dedicated to crafting explanations strictly from a "black box" approach.

Forms of Explanations XAI employs a range of explanation types to make AI decisions both understandable and interpretable. Among these, "counterfactual explanations" stand out by illustrating the minimal changes required in the input to alter a decision, effectively showing how small adjustments can lead to different outcomes. Additionally, "rule-based" explanations articulate the decision-making process through a set of human-readable rules, often derived from decision trees or rule extraction algorithms. Another insightful approach within XAI involves the use of prototypes [239]—representative examples from the dataset that exemplify the key characteristics of specific decisions or output classes. These "prototypes" act as exemplars, simplifying complex models by showcasing typical instances that influence the model's predictions. In our research, we explore these diverse forms to assess their effectiveness and relevance. Nonetheless, our proposed framework primarily focuses on generating *counterfactual explanations*. This choice is influenced by their philosophical roots and their alignment with counterfactual thinking—a natural human cognitive process [30]. By considering alternative realities, counterfactual explanations help us intuitively grasp how decisions are made, thereby playing a vital role in XAI.

3.2 Counterfactual Explanations

Counterfactual explanations have their roots in philosophy [97] where counterfactual thinking [30] involves considering alternative realities and outcomes that are different from the actual ones. Counterfactual explanations are a specialized type of explanation in the realm of machine learning that provide insightful "what if" scenarios. Specifically, they illustrate how the output of a machine learning model would change if an input data point were modified from its original value x to a new value x', consequently altering the model's prediction from y to y'. This form of explanation is particularly valued for its intuitiveness and clarity, making complex model behaviors more accessible and understandable to users.

A practical example, inspired by GDPR regulations, involves querying a bank's AI system that has denied a loan application with the question, "What would need to change for my loan to be approved?" This inquiry can yield a range of possible answer combinations. The algorithm aims to identify the solution that requires the smallest adjustments, tailored specifically to the situation, while remaining practical and actionable in real-world scenarios[277]. A critical aspect of counterfactual explanations is their reliance on concepts of distance and similarity.

However, the similarity may not always be clear and understandable to an end-user, as it could potentially constitute an adversarial example that is indistinguishable from the original data sample. Instead, recent ideas suggest that the notion of minimality in the context of counterfactuals should refer to the semantics of the data sample rather than the feature space [354]. Numeric representations of real-world phenomena based on low-level features, such as pixel brightness or sound frequency, are not helpful or trustworthy to humans [307]. Humans prefer intuitive, high-level criteria when describing the factors that direct their decisions. For example, how "furry" a dog is or how "dry" a cough sounds when determining a dog breed or a respiratory issue, respectively. Low-level features may be useful information for machine predictions, but not for human-readable explanations. However, there is no mathematical difference between a vector representing low-level characteristics (e.g., pixel values) and one representing ally rich features [27]. This makes it feasible to create systems that provide counterfactual explanations in terms of semantic features instead of low-level characteristics. This argument has been grounded both theoretically and practically by the community. [27] demonstrates the equivalence between counterexamples and adversarial examples in cases where higher-level semantics are not employed.

Furthermore, recent research has explored diverse approaches to provide explanations that incorporate the semantic aspects of input data. For example, [100] utilize the intermediate output of a classifier to capture higher-level information concerning the input image. [8], on the other hand, infer an image's semantic concepts (referred to as "xconcepts") by clustering the outputs of these features, assuming that elements within the same cluster share semantic similarities. Meanwhile, [348] employ an external neural network to generate semantic embeddings of these features, with the notion that proximate embeddings signify semantic equivalence. All of these algorithms employ distinct methodologies to estimate the semantic relationships among the elements depicted in images.

Counterfactual Explanations in Visual Classification also involve methods of pixel-level editing aimed at identifying and modifying key areas of an image that significantly influence the predictions of the model [100, 348, 13]. These methods, some of which incorporate sophisticated generative technologies, serve as a critical component in understanding and adjusting the decision-making processes of visual classifiers.

In contrast to other counterfactual methods that extract features, the Counterfactual Visual Explanations (CVE) proposed by [348] are distinguished by their focus on ensuring semantic consistency during the exchange of local regions. This is achieved through an auxiliary component that assesses semantic similarity, enhancing the relevance and accuracy of pixel-level comparisons. This semantically oriented strategy not only establishes a standard for pixel-level evaluation but also sets it apart from methodologies that require direct access to classifiers, highlighting a key differentiation from our own approach. Additionally, a separate line of research delves into the modification of human-understandable concepts to generate Counterfactual Explanations (CEs). This approach prioritizes the clarity and interpretability of the edits, aiming to bridge the gap between machine learning outputs and human cognitive processes.

The aforementioned approaches have the capability to generate local counterfactual explanations by leveraging the semantic attributes of input instances. However, these methodologies exhibit certain limitations.

Firstly, they require “white-box” access to the classifier, a condition that is exceptionally uncommon and applicable primarily within a developer-centric context.

Secondly, their applicability is constrained to specific model types, predominantly Convolutional Neural Networks (CNNs), and is restricted to a particular input domain, namely images. Thirdly, these methods do not permit users to specify the features they deem as meaningful for the explanations. In the realm of explainability, it is crucial to acknowledge that different users possess distinct terminologies and expectations regarding the depth of abstraction in explanations, as it is already discussed in 3.1. Consequently, the choice of data used for explanation generation should encompass semantic information tailored to the requisite level of abstraction, aligning with the specific use-case at hand.

Additional techniques that have been proposed [277] describe these explanations as “feasible paths” within the data that adhere to the data’s distribution and meet both feasibility and actionability requirements. [415] use a text-to-image generative adversarial network to create counterfactual visual explanations, a method that also incorporates external knowledge rather than relying solely on a model’s predefined features and classes. For numeric tabular data, [98] have devised a heuristic approach to identify the smallest necessary changes for altering a classifier’s prediction, complemented by a visualization tool for users.

It’s important to note that these techniques are not actively employed in NLP, a domain where the capacity to easily generate high-quality data has greatly enhanced the development of counterfactual explanations. In this context, most methodologies are geared towards producing new counterfactual instances, going beyond mere explanations to actually modify elements of the text to illustrate alternative outcomes [302, 31, 375, 133, 183, 304, 41].

For instance, tools like MiCE [302] and DoCoGen [31] focus on optimizing text modifications based on the output of a specific predictor, $g()$. They achieve this by pseudo-randomly masking words and optimizing the suggested replacements to alter g’s output. In contrast, tools like Polyjuice [375] target generic text perturbations that shift the semantic meaning of sentences without being tied to any particular predictor. These are considered general-purpose counterfactuals and are versatile, used for everything from data augmentation to generating counterfactual explanations or tailoring to specific tasks or datasets.

Additionally, a significant group of editors is dedicated to creating adversarial examples, which are designed to uncover and highlight vulnerabilities in classifiers. Unlike other counterfactual editors, these do not necessarily strive for minimal or fluent edits and may introduce noise among other changes. Notable examples within the NLP field include TextFooler [133] and Bert-Attack [183], both integrated into the TextAttack framework [251]. These methods typically use gradient descent on text instances to modify the class prediction of a model while optimizing other metrics.

Rather than generating random permutations to create counterexamples, some editors focus only on modifying key features of a text. The importance of these features is determined in various ways, such as training a classifier to detect correlations between terms and tasks, assessing the impact of feature deletion on predictions, or leveraging a predictor’s attention mechanisms. Important terms may then be replaced with synonyms, antonyms, significant terms from other tasks, or through the use of pre-trained seq2seq models [229, 302, 375, 79].

3.3 Evaluation of AI Counterfactual Methods

The primary goal of counterfactual explanations is to identify what specific aspects of the input data would need to be different to achieve an alternative outcome. This process not only sheds light on the sensitivity of the model’s output to various inputs but also helps in debugging and refining the model by understanding pivotal input factors. In essence, these explanations provide a minimal set of changes required to a given data sample that, if applied, would change the model’s output. This minimalism helps in pinpointing the most influential data features, simplifying the often complex interplay of variables in data-driven models.

Such explanations play a crucial role in areas where understanding the decision-making process of AI systems is essential, such as in credit scoring, healthcare diagnostics, and other high-stakes environments. By enabling a clearer view of how different inputs affect outputs, counterfactual explanations foster greater trust and transparency in AI systems. They allow stakeholders to make more informed decisions about deploying AI systems in real-world scenarios, ensuring that these systems operate fairly and effectively [106].

Chapter 4

Framework for Computing Conceptual Counterfactual Explanations

4.1 Introduction

The effectiveness of Conceptual XAI methods is significantly influenced by the semantic context within which the data is interpreted [148]. This implies that the meaningfulness and interpretability of the data are crucial for these AI methodologies to function properly. Highlighting the importance of semantics [27] emphasize that "there is no explanation without semantics." They further solidify this viewpoint by providing a formal proof, which establishes that the presence of semantic elements is what distinctly separates counterfactual explanations from adversarial examples. This distinction is critical because while both counterfactual explanations and adversarial examples modify inputs to alter AI outputs, counterfactual explanations aim to provide insights into the decision-making process by illustrating how changes in input can lead to different outcomes, whereas adversarial examples typically aim to deceive the AI system. Thus, understanding and integrating the semantic context is essential for developing effective Conceptual XAI strategies.

Semantics adds crucial information to instances, but acquiring this data can be challenging. For example, in systems that identify cancer through X-rays, semantic information is only obtained after doctors annotate the images. The importance of annotations in creating conceptual counterfactual explanations was first emphasized in [84, 58, 59], which introduced the concept of an "Explanation Dataset." This dataset comprises instances accompanied by semantic annotations and serves as a foundation for generating various explanations for models, such as rule-based explanations [193], counterfactual explanations, and others. The concept and utility of the Explanation Dataset have been extensively discussed in [84, 58], including a brief overview of its background and the formalization process used.

The Explanation Dataset was introduced within the formalism of Description Logics (DLs) [14]. It establishes specific assumptions regarding the structure of Description Logics (DL) knowledge bases. To elaborate, it defines a vocabulary $\mathcal{V} = \langle \mathsf{CN}, \mathsf{RN}, \mathsf{IN} \rangle$, where $\mathsf{CN}, \mathsf{RN}, \mathsf{IN}$ are finite, mutually exclusive sets encompassing concept, role, and individual names. Within this context, one conceives $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ as a knowledge base, where the ABox $\mathcal{A}$ consists of assertions in the form of $C(a)$ and $r(a, b)$, with $C \in \mathsf{CN}$, $r \in \mathsf{RN}$, and $a, b \in \mathsf{IN}$. Simultaneously, the TBox $\mathcal{T}$ comprises terminological axioms, taking the form of $C \sqsubseteq D$ with $C, D \in \mathsf{CN}$ or $r \sqsubseteq s$ with $r, s \in \mathsf{RN}$. Here, the symbol '$\sqsubseteq$' signifies inclusion or subsumption. For instance, within this framework, a concept name (in CN) could be denoted as Dog, an individual name (in IN) might represent a specific dog, such as Lassie, and a role name (in RN) could define a relation, such as "eating". Consequently, an ABox could contain an assertion like $\mathsf{Dog}(\mathsf{Lassie})$, signifying that Lassie is characterized as a Dog, while a TBox could include the axiom $\mathsf{Dog} \sqsubseteq \mathsf{Animal}$, conveying that all dogs are categorized as animals (with Animal also being a concept name in CN).

In such a knowledge base, both the ABox and the TBox can be represented as labeled graphs. To elaborate, an ABox $\mathcal{A}$ can be denoted as the graph $\langle V, E, \ell_V, \ell_E \rangle$ (referred to as an "ABox graph"). In this context, $V = \mathsf{IN}$ represents the set of nodes, $E = \{\langle a, b \rangle \mid r(a, b) \in \mathcal{A}\} \subseteq \mathsf{IN} \times \mathsf{IN}$ signifies the collection of labeled edges, $\ell_V : V \rightarrow 2^{\mathsf{CN}}$ with $\ell_V(a) = \{C \mid C(a) \in \mathcal{A}\}$ serves as the node labeling function, and $\ell_E : E \rightarrow 2^{\mathsf{RN}}$ with $\ell_E(a, b) = \{r \mid r(a, b) \in \mathcal{A}\}$ functions as the edge labeling function.

Meanwhile, a TBox $\mathcal{T}$ that exclusively contains hierarchies of concepts and roles can be represented as a directed graph $\langle V, E\rangle$ (referred to as a "TBox graph"). In this scenario, $V = \mathsf{CN} \cup \mathsf{RN} \cup \{\top\}$ constitutes the set of nodes. The set of edges E encompasses an edge for each axiom in the TBox, in addition to edges originating from atoms that solely appear on the right side of subsumption axioms and atoms that do not appear in the TBox, all leading to the $\top$ node. More formally, this is expressed as:

$$E = \{\langle a, b\rangle \mid a \sqsubseteq b \in \mathcal{T}\} \cup \{\langle a, \top\rangle \mid c \sqsubseteq a \in \mathcal{T} \wedge a \sqsubseteq d \notin \mathcal{T} \wedge c, d \in \mathsf{CN} \cup \mathsf{RN}\} \cup \{\langle a, \top\rangle \mid a \notin sig(\mathcal{T})\}.$$

It's important to note that this notation employs an overloaded symbol $\top$, which represents both the universal concept and the universal role. Lastly, we define classifiers as functions $F : \mathcal{D} \rightarrow \mathcal{C}$, where $\mathcal{D}$ denotes the domain of the classifier, and $\mathcal{C}$ comprises the set of class names.

The initial step in attempting to comprehend a black box system involves the crucial decision of selecting the data to input into it. In this study, we delve into the advantages of supplying it with data that is enriched by information readily available in a knowledge base. This data is represented as what we refer to as "exemplars," which are essentially individuals described within the underlying knowledge and can be mapped to the feature space used by the classifier. Gathering this semantic information to characterize exemplars can be accomplished through various means: it can be sourced from publicly available knowledge graphs on the internet (such as WordNet [243]), extracted using knowledge extraction techniques (like scene graph generation), or ideally, provided by domain experts.

To illustrate, consider a motivating example where a collection of X-ray images has been meticulously annotated by medical professionals. These annotations, using standardized medical terminology, have been translated into a description logics knowledge base. Possessing such a set of exemplars empowers us to furnish explanations grounded in the underlying knowledge rather than being confined solely to the classifier's features.

In this study, we delve into the advantages of supplying it with data that is enriched by information readily available in a knowledge base. This data is represented as what we refer to as "exemplars," which are essentially individuals described within the underlying knowledge and can be mapped to the feature space used by the classifier. Gathering this semantic information to characterize exemplars can be accomplished through various means: it can be sourced from publicly available knowledge graphs on the internet (such as WordNet [243]), extracted using knowledge extraction techniques (like scene graph generation), or ideally, provided by domain experts.

Thus, the definition of the Explanation Dataset as presented in [84, 58] is the following:

Definition 1 (Explanation Dataset [84, 58])**.** *Let $\mathcal{D}$ be a domain of item feature data, $\mathcal{C}$ a set of classes, and $\mathcal{V} = \langle \mathsf{IN}, \mathsf{CN}, \mathsf{RN}\rangle$ a vocabulary such that $\mathcal{C} \cup \{\mathsf{Exemplar}\} \subseteq \mathsf{CN}$. Let also $\mathsf{EN} \subseteq \mathsf{IN}$ be a set of* exemplars*. An* explanation dataset *$\mathcal{E}$ in terms of $\mathcal{D}$, $\mathcal{C}$, $\mathcal{V}$ is a tuple $\mathcal{E} = \langle \mathcal{M}, \mathcal{K}\rangle$, where $\mathcal{M} : \mathsf{EN} \rightarrow \mathcal{D}$ is a mapping from the exemplars to the item feature data, and $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ is a DL knowledge base over $\mathcal{V}$ such that $\mathsf{Exemplar}(a) \in \mathcal{A}$ iff $a \in \mathsf{EN}$, the elements of $\mathcal{C}$ do not appear in $\mathcal{K}$, and $\mathsf{Exemplar}$ and the elements of EN do not appear in $\mathcal{T}$.*

In this chapter, we introduce an efficient algorithm that builds upon these definitions and offers counterfactual explanations within the context of knowledge graphs. Within this framework, we also present an algorithm for generating such explanations, making certain assumptions about the underlying knowledge.

4.2 Counterfactual Explanations in Terms of the Explanation Dataset

Intuitively, an explanation dataset comprises items for which we possess readily available semantic information, coupled with a feature representation suitable for input to the classifier. To distinguish these individuals that can be mapped by $\mathcal{M}$ to the classifier's domain, we introduce the concept name $\mathsf{Exemplar}$. Notably, $\mathsf{Exemplar}$ is deliberately excluded from the TBox to prevent potential complexities that could arise from reasoning processes. Given that counterfactual explanations aim to address the question of "What changes are necessary for a data sample to be classified into class B rather than class A," they often manifest in the form of *"input edits."* Within this context, counterfactual explanations take on the shape of what we refer to as *"semantic edits"*, which are employed on an ABox linked to an explanation dataset. To clarify, when we have an exemplar and a target class in mind, our objective is to identify a series of modifications that, when implemented on the ABox, result in the exemplar becoming *indistinguishable* from any other exemplar classified within the desired class.

Definition 2 (Counterfactual Explanation [58])**.** *Let $F : \mathcal{D} \to \mathcal{C}$ be a classifier and $\langle \mathcal{M}, \mathcal{K} \rangle$ an explanation dataset where $\mathcal{M} : \mathsf{EN} \to \mathcal{D}$ is a mapping function,* EN *is a set of exemplars and $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ is a knowledge base. A* counterfactual explanation *for an exemplar $a \in \mathsf{EN}$ and class $C \in \mathcal{C}$ is a tuple $\langle c, E \rangle$ where $c \in \mathsf{EN}$ and $F(\mathcal{M}(c)) = C$, and E is a set of* edit operations *that when applied on the connected component of a on the ABox graph make it equal to the connected component of c. An edit operation on an ABox can be any of:*

- *Replacement of assertion $D(a)$ with $E(a)$, symbolized $e_{D \to E}$*
- *Replacement of $r(a, b)$ with $s(a, b)$, symbolized $e_{r \to s}$*
- *Deletion of $D(a)$ or $r(a, b)$, symbolized $e_{D \to \top}$ or $e_{r \to \top}$*
- *Insertion of $D(a)$ or $r(a, b)$, symbolized $e_{\top \to D}$ or $e_{\top \to r}$*

where $D, E \in \mathsf{CN}$ and $r, s \in \mathsf{RN}$.

For example, consider an image classifier F that classifies to the classes $\mathcal{C} = \{\mathsf{WildAnimal}, \mathsf{DomesticAnimal}\}$, and two exemplars e_1, e_2 each classified to a different class: $F(e_1) = \mathsf{WildAnimal}$ and $F(e_2) = \mathsf{DomesticAnimal}$. The connected components of each exemplar in the ABox graph might be:

$$\mathcal{A}_{e_1} = \{\mathsf{Exemplar}(e_1), \mathsf{depicts}(e_1, a), \mathsf{depicts}(e_1, b), \mathsf{isIn}(a, b), \mathsf{Animal}(a), \mathsf{Forest}(b)\}$$

$$\mathcal{A}_{e_2} = \{\mathsf{Exemplar}(e_2), \mathsf{depicts}(e_2, c), \mathsf{depicts}(e_2, d), \mathsf{isIn}(c, d), \mathsf{Animal}(c), \mathsf{Bedroom}(d)\}$$

Then an explanation for exemplar e_1 and class $\mathsf{DomesticAnimal}$ would be the replacement of assertion $\mathsf{Forest}(b)$ with $\mathsf{Bedroom}(b)$, which would be symbolized $\langle e_2, \{e_{\mathsf{Forest} \to \mathsf{Bedroom}}\} \rangle$ and it would be interpreted by a user as "If image e_1 depicted animal a in a $\mathsf{Bedroom}$ instead of a Forest, then the image would be classified as a $\mathsf{DomesticAnimal}$". Of course there is no way to know if the image e_1 with the Forest replaced with a $\mathsf{Bedroom}$ would be classified to the target class, because we do not have a way to edit the pixels of the image and feed it to the classifier. The explanation however provides useful information to the user and can potentially aid in the detection of biases of the classifier. For example, after viewing this explanation, the user might choose to feed the classifier images depicting wild animals in bedrooms to see whether or not they are misclassified as domestic animals.

Global Edits

To offer the end user more comprehensive insights, we can aggregate counterfactual explanations for multiple exemplars aiming to transition to a desired class. This allows us to present statistical information regarding the alterations that tend to influence the classifier's prediction, constituting a sort of "global" explanation.

For example, one could ask "What are the most common semantic edits that when applied on exemplars depicting bedrooms lead to them to be classified as wild animals?". To achieve this, we begin by computing the multiset $\mathcal{G}$, which comprises all counterfactual explanations derived from each exemplar within the source subset transitioning to the target class. Subsequently, we present the user with the *importance* of each atom in terms of its impact on changing the prediction from the source exemplars to the target class.

This importance is quantified as follows:

$$\mathsf{Importance}(y) = \frac{|\{e_{x \to y} \in \mathcal{G}\}| - |\{e_{y \to x} \in \mathcal{G}\}|}{|\mathcal{G}|}$$

where , $x, y \in \mathsf{CN}$, or $x, y \in \mathsf{RN}$.

Here, x and y can both belong to either CN (concept names) or RN (role names), and the formula calculates the significance of each atom based on the frequency of transitions from x to y compared to transitions from y to x, normalized by the size of the multiset G.

In essence, the significance of an atom reflects the frequency with which it is incorporated into semantic edits within a collection of counterfactual explanations. A negative significance would suggest that the atom tends to be omitted, either through replacement or by the deletion of assertions.

For instance, one can assemble a group of exemplars classified as $\mathsf{WildAnimal}$ and their respective counterfactual explanations aimed at transitioning to the target class $\mathsf{DomesticAnimal}$. From this, we can calculate the significance of the presence (or absence) of a concept or a role, shedding light on their role in distinguishing between these two classes.

Mathematically, the above procedure can be expressed by introducing the following definitions.

Definition 3 (Regional of Explanation Dataset)**.** *Let* CN *be a set of concept names,* $\mathcal{Q}$ *be a set of concepts* $\mathcal{Q} \subseteq \mathsf{CN}$*, and* $D = \{x_i, C_i\}$ *be an explanation dataset. A* ***region*** *of* D *with description* $\mathcal{Q}$ *is the subset* $R_{\mathcal{Q}} \subseteq D$ *of the explanation dataset for which:* $(x_i, C_i) \in R_{\mathcal{Q}} \iff \forall c_1 \in \mathcal{Q}, \exists c_2 \in C_i : c_2 \sqsubseteq c_1$

A region within an explanation dataset represents a subset that meets specific constraints, functioning essentially as a query. For instance, consider a region description $\mathcal{C} = \{\mathsf{Animal}\}$; then the region $R_{\mathcal{Q}}$ will include any samples (x_i, C_i) from the explanation dataset where their semantic description C_i contains any concept c that falls under the category Animal as defined by the TBox.

Global counterfactual explanations are then derived as statistical measures across all optimal local counterfactual explanations from the elements within a region. These explanations specifically assess the frequency of concept introduction (through replacement or insertion) and calculate the frequency of their removal.

Definition 4 (Global Counterfactual Explanation)**.** *Let* $R_{\mathcal{Q}}$ *be a region of an explanation dataset, and* $E_{R_{\mathcal{Q}}}$ *be the multi set containing the labels of optimal local counterfactual explanations from each element of* $R_{\mathcal{Q}}$ *to the desired class. Given a set of concepts* $\mathcal{C} \subseteq \mathsf{CN}$*, a* ***global counterfactual explanation*** *is an assignment of importance to every concept* $\mathsf{C} \in \mathcal{C}$*, where the importance of a concept* C *is defined as:* $\frac{|\{e_{x \to \mathsf{C}} \in E_{R_{\mathcal{Q}}}\}| - |\{e_{\mathsf{C} \to x} \in E_{R_{\mathcal{Q}}}\}|}{|R_{\mathcal{Q}}|}$*, where* $x \in \mathsf{CN}$

Consider an explanation dataset for a classifier that decides whether an image depicts a **bedroom** or a **veterinarian's office**. A segment of this dataset, described by $\{\mathsf{Animal}\}$, includes three elements: $(x_1, \{\mathsf{Cat}, \mathsf{Dog}\}), (x_2, \{\mathsf{Insect}\}), (x_3, \{\mathsf{Human}, \mathsf{Sofa}\})$. The classifier identifies the first image as a "veterinarian's office" and the other two as **bedrooms**. The optimal local counterfactual explanations for each element to the class **veterinarian's office** might be: $E_1 = \emptyset$ (since x_1 is already classified as desired), $E_2 = \{e_{\top \to \mathsf{Human}}, e_{\mathsf{Insect} \to \mathsf{Cat}}\}$, and $E_3 = \{e_{\mathsf{Human} \to \mathsf{Cat}}, e_{\mathsf{Sofa} \to \top}\}$. The collection $E_{R_{\mathcal{C}}}$, which contains all labels from the optimal counterfactual explanations, will be $E_{R_{\mathcal{C}}} = \{e_{\top \to \mathsf{Human}}, e_{\mathsf{Insect} \to \mathsf{Cat}}, e_{\mathsf{Human} \to \mathsf{Cat}}, e_{\mathsf{Sofa} \to \top}\}$. Consequently, a generalized counterfactual explanation for this segment would be: a) Cat with importance $\frac{2}{3}$, b) Insect and Sofa with importance $-\frac{1}{3}$, and c) Human with importance 0. A negative importance implies that the concept is usually removed, whereas a positive importance indicates its introduction.

4.3 Algorithm for Computing Conceptual Counterfactual Explanations using only Concepts

Unfortunately, computing the graph edit distance is NP-hard [401], and even though there are optimized algorithms for its computation [3], it will not be feasible for explanation datasets with a large number of exemplars. Therefore, the first approximation we propose to efficiently calculate semantic counterfactual explanations is to remove the information on the edges and transform the problem from calculating the graph edit distance to calculating the simpler set edit distance. Thus, an instance that contains a *Cat on* a *Couch* will be described only by the objects *Cat* and *Couch*.

Additionally, to generate the proposed counterfactual explanations in a practical setting, the use of a classifier denoted as F, an explanation dataset D, and a TBox T are required. The process of computing explanations involves three key steps. Initially, a graph, as defined in Definition 1, is constructed. Subsequently, suitable paths within this graph are identified, and finally, for generalized counterfactual explanations, these paths are aggregated to calculate their importance based on Definition 8. An overview of the graph creation process is depicted in Algorithm 1.

The concept distance between two concepts is determined by identifying the shortest path on an undirected TBox graph. This calculation employs Dijkstra's algorithm, which operates with a complexity of $O(|\mathsf{CN}| + |T| \log |\mathsf{CN}|)$.

In calculating the Concept Set Edit Distance, as outlined in Definition 4, from a set of concepts $\mathcal{A}$ to a set of concepts $\mathcal{B}$, common elements are first removed from both sets. A bipartite graph is then established in $O(|\mathcal{A}||\mathcal{B}|)$ time, where each element of $\mathcal{A}$ is connected to all elements of $\mathcal{B}$ with edges weighted by their concept distances. The minimum weight full matching of this bipartite graph is computed using an implementation of Karp's algorithm, achieving a time complexity of $O(|\mathcal{A}||\mathcal{B}| \log |\mathcal{B}|)$.

The graph required for generating counterfactual explanations is created using Algorithm 1, with the total time for creation being $O((n + t \log n)m^4k^2 \log m)$, where n represents the size of $|\mathsf{CN}|$, m the maximum cardinality of a set of concepts, k the size of the explanation dataset, and t the size of the TBox. This graph creation is performed only once per explanation dataset and TBox.

For the computation of local counterfactual explanations as per Definition 6, the already constructed graph, inclusive of edge costs and labels, is utilized. Dijkstra's algorithm is again employed to ascertain the shortest path.

To effectively tackle the edit distance problem, it is critical first to ascertain the cost associated with each modification. Our primary goal is to develop counterfactual explanations that not only retain but closely mirror the original exemplars in terms of semantic content. This necessitates that the cost of an edit accurately represents the extent of semantic alteration inflicted upon the exemplar post-edit. Moreover, it is paramount that these edits maintain a level of transparency, ensuring that explanations provided to users are both comprehensive and easily comprehensible. For example, while the proximity of concepts might be quantifiable through their embedded representations within a word embedding system or a graph neural network, such metrics may fail to clarify the rationale behind the perceived closeness or distance of these concepts. Thus, the methodologies employed in these calculations should be explicit and clear.

In our approach, we harness the informational capacity of the TBox. Specifically, when addressing the first type of ABox edits, which involve replacing one concept assertion with another $e_{A \to B}$, we determine the cost of substituting concept A for concept B based on their proximity within the TBox graph. This calculation is done without considering the directionality of the edges in the graph. For instance, if we consider a specific TBox setup:

$$\mathcal{T} = \{\mathsf{Cat} \sqsubseteq \mathsf{Mammal}, \mathsf{Dog} \sqsubseteq \mathsf{Mammal}, \mathsf{Ant} \sqsubseteq \mathsf{Insect}, \mathsf{Mammal} \sqsubseteq \mathsf{Animal}, \mathsf{Insect} \sqsubseteq \mathsf{Animal}\}$$

the cost of replacing a $\mathsf{Cat}(a)$ assertion with $\mathsf{Mammal}(a)$ would be 1, the cost of replacing $\mathsf{Cat}(a)$ with $\mathsf{Dog}(a)$ would be 2, and the cost of replacing $\mathsf{Cat}(a)$ with $\mathsf{Ant}(a)$ would be 4. Similarly, the cost of replacing a role assertion $r(a, b)$ with $s(a, b)$ (symbolized $e_{r \to s}$) is assigned as the shortest path distance on the undirected TBox graph from r to s. It is important to note that this may not necessarily be the optimal method for computing semantic similarities of concepts and roles, as other measures exist in the literature [49].

In the process of adding/removing concept, represented by the notation $e_{\top \to a}$, costs are determined based on how far the inserted atom (a concept or a role) is from the $\top$ node within the TBox graph. This structure implies that inserting atoms that are more specific incurs a higher cost compared to those that are more general. Similarly, when removing atoms, as denoted by $e_{a \to \top}$, the cost also depends on the proximity of the concept or role being deleted to the $\top$ node in the undirected TBox graph. Thus, the deletion of more specific concepts and roles is more costly.

Moreover, the system allows for the manual adjustment of costs by users, which can be particularly useful in applications where certain modifications are impractical or impossible in real-life scenarios. For instance, in cases where exemplars represent individuals and concepts symbolize their age groups, such as Young and Old, the edit $e_{\mathsf{Old} \to \mathsf{Young}}$ might be deemed unrealistic, as it would necessitate an impossible reversal of time. This type of constraint is commonly referred to as an "actionability constraint" [322, 237]. In such scenarios, an

Algorithm 1: Explanation Graph Construction

```
Data: A classifier $F$, an explanation dataset $D$, an undirected TBox Graph $G_T$
Result: Explanation Graph $G_E$
//the explanation graph will have a node for each element in the explanation dataset
Initialize Directed Graph $G_E = (V_E = D, E_E = \emptyset)$;
foreach $(x_i, C_i) \in D$ do
    foreach $(x_j, C_j) \in D \setminus \{(x_i, C_i)\}$ do
        Initialize Graph $G_C = (V_C = C_i \cup C_j, E_C = \emptyset)$;
        foreach $k \in C_i$ do
            foreach $l \in C_j$ do
                //Compute concept distance using TBox graph
                $d_T(k, l) = |\mathsf{ShortestPath}(G_T, k, l)|$
                //Add an edge to $G_C$ with weight $d_T$
                $E_C = E_C \cup \{(k, l, d_T)\}$
            end
        end
        //Compute minimum weight full matching of the bipartite graph $G_C$
        $\{(c_m, c_n)\}, w = \mathsf{MinFullMatch}(G_C)$
        //Concept Set Edit Distance
        $D_T(C_i, C_j) = w$
        //Compute inverse significance
        $\frac{1}{\sigma(i,j)} = \frac{D_T(C_i,C_j)}{|F(x_i)-F(x_j)|}$
        //Add an edge to the explanation graph $G_E$ with weight $\frac{1}{\sigma}$ and as a label the edits corresponding to the
          minimum weight full match
        $E_E = E_E \cup \{(v_i, v_j, \frac{1}{\sigma(i,j)}, \{e_{c_m \to c_n}\})\}$
    end
end
return $G_E$
```

infinite cost may be assigned to discourage or prohibit specific edits. This ability to flexibly assign costs adds a practical layer of adaptability to the model. It ensures that the model can accommodate specific real-world constraints and requirements, guaranteeing that all generated edits remain actionable.

Additional Criteria for Good Counterfactuals

In the context of this framework, the simplest counterfactual explanation for an exemplar e and a target class C would be the exemplar x (along with the edits) that is the closest with respect to edit distance to e while considering exemplars that are classified to C. If we have access to the output probabilities of the classifier for each class, then we can utilize this information and provide additional criteria to determine which counterfactual explanations to show to a user.

Target Significance

The first additional criterion, defined as *significance* in [84], is to find the exemplar x that **maximizes** the fraction:

$$target\_significance = \frac{P_C(x)}{\mathsf{edit\_distance}(e, x)} \tag{4.3.1}$$

, where $P_C(x)$ is the probability for exemplar x to be classified to target class C. Intuitively, we are searching for a small set of low-cost edits (minimize edit_distance) that largely effect the output of the classifier for the desired class C (maximize $P_C(x)$).

Source-Target Significance

Another option for a criterion would be to also take under consideration the prediction probability for the class that the original exemplar is classified to. Similarly to before, a counterfactual for exemplar e would be exemplar x (along with the edits) that **maximizes** the fraction:

$$source\_target\_significance = \frac{P_C(x) - P_D(x)}{\mathsf{edit\_distance}(e, x)} \tag{4.3.2}$$

, where D is the class that e is classified to. Counterfactual explanations are supposed to answer the question "Why class **D** and not class **C**?", and while the previous criteria emphasize the "...and not class **C**" part of the question, intuitively source-target significance puts more weight on the "Why class **D**" part.

Entropy

A final criterion we explore in this work is to consider the *confidence* of the classifier for classifying an exemplar to the target class C. As a measure of confidence we use the entropy at the output of the classifier, where a lower value indicates a more confident prediction. To do this, we find exemplar x that is classified to target class C and **maximizes** the fraction:

$$entropy = \frac{\sum_{i \in \mathcal{C}} P_i(x) \log P_i(x)}{\mathsf{edit\_distance}(e, x)} \tag{4.3.3}$$

, where $\mathcal{C}$ is the set of classes of the classifier.

4.4 Evaluation

Our research is dedicated to exploring counterfactual explanations with the aim of uncovering biases in classifiers across various domains. To demonstrate the versatility and broad applicability of our proposed methodology, we plan to expand our study into two distinct domains: image classification and audio classification. In the realm of image classification, we seek to refine the interpretability of visual data processing. More critically, in the domain of audio classification, we intend to focus on medical applications where the provision of precise and reliable explanations is essential. By extending our methodology to these areas, we aim to enhance the understanding and reliability of classifier decisions in fields where accuracy is paramount.

4.4.1 Evaluation in Image

In the domain of image classification, we began our evaluation process by leveraging the CLEVR-Hans3 dataset [331], which provides a controlled environment with known inherent biases. During this initial phase, we trained a classifier on images where a "grey cube" appeared consistently within a specific class. Our objective was to ascertain whether the classifier would recommend adding a "grey cube" to an image to categorize it accordingly, thereby uncovering any intrinsic biases. The results confirmed that our counterfactual algorithm successfully identified these biases. Subsequently, we demonstrated the practical application of our framework on a black-box classifier trained with the Places dataset [419]. For this demonstration, we utilized semantic information from multiple sources, including COCO [198], and WordNet [75]. This aspect of our research illustrated how our approach could intelligently navigate and make sense of complex datasets by integrating various semantic inputs. This integration enhances the transparency and comprehensibility of the AI's classification decisions, highlighting the robustness and adaptability of our methodology in practical scenarios.

Explaining a CLEVR-Hans3 Classifier

Setting Our experimental approach begins with highly controlled datasets and progressively incorporates scenarios that mirror real-world complexities. In this vein, we utilized the CLEVR-Hans3 dataset, featuring images of colored 3D geometrical objects categorized into three distinct classes. Each image in this dataset provides detailed information about the objects present, including their shape (Sphere, Cube, Cylinder), size

(Large, Small), material (Metallic, Rubber), and color (Blue, Yellow, Brown, Grey, Green, Purple, Cyan, Red). The dataset delineates classes based on specific combinations of these attributes:

- **Class A**: includes a Large Grey Cube and a Large Cylinder,
- **Class B**: comprises a Small Metal Cube and a Small Sphere,
- **Class C**: contains a Large Blue Sphere and a Small Yellow Sphere.

Importantly, the first two classes exhibit intentional biases in the training set, where, for example, the Large Cube in Class A is always Grey, and the material of the Small Sphere in Class B is always Metal, though these attributes vary randomly in the test set.

To evaluate our enriched counterfactual analysis system against the FACE algorithm, which calculates conceptual counterfactuals with a focus on actionability and feasibility constraints [322] and operates exclusively on training set images, we created two distinct explanation datasets. The first is restricted to training set images to facilitate this comparison, while the second includes test set images to detect biases ingrained during training. For our classifier, we employed a ResNet34 model [113], which achieved 99% accuracy on the confounded training set but showed diminished performance on the test set, particularly in the confounded classes (F1 scores were Class A: 0.27, Class B: 0.54, Class C: 0.92), as anticipated.

To enhance our explanation capability, we defined a concept for each combination of shape, size, material, and color (including the absence of any attribute), resulting in a total of 324 distinct concepts. We further developed a terminological box (TBox), adding an inclusion axiom from each concept to any other concept sharing a similar description but missing one element. For instance, the concept GrayCube is subsumed by Gray and Cube. This ontological structure allows us to assign concept sets to each element in the dataset, drawing on the detailed descriptions provided in the corresponding JSON files. This comprehensive approach underscores our commitment to advancing the transparency and accuracy of AI classifiers through sophisticated counterfactual explanations.

Results

Local Counterfactuals In Figure 4.4.1, we present local counterfactual explanations for three images randomly selected from those originally classified in Class B (featuring a Small Metal Cube and Small Sphere, where the Small Sphere is consistently Metal in the training set), targeting reclassification into Class A (characterized by a Large Cube and Large Cylinder, with the Large Cube consistently Grey in the training set). The second column of the figure displays suggestions from the FACE algorithm, and the third column shows suggestions from our algorithm. At first glance, the results from both algorithms may not seem particularly intuitive. We believe this stems from the nature of the explanations, which are sequences of samples from the training set.

Upon closer examination, it becomes evident that our method typically maintains a consistent number of objects per image. This consistency is largely due to the high costs associated with adding or removing concepts, as opposed to merely replacing them. In contrast, FACE, which is based on the overall distribution of the dataset and operates at the pixel level without recognizing the objects, tends to suggest transitions to images that include a larger number of objects.

Global Counterfactuals For our initial global counterfactual explanation, we focused on images from the CLEVR-Hans3 test set that were classified as Class B. We analyzed the modifications our system recommended to reclassify these images as Class A. We identified that frequently recurring changes carry significant weight in delineating the transition between these classes, serving as indicators of key class characteristics. Positive importance in our analysis suggests the addition of features, whereas negative importance implies their removal.

Our findings promptly highlighted the classifier's bias towards the confounded Class A. As previously discussed, a notable bias in Class A is that the Large Cube is invariably Grey in the training set. This bias was evident in our results, particularly seen in the first three bars of figure 4.4.2, where the most critical insertions included the concepts: (*Gray*, *GrayLargeCube*, *GrayLarge*), rather than simply (*LargeCube*).

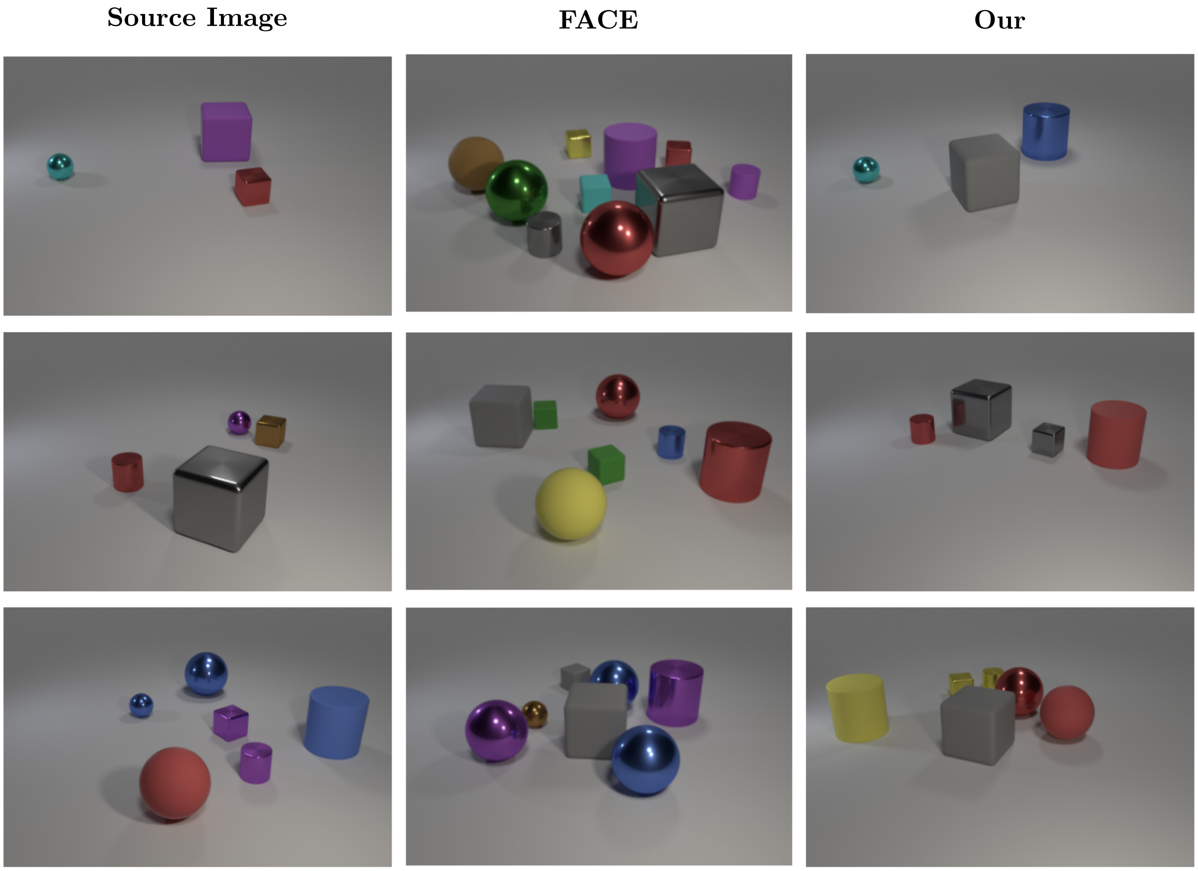


Figure 4.4.1: Counterfactuals for 3 images (first column) which classified in class B with target class A, using FACE (second column) and our proposed method (third column). The first column shows the source images, the second column shows the results from FACE and the third column the results of our method.

Explaining a Places Classifier

Setting In the context of the study, it was determined that the exploration of more intuitive and realistic example was necessary; hence, the COCO dataset [199] was utilized. This dataset, comprising real-world images annotated with objects, allowed for the automatic linking of these objects to external knowledge bases such as WordNet.

During the analytical phase where COCO's labels were scrutinized to determine a class transformation strategy that would effectively utilize them, it was concluded by the researchers that the images should be categorized primarily into two classes: those related to "Restaurants" and those associated with "Bedrooms". For the restaurant-related category, the following subsets of images were collected from the COCO dataset:

1. Images featuring the combination of {dining table, person, pizza}, exceeding 1000 images.
2. Images displaying {dining table, person, wine glass}, totaling over 1200 images.

In the bedroom-related category, images were grouped based on the presence of specific labels:

1. {bed, person}, encompassing approximately 1300 images.
2. {bed, book}, with around 800 images.
3. {bed, teddy bear}, including about 300 images.

Additionally, a selection of potentially confusing images for the classifier was included. These images featured unusual combinations of COCO labels:

1. {bed, fork}, consisting of 10 images.

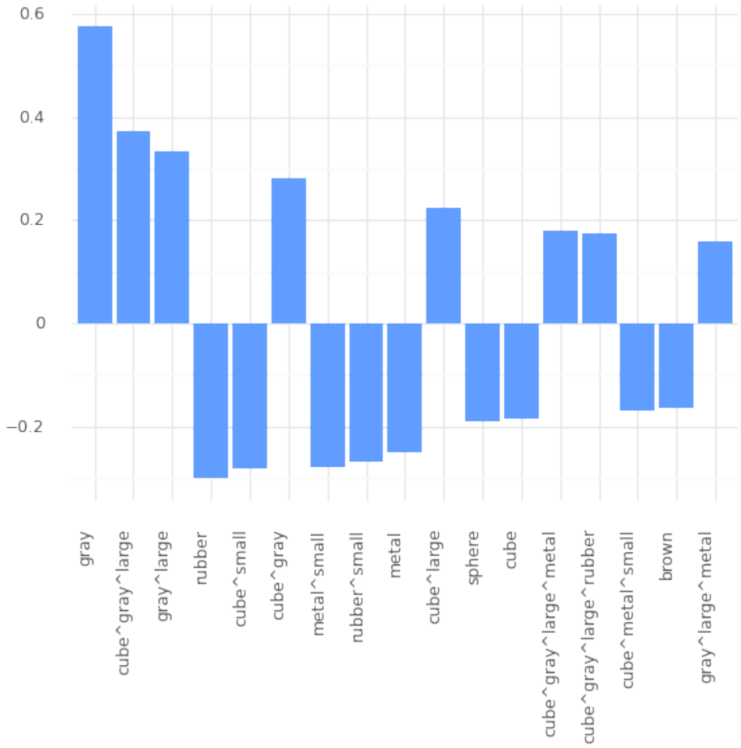


Figure 4.4.2: Global explanation for the subset of CLEVR-Hans3 which is classified in class B, with target class A.

2. {bed, spoon}, with 20 images.
3. {bed, wine glass}, also 20 images.
4. {bed, pizza}, including 10 images.
5. {dining table, bed}, which included 170 images.

For each image, a description of the objects present was provided. These descriptions were linked to WordNet synsets using the NLTK Python package[1]. WordNet synsets served as the set of concept names CN, and the hierarchical structure of hyponyms and hypernyms was utilized as a TBox in the study. A pre-trained image classifier from the PLACES dataset [420], provided by the dataset creators[2], was employed for scene classification. Predictions were then made on the aforementioned subsets of COCO images. This classifier functioned as the black-box model for which explanations were provided in the study.

Results

Local Counterfactuals As depicted in the first row of Figure 4.4.3, a counterfactual explanation is provided for an image initially classified as a "Bedroom." To alter its classification to the target class "Playhouse," only one conceptual edit is necessary, specifically the addition of a Child ($e_{\top\to\mathsf{Child}}$). This scenario is particularly intriguing as the prediction of "Playhouse" is incorrect, revealing a potential classifier bias. The classifier's tendency to categorize a "Bedroom" with a Child as a "Playhouse" suggests an erroneous associative bias. In the second row of Figure 4.4.3, another local counterfactual explanation demonstrates the transformation of an image from "Bedroom" to "Veterinarian's Office." This transformation is achieved by the addition of a Cat. The resultant image classification as a "Veterinarian's Office" is also incorrect, further highlighting possible biases in the classifier's training or logic.

[1]

[2]

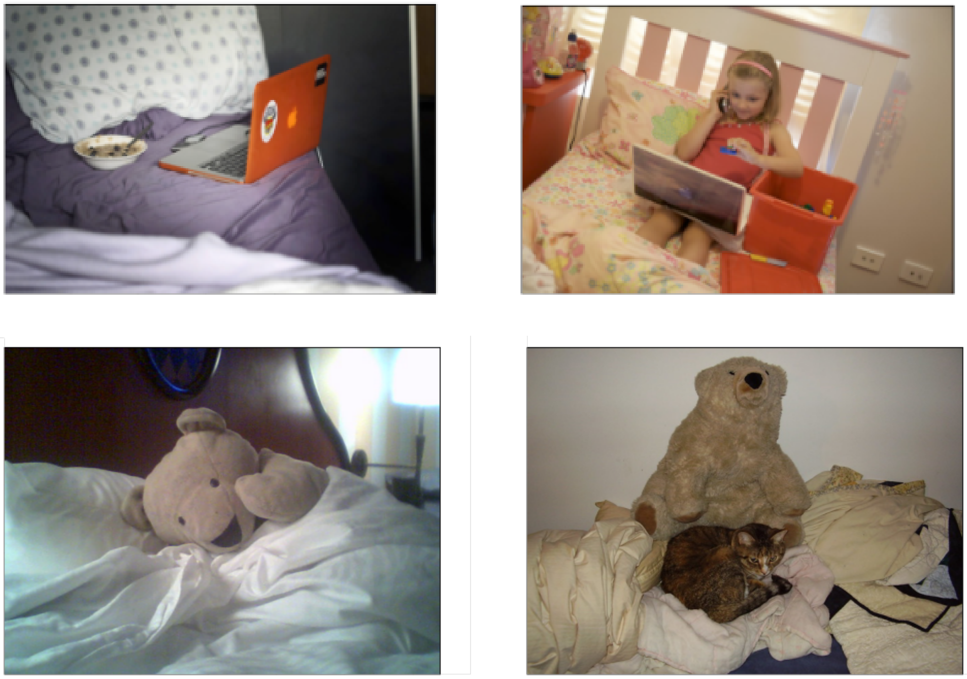


Figure 4.4.3: Counterfactual explanation for changing the prediction of the image on the left from "Bedroom" to "Playhouse" is simply to add a child ($e_{\top\to\mathsf{Child}}$) (top) and from "Bedroom" to "veterinarians office" is simply to add a cat ($e_{\top\to\mathsf{Cat}}$) (bottom).

Figure 4.4.4 presents a counterfactual explanation involving a more complex transition, with a two-step path on the graph. The original classification of the source image is a "Bedroom," with the target classification being a "Computer Room." The transformation is smoothly executed by initially adding a person (noting that the source image already contains two laptops). Subsequent additions include two more individuals and two additional laptops, effectively transforming the scene into a "Computer Room."

Global Counterfactuals In Figures 4.4.5 and 4.4.6, we observe two instances of generalized counterfactual explanations within the COCO dataset. In these figures, the numerical values on each bar indicate the significance of either inserting (positive values) or removing (negative values) certain concepts, aiding the transition from a source region within an explanation dataset to a designated target class. The exact source regions and target classes are not specified, yet they can be inferred from the most frequent concept modifications.

In the first example, Figure 4.4.5, the most frequent removals from the source images include concepts such as {furniture, bed, animal, carnivore, dog}, while the primary additions feature {home appliance, refrigerator, white goods, consumer goods}. These modifications suggest that the source region predominantly contains images of bedrooms, possibly with a pet presence, transitioning to images of kitchens. Indeed, the initial and target classes were confirmed to be "bedroom" and "kitchen," respectively.

The second example, Figure 5.2.4, illustrates that the most common removals are related to {instrumentality, artifact, electronic, furniture, telecommunications, TV, broadcasting, kitchen}, and the additions focus on {carnivore, animal, mammal, feline, cat, dog}. Given the classification context of rooms and places, one might initially speculate the source as a kitchen and the target as a location associated with domestic animals. However, the actual classifications were from "bedroom" to "veterinarian." This discrepancy prompts a notable observation: "kitchen" elements appear rather than "bed" due to the inclusion of studio-apartment bedroom images that feature partial kitchen views, which are typically absent in veterinarian offices.

It is important to highlight that these examples were not selectively chosen; rather, during our experiments, it was often possible to deduce the source region and target class by examining the frequency of edits. One

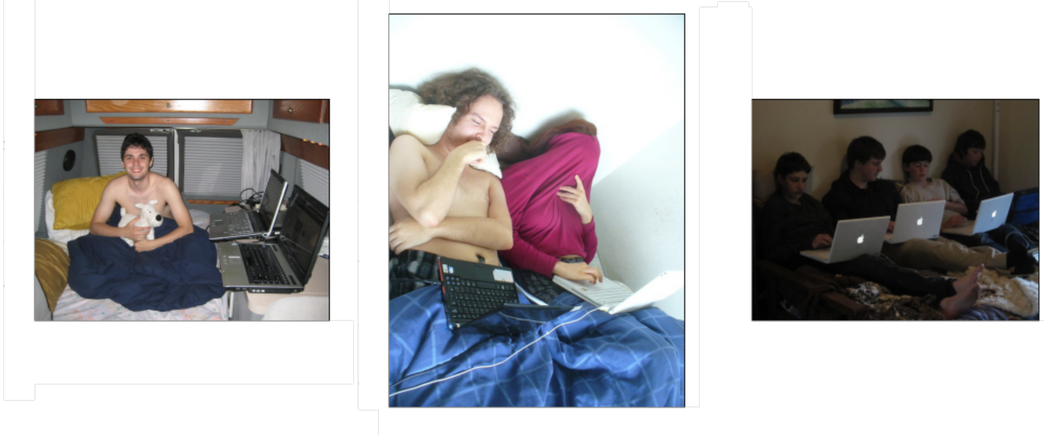


Figure 4.4.4: Counterfactual explanation for changing the prediction of the image on the left from "Bedroom" to "Computer Room", which requires two steps.

intriguing case involved the target class "computer room," where the explanation frequently suggested adding people but not laptops or computers. Upon further investigation, it was revealed that many images labeled as "computer room" in our dataset featured people in settings resembling labs, with no computers visible.

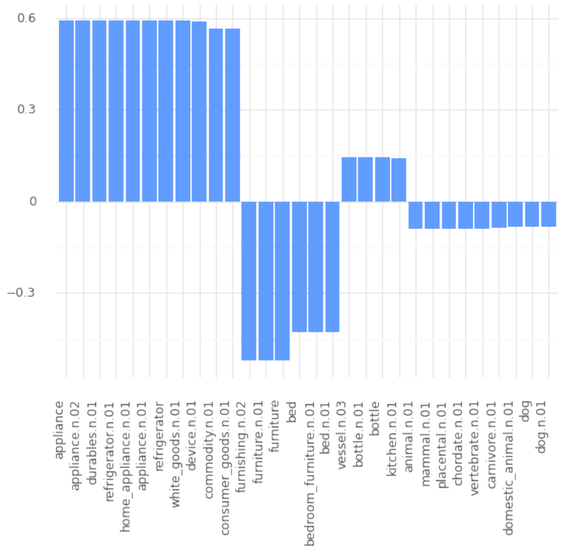


Figure 4.4.5: Global Counterfactual Explanations using as the explanation dataset the COCO which is classified as "bedroom", with the target class being "kitchen".

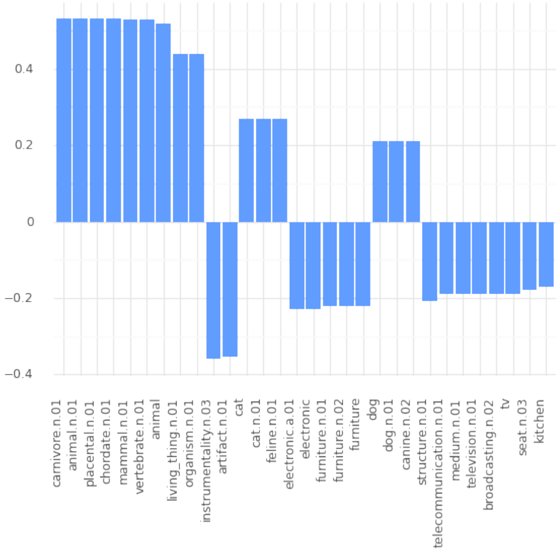


Figure 4.4.6: Global Counterfactual Explanations using as the explanation dataset the COCO which is classified as "bedroom", with target class "veterinarian".

4.4.2 Evaluation in Audio

COVID-19 Classification

Our final experiment serves as a critical extension of our analytical framework into the audio domain, emphasizing its broad applicability, particularly in high-stakes environments such as medical diagnostics. In this case, the focus is on the diagnosis of COVID-19—an area where the consequences of decisions can significantly impact patient outcomes and public health. The importance of explainability in medical applications cannot be overstated. It ensures that the decision-making processes of AI systems are transparent, allowing healthcare professionals to understand the basis of automated recommendations. This transparency is crucial for building trust between these technologies and their human users, facilitating a more informed and ethical integration of AI in healthcare. Explainability in medical diagnostics aids in identifying and correcting biases, as well as in verifying the clinical relevance of the features used by AI systems. By understanding the “why” and ‘how” behind a diagnosis, medical professionals can make better-informed decisions, potentially catching errors before they affect patient care. Moreover, explainable AI (XAI) supports compliance with regulatory requirements that are increasingly demanding transparency in automated systems used in healthcare settings.

The aim of this experiment was not only to test our framework’s efficacy in a new domain but also to demonstrate its domain-agnostic and modality-agnostic nature. These attributes highlight the flexibility and scalability of our method, showcasing its capability to generalize across different modalities beyond visual data. By applying our framework to audio inputs—specifically, to the sounds of coughing—we illustrate how it can adapt to different types of data and extract meaningful insights that are critical in a clinical context. In doing so, we conducted our study using a classifier trained on the Coswara Dataset [320], a collection recognized for its role in a major IEEE COVID-19 sensor informatics challenge [3]. The classifier evaluates audio files containing cough sounds to assess the probability of COVID-19 infection.

The explanation system utilizes the richly annotated audio database of the Smarty4covid dataset [398] to elucidate the classifier’s decision-making process by highlighting significant audio features, such as cough characteristics and other respiratory symptoms.

Despite achieving a commendable c-statistic of up to 0.764 ± 0.038 in a 5-fold evaluation with the Coswara dataset, the classifier’s performance deteriorated (with a c-statistic of less than 0.50) when tested on the Smarty4covid dataset [398]. This notable decline suggests the presence of potential biases in the dataset or the classifier’s methodology, underscoring the importance of further analysis and adaptation in diverse data environments.

This exploration into the audio domain underscores the method’s robustness and its potential to assist in diverse medical scenarios, from routine diagnostics to pandemic responses. It establishes a precedent for applying XAI in varied settings, reinforcing the adaptability and crucial role of explainability in ensuring that AI-driven tools enhance, rather than hinder, medical diagnostics. Through this initiative, we contribute to the broader discourse on the necessity of robust, transparent, and accountable AI systems in healthcare, paving the way for future innovations that adhere to both scientific rigor and ethical standards.

Setting In this experiment, we utilize explanations to interpret the decisions of a classifier that was trained using a segment of the Coswara Dataset [320]. For the classifier we selected the winning entry in the IEEE COVID-19 sensor informatics challenge. The classifier’s task is to analyze audio recordings of coughs and determine the likelihood of a COVID-19 infection in the individual. The classifier employs a sophisticated model using 2D Convolutional Neural Networks (CNN) [191] that processes audio segments converted into Mel spectrograms. These spectrograms are utilized as inputs to determine the likelihood of the sounds being coughs, breaths, or voices. The Mel spectrogram representation of the audio segments is particularly detailed, with the frequency axis having a fixed size of 128 units. The time axis size, denoted as d, is adjustable and was optimized using a grid search technique. This search ranged from 128 to 1024, corresponding to audio lengths of approximately 1 to 10 seconds, respectively. Each CNN within the system comprises several stacked blocks, each containing l convolutional layers. These layers are followed by a 2 × 2 max pooling layer and a dropout layer, with the dropout probability set to the standard rate of 0.5.

In each convolutional block, the layers are equipped with k 3×3 kernels activated by the ReLU function and utilize identical padding to maintain dimension consistency across inputs and outputs. The output from the last convolutional layer is then flattened and passed to a fully connected layer that includes 3 softmax-activated neurons, effectively categorizing the input into cough, breath, or voice. Hyperparameter tuning was rigorously performed through a grid search to find optimal settings for l (ranging from 1 to 3), k (ranging from 64 to 128), and b (ranging from 3 to $\log_2(d)$). For this purpose, 80% of the development dataset, equating to 5,855 audio recordings, was utilized for training, while the remaining 20% (1,465 recordings) served as the validation set. The architecture of the classifier, which significantly influences its accuracy and efficiency, is depicted in Figure 4.4.7.

[3] https://healthcaresummit.ieee.org/data-hackathon/

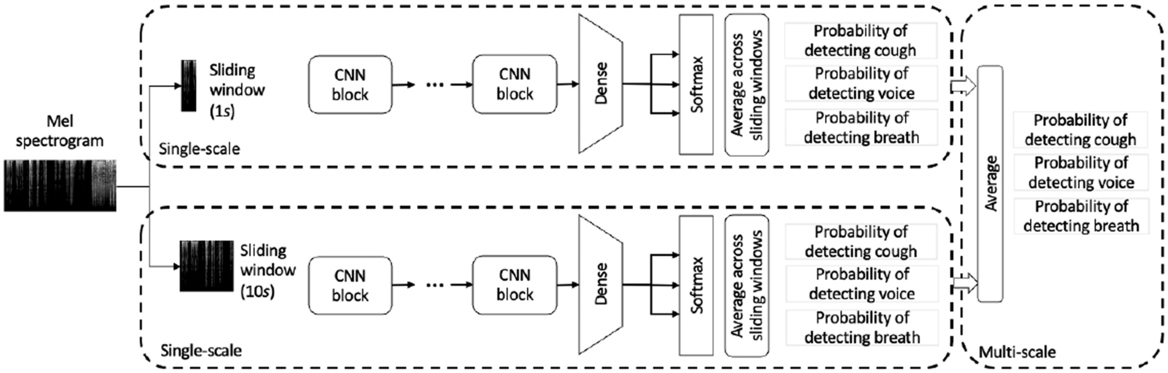

Mel spectrogram
Sliding window (1s)
CNN block
CNN block
Dense
Softmax
Average across sliding windows
Probability of detecting cough
Probability of detecting voice
Probability of detecting breath
Single-scale
Sliding window (10s)
CNN block
CNN block
Dense
Softmax
Average across sliding windows
Probability of detecting cough
Probability of detecting voice
Probability of detecting breath
Single-scale
Average
Probability of detecting cough
Probability of detecting voice
Probability of detecting breath
Multi-scale

Figure 4.4.7: Overview of the classifier used to categorize audio clips into coughs, voices, and breathing sounds [398]. Both single-scale and multi-scale methodologies are depicted.

The selection of an explanation dataset is crucial for medical applications, particularly when the dataset is expected to contribute significantly to medical diagnostics and research. There is a notable scarcity of datasets that encompass both granular data, such as images, and high-level semantic information, such as symptoms. To address this gap, we sourced our explanation dataset from the Smarty4covid platform[4], resulting in the creation of the Smarty4covid dataset [398]. This dataset includes a curated collection of audio samples that are crucial for our analyses.

To facilitate the integration of data from diverse sources, including Coughvid [265], COVID-19 Sounds [380], and Coswara, a sophisticated web-ontology language (OWL) knowledge base[5] was developed. The development of this knowledge base was essential for performing complex queries aimed at identifying users with specific attributes. The processes of crowd-sourcing, meticulous data cleaning, and systematic data labeling were integral to the creation of the smarty4covid OWL knowledge base, which is meticulously maintained alongside the associated data records within the same Zenodo Repository.

These efforts ensure that our explanation dataset is not only comprehensive and rich in both low-level and high-level information but also structured in a manner that supports advanced data analysis and research in the medical field.

The smarty4covid OWL knowledge base utilizes a structured vocabulary divided into concept names (CN), role names (RN), and individual names (IN), which are distinctly separate from each other. This architecture allows for the construction of two main components within the knowledge base: the Assertional Database (ABox) and the Terminology Database (TBox). The ABox comprises assertions of the form $C(a)$, $r(a, b)$ where C is a concept from CN, r is a role from RN, and a, b are individuals from IN. The TBox, on the other hand, contains terminological axioms formatted as $C \sqsubseteq D$, where C and D are concepts from CN, and relational hierarchies such as $r \sqsubseteq s$ where r and s are roles from RN. These axioms are foundational for establishing the hierarchies of concepts and roles within the TBox, structuring the ontology to reflect the complex relationships and characteristics of the data.

The set of individual names (IN) in the smarty4covid OWL knowledge base is meticulously curated to include unique identifiers for each participant, their questionnaires, audio files, and the healthcare professionals involved in the labeling process, along with detailed characterizations of the audio records. Additionally, IN encompasses unique identifiers for each reported symptom, COVID-19 test result, and pre-existing medical condition, directly linked to the corresponding participant and their questionnaire.

These individuals are interconnected through well-defined roles that are crucial for the operational integrity of the knowledge base. The hierarchy of these role names (RN) and their relationships are detailed in Figure 4.4.8. Each role is defined with a specific domain and range that delineate the types of entities that can be linked through these roles. For example, the role "hasCharacterization" connects audio files to their respective characterizations as labeled by healthcare professionals. Conversely, "characterizedBy" establishes

[4] https://www.smarty4covid.org
[5] https://www.w3.org/OWL

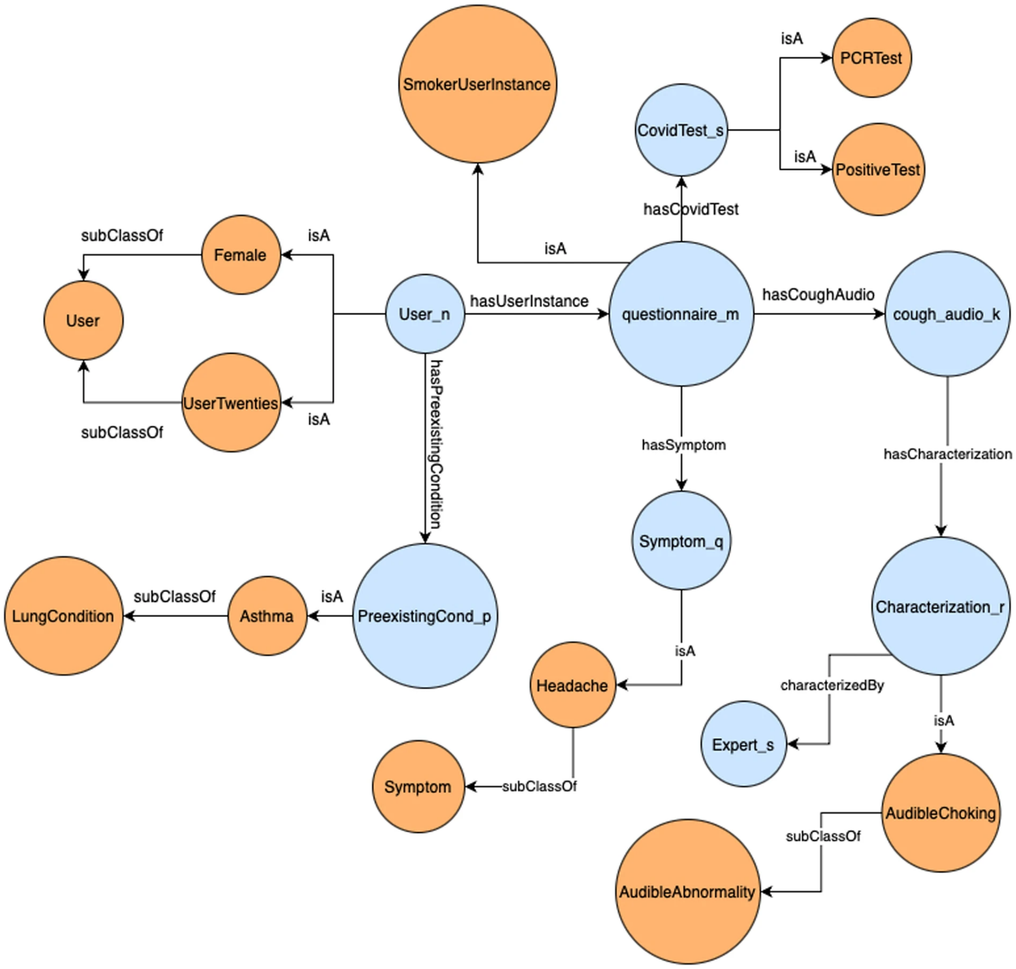


Figure 4.4.8: Example of the Smarty4covid knowledge base architecture. Blue nodes symbolize individual entities, while orange nodes depict concepts. Edges marked as IsA indicate concept assertions from the ABox, and edges labeled subClassOf denote inclusion axioms from the TBox.

links from these characterizations back to the healthcare professionals themselves. The role "hasAudio" and its subsidiary roles create links between questionnaires and corresponding audio files. The roles "hasCovidTest" and "hasSymptom" connect questionnaires to instances of COVID-19 tests and self-reported symptoms, as well as vaccination statuses, respectively. Furthermore, the role "hasPreexistingCondition" forms connections between participants and their reported pre-existing conditions, while "hasUserInstance" links participants to their submitted questionnaires. Through such structured relationships and a clear hierarchical system, the Smarty4covid OWL knowledge base serves as a critical tool for researchers and healthcare providers to navigate and utilize the rich dataset effectively, supporting advanced studies and interventions in the field of COVID-19 and respiratory illnesses. To enhance the quality of the data, multiple crowd-sourced campaigns were conducted. These efforts were aimed at ensuring the quality of the audio files and annotated results, as well as enriching the data through expert contributions.

These samples are further enriched with annotations such as gender, symptoms, and medical history, structured within an ontology to provide a comprehensive understanding of each case. It is important to highlight that in our study, we focused exclusively on audio-relevant features, intentionally omitting unrelated factors like vaccination status or travel history, to maintain a clear focus on audible symptoms.

Results The analysis of global counterfactuals, transitioning classifications from "COVID-19 Negative" to "COVID-19 Positive" (referenced in Table 4.1), revealed that the most significant addition was the concept "Symptom". This concept acts as an umbrella term encompassing all symptoms cataloged in our knowledge

Concept	Importance	Concept	Importance
Symptom	-1.298	Runny Nose	-0.22
Respiratory	-1.278	Dry Cough	-0.19
Female	0.25	Cough	-0.189
Male	-0.254	Sore Throat	-0.13

Table 4.1: Global counterfactual transitions from “COVID-19 Negative” to “COVID-19 Positive” based on a classifier trained using Coswara Dataset cough audio, with explanations derived from Smarty4covid dataset.

base. However, not all symptoms contribute equally to altering the classifier’s predictions. Notably, the concept “Respiratory”, which is a subset of “Symptom” and a precursor to specific respiratory symptoms (like “Dry Cough”, “Runny Nose”, and “Cough”), is frequently added, highlighting its relevance in the diagnosis of COVID-19.

A particularly significant finding from our experiment was the identification of an unwanted bias within the classifier, which correlated the likelihood of COVID-19 positivity with the user’s gender. This bias was uncovered through an analysis that revealed changes in the gender of the subjects as a common modification in our counterfactuals. Further investigation into the training data—specifically the Coswara dataset—revealed a disproportionate representation of COVID-19 positivity rates: 42% of the females tested were positive, compared to 27% for males. This discrepancy likely led the classifier to develop an erroneous association between gender and COVID-19 status, which we identified as a critical issue needing correction to avoid reinforcing gender-based biases in medical diagnostic processes.

This experiment underscores the importance of explainability in AI-driven healthcare applications, particularly in ensuring that machine learning models do not perpetuate existing biases and that they adhere closely to medical relevance rather than spurious correlations found in training data.

4.5 Conclusion

The experiments conducted have yielded intriguing results, demonstrating that both local and generalized counterfactual explanations are informative, understandable, and practical for application. Particularly in the CLEVR-Hans3 scenario, pre-existing biases within the classifier were successfully identified. In contrast, the exploration within the COCO dataset revealed previously unrecognized biases. For instance, it was noted that the depiction of people significantly outweighed the presence of laptops in images classified under the “computer room” category. This insight was unexpected and emphasized an assumption by the classifier that veterinarian offices would frequently depict beds among other items. Such findings underscore the potential of using high-level external terminology for explanations, which has shown to be more intuitive and relatable compared to low-level feature-based explanations, as seen when juxtaposed with the FACE algorithm. Finally, the exact same algorithm was also successfully applied in the audio domain, revealing a bias in the winning classifier of the IEEE COVID-19 challenge.

Despite the promising results, the proposed framework relies heavily on the availability of semantically annotated data, which is not widely available across all domains. To address this limitation, future efforts will focus on two main strategies. The first involves the automatic semantic annotation of data using advanced information extraction techniques such as object detection for images (see also Section 5.2.1) or linking textual content to encyclopedic knowledge [241]. The second strategy, particularly relevant for critical domains like medicine, involves investing in manually annotated and curated explanation datasets by domain experts, which could enhance the reliability and user trust in generated explanations.

Looking ahead, the framework will be further developed to incorporate Description Logics, including roles and individuals, and to accommodate more complex axioms within the TBox. This expansion is expected to enrich the theoretical and practical outcomes of the counterfactual explanations generated. Experiments will also extend to different types of data, including text and tabular data, with the potential incorporation of human evaluators to enhance the assessment process.

Future research will delve into the properties of explanation datasets as defined within the current framework and as explored in related works [192, 57]. An exploration into the effects of the size of an explanation dataset, such as utilizing the entire COCO dataset, is planned. Additionally, the use of the same explanation dataset linked to alternative TBoxes (for example, ConceptNet instead of WordNet) will be explored, necessitating further experiments on different conceptual or semantic distances.

Further investigation will explore the application of the proposed method in diverse setups beyond exporting counterfactual explanations. Notably, this method could be employed to assess the efficacy of generative systems, such as those converting text to images. Additionally, the utility of this approach in story generation systems is noteworthy [270, 346, 218], where it can be used to measure the consistency between generated narratives and corresponding visual outputs. The accuracy of this alignment can be quantitatively assessed using the proposed methodology. Furthermore, the reciprocal relationship between image and text generation systems warrants examination. This includes measuring the frequency of inaccuracies in image comprehension or text production by evaluating the rate of hallucinations [103, 268, 94] within these systems, facilitated by an explanatory dataset.

Chapter 5

Conceptual Counterfactuals using Graphs

5.1 Introduction

Understanding a classifier solely through isolated concepts often proves inadequate, as the relationships or actions linking these concepts are crucial for effective classification. Consider the classifier used in the camera system of an autonomous vehicle, which is designed to detect the presence of pedestrians. This example underscores the difficulty in distinguishing between categories such as “driver”, “pedestrian”, “motorbike”, “bicycle”, and “person” without an understanding of the interactions between the person and the vehicle. The incorporation of edge information is essential; omitting it can lead to an incomplete assessment of the classifier’s biases, particularly if the classifier focuses its attention on the edges, yet this data is not included in our analysis. Therefore, it is imperative to utilize the information from the edges along with the concepts.

In this chapter, we introduce two distinct methodologies that utilize the framework presented in Chapter 4 and the “Explanation Dataset" to compute counterfactuals while preserving the relationships between the concepts present in the instances. These approaches ensure that the interconnectedness of concepts is not overlooked, thereby maintaining the integrity and context of the counterfactual explanations. This is essential for developing a deeper understanding of the underlying models and for enhancing the robustness of decision-making processes influenced by AI systems.

5.2 Transforming Graph into a Set of Concepts

As previously noted, computing the graph edit distance is an NP-hard problem [401]. Although there are optimized algorithms designed to compute it [3], they are not feasible for explanation datasets containing a large number of exemplars.

One way to overcome the complexity is to simplify the problem and work again with *sets* instead of *graphs*, which will allow us to use an algorithm similar to the one presented in Section 4.3 for the computation of explanations [84]. Of course, converting a graph into a set without losing information is not generally possible. In this work, we convert the connected components of exemplars on the ABox graph into **sets of sets** of concepts by rolling up the roles into concepts. Specifically, we add information about *outgoing edges* to the label of each node in the ABox graph by defining new concepts $\exists r.C$ for each pair of role name r and concept name C and then adding $\exists r.C$ to the label of a node a if $r(a,b), C(b) \in \mathcal{A}$ for any $b \in \mathsf{IN}$. Then every exemplar of the explanation dataset is represented as the set of labels of nodes that are part of the connected component of the exemplar on the ABox. For instance, an exemplar e with a connected component:

$$\begin{aligned}\mathcal{A}_e = \{&\mathsf{Exemplar}(e), \mathsf{depicts}(e,a), \mathsf{depicts}(e,b), \mathsf{depicts}(e,c), \mathsf{Cat}(a),\\ &\mathsf{eating}(a,b), \mathsf{Fish}(b), \mathsf{in}(b,c), \mathsf{Water}(c)\}\end{aligned}$$

would be represented as the set of labels (ignoring the Exemplar node):

$$\{\{\mathsf{Cat}, \exists \mathsf{eating.Fish}\}, \{\mathsf{Fish}, \exists \mathsf{in.Water}\}, \{\mathsf{Water}\}\}.$$

Now, to compute counterfactual explanations, we have to solve a *set edit distance* problem between *concept set descriptions* of exemplars.

More specifically, the process of generating counterfactual explanations typically unfolds in two distinct phases. Initially, the preprocessing phase involves calculating the edit paths between each pair of examples within an explanation dataset. This step also includes gathering the classifier's predictions for all examples, incorporating prediction probabilities when they are available. Following this, the second phase aims to identify, from the set of examples, the one that not only shares the minimal edit distance but also belongs to the designated target class. This selection process additionally focuses on either maximizing or minimizing a specific criterion, which is detailed in Section 25.

As for the computational complexity involved, the method relies on a graph-based framework, which inherently presents challenges due to the computational intensity of calculating graph edit distances—a problem classified as NP-Hard. Consequently, this computation must be executed $|\mathsf{EN}|^2$ times. In our experimental setup, we employ a depth-first graph edit distance algorithm as outlined in [3]. This particular algorithm is facilitated by the NetworkX [110] Python package, a choice that underscores its computational efficacy and suitability for handling complex graph-related tasks.

In the process of describing concept sets, we begin by identifying the connected components among the exemplars in the ABox graph. This involves analyzing the relationships and connections between various nodes, which represent different concepts. Once these connections are established, we enhance the nodes' labels with concepts of the form $\exists r.C$. This is done for any node a where there is a relationship $r(a, b)$ and a concept $C(b)$ present in the ABox.

To quantify the difference between two sets of node labels, ℓ_a and ℓ_b, where each label consists of a collection of concepts (either basic atomic concepts or more complex forms like $\exists r.C$), we construct a bipartite graph. In this graph, each concept in ℓ_a is linked to every concept in ℓ_b. The connections between these concepts are assigned a cost based on definitions from the TBox $\mathcal{T}$, as detailed in Section 4.3. When focusing on *concept set descriptions* rather than traditional graphs, where roles are integrated into $\exists r.C$ concepts, the costs for adding or removing an $\exists r.C$ concept to or from a set correspond to those of inserting or deleting both a role assertion $r(a, b)$ and a concept assertion $C(b)$. Therefore, the overall cost is the aggregate of the costs for $e_{\top \to r}$ and $e_{\top \to C}$. In scenarios where a concept C is replaced with $\exists r.D$, a two-step modification is required: the concept C must first be deleted ($e_{C \to \top}$), followed by the insertion of $\exists r.D$ ($e_{\top \to \exists r.D}$). Conversely, when replacing $\exists r.C$ with $\exists s.D$ ($e_{\exists r.C \to \exists s.D}$), the process is akin to changing a role assertion from $r(a, b)$ to $s(a, b)$ and switching a concept assertion from $C(b)$ to $D(b)$, leading to a total cost derived from the combined changes in roles and concepts ($e_{r \to s}$ and $e_{C \to D}$).

To find the optimal transformation from one set of labels to another—effectively minimizing the "edit distance" between them—we employ Karp's algorithm, a method outlined in [143]. This algorithm helps us determine the least costly series of edits required to match one set of concepts to another.

Further, to calculate the edit distance between two more complex structures, L_1 and L_2, where each is a set of sets of concepts, we follow a multi-step approach. Initially, the edit distance for each individual label in L_1 is calculated against every label in L_2 using the method described earlier. This involves a detailed computation for each label pair, which we perform $|L_1||L_2|$ times to cover all possible combinations. The overall edit distance between L_1 and L_2 is then determined by applying the same bipartite graph approach, but this time we adjust the edge weights in the graph based on the previously calculated set edit distances.

Lastly, after processing the explanation dataset and recording the necessary edit paths, generating an explanation for these edit distances can be accomplished with a time complexity of $O(|\mathsf{EN}|)$. This optimization ensures that the explanation generation process is both efficient and scalable, allowing for quick and clear understanding of how different sets of concepts differ from one another.

5.2.1 Evaluation

We conducted four experiments to validate our proposed framework, each with a distinct objective. The first was a comparative user study that evaluated our framework against a state-of-the-art image counterfactual method [348], utilizing the CUB dataset [355].

The second demonstrated a practical scenario where our framework clarified the decision-making process of a black-box classifier trained on the Places dataset [419]. For this, we incorporated semantic data from COCO [198], the Visual Genome [161], and WordNet, assessing how the choice of dataset influenced the clarity of the explanations. Additionally, another part of our evaluation addressed a critical aspect of classification tasks that emphasized the importance of edge detection and involved the use of a scene graph generator to provide semantic insights when no relevant semantic information was previously available.

Human Evaluation on the CUB Dataset

Setting To evaluate the effectiveness of the proposed methodology compared to state-of-the-art results [348], we conducted a human study, utilizing the same source images and tasks as described in prior research.

The CUB dataset [355] initially comes without ground truth scene graphs, presenting a challenge for detailed graphical representations. To address this, we devised a method to construct a graph representation by capitalizing on the available structured annotations. We initiated this process by creating a central node to symbolize the bird, which serves as the focal point of our graph. From this central node, we established "has" edges that connect to various parts of the bird, such as wings, beak, and feathers. Each of these parts is further connected to its specific attributes through edges that are labeled according to the type of feature they represent, such as color, shape, and size. This method of linking not only allows for a clear depiction of the bird's characteristics in a structured graph form but also enhances the dataset's utility for more complex analytical tasks that require detailed and organized visual information.

Due to the lack of a universally accepted metric for assessing the semantic consistency of visual counterfactuals, this approach was necessary. We employed the Label Studio platform for the human survey [1], which offers considerable flexibility and functionality for setting up studies. A screenshot of the annotation interface is shown in the accompanying Figure 5.2.2. Thirty-three participants, primarily graduate students and PhD candidates in computer science, volunteered for this study without compensation. They received only the call for participation and instructions for the labeling process, and the study was conducted online.

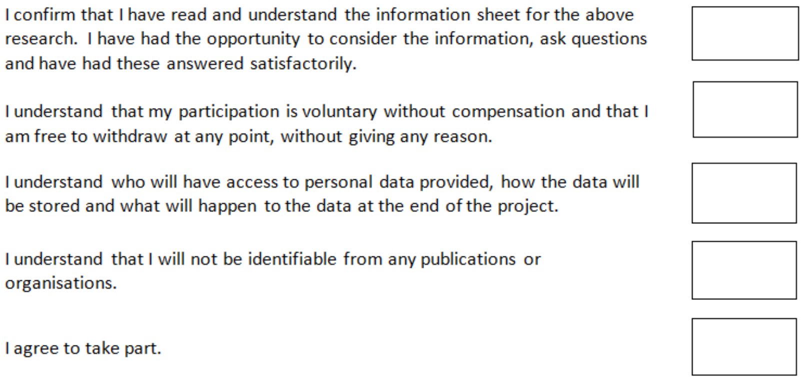
Firstly, an information sheet detailing the objectives and phases of the human surveys was initially distributed to the participants. It was made clear that their participation would be voluntary and uncompensated. Additionally, a consent form in the form of a checklist was distributed to obtain the annotators' consent (see Figure 5.2.1). This form was employed in all human surveys conducted throughout this thesis. Ultimately, the thirty-three individuals who participated were identified as young adults, aged between 19 and 25, encompassing both male and female participants, with no prior knowledge of bird species. The human survey conducted was entirely anonymous, with no personal data being collected from the annotators.

For the technical setup, we acquired two pre-trained classifiers, a VGG-16 [324] and a ResNet-50 [114], to make predictions on the CUB test set. These classifiers were selected because they utilize the same pretrained weights as those used in the research by [348]. This dataset served as our "explanation dataset', with the annotations of the images encoded in a deep learning knowledge base.

Following the methodology outlined in [348], we selected several bird images from the CUB dataset and retrieved the closest counterfactual image for each, ensuring that the counterfactual did not belong to the same bird species as the source. Our algorithm replicated this task with the original source images.

In Figure 5.2.2, a screenshot of the platform provided to our evaluators for the comparative user survey is displayed. To enhance the visibility of the images and their intricate details, we have equipped the platform with user-friendly tools such as "zoom-in"/"zoom-out" capabilities, alongside options to "pan" and "move" within the image. These features ensure that evaluators can examine each image thoroughly before making their selections.

[1] https://labelstud.io/



I confirm that I have read and understand the information sheet for the above research. I have had the opportunity to consider the information, ask questions and have had these answered satisfactorily.	□
I understand that my participation is voluntary without compensation and that I am free to withdraw at any point, without giving any reason.	□
I understand who will have access to personal data provided, how the data will be stored and what will happen to the data at the end of the project.	□
I understand that I will not be identifiable from any publications or organisations.	□
I agree to take part.	□

Figure 5.2.1: This image shows the consent form used for human evaluation. Annotators are required to complete this form prior to beginning their annotation tasks.

	ResNet-50	VGG-16
S.O.T.A. [348]	14.65%	13.68%
Ours	34.93%	23.65%
Can't Tell	**50.42%**	**62.67%**

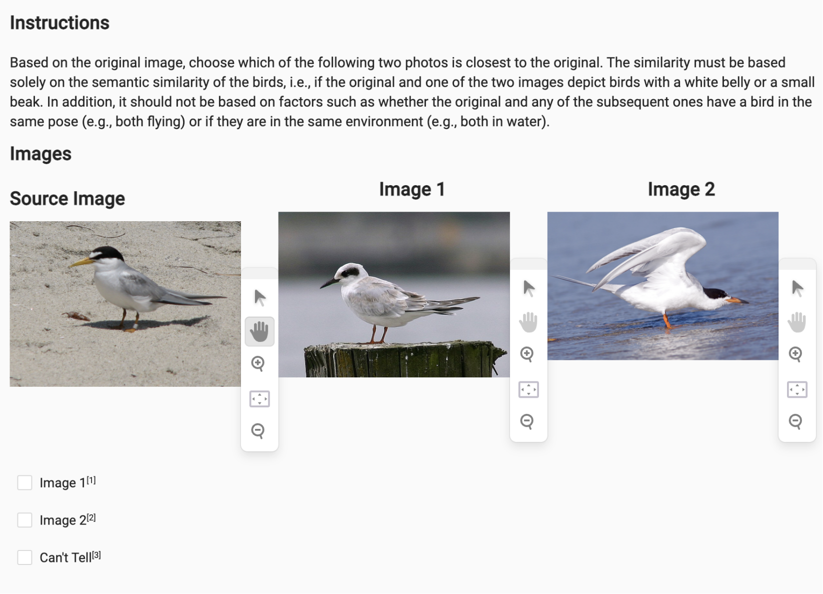
Table 5.1: Human evaluation results on which of the two counterfactual bird images is semantically closer to the source image.

On this screen, the source image is positioned on the left, while two comparative options are displayed in the center and rightmost columns. These options consist of an image generated by our method versus a counterfactual image produced using the [348] method. To maintain impartiality and prevent any bias, the placement of these images is randomized for each sample.

For each annotation task, the annotator is presented with three choices: "Image 1", "Image 2", or "Can't tell". They are required to select one, indicating their judgment on which image most closely matches or represents the source image's context, based on the visual and contextual cues provided. This structured approach allows for a systematic assessment of the effectiveness and relevance of the counterfactual images in comparison to the original, ensuring a fair and unbiased evaluation process.

Results The comparison between the two image retrieval methods (see Figure 5.2.3) showed that they produced highly similar, and at times identical, results. This similarity led to challenges for the evaluators in distinguishing between the counterfactual images generated by each method, as reflected in their similar performance documented in Table 5.1. However the results of our method are improved showed both qualitative and quantitative. By carefully investigating the results in Figure 5.2.3 we can see that in the most cases the image returned from our method is semantically closer to the one returned by [348]. For instance, in the final entry of the left column, it is evident that the original bird and the one identified by our algorithm belong to the same class, suggesting that our result may represent a misclassification by the algorithm. Conversely, the result provided by [348], while closely resembling the species of the original image, actually pertains to a different class, distinguished by noticeable variations such as the coloration on the head. Nevertheless, the images generated by both algorithms bear a striking resemblance to the original, and at first glance, they appear nearly identical. This similarity is even more pronounced in the first entry of the right column, where there are no noticeable semantic differences between the birds in the original image and those in both counterfactual instances.

Notably, our algorithm operates without internal access to the model, unlike the state-of-the-art (SOTA) algorithm, which does have such access. Our method successfully matched the SOTA's results by leveraging only the semantic information associated with the CUB images, without requiring direct access to the underlying classifiers.



Instructions

Based on the original image, choose which of the following two photos is closest to the original. The similarity must be based solely on the semantic similarity of the birds, i.e., if the original and one of the two images depict birds with a white belly or a small beak. In addition, it should not be based on factors such as whether the original and any of the subsequent ones have a bird in the same pose (e.g., both flying) or if they are in the same environment (e.g., both in water).

Images

Source Image

Image 1

Image 2

□ Image 1[1]

□ Image 2[2]

□ Can't Tell[3]

Figure 5.2.2: A screenshot from the annotating platform. The first image always depicts a source image, whereas the second and the third are randomly the counterexample produced by [348] method and the proposed one.

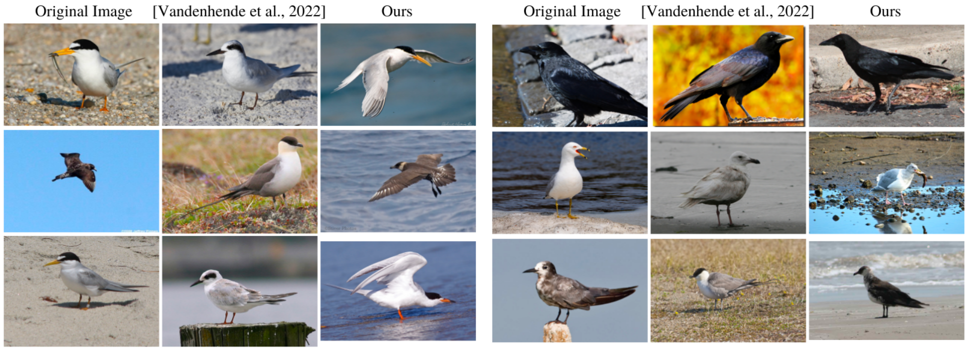


Figure 5.2.3: The first column shows the original image, the second one [348]'s retrieved image and the third one the image retrieved by our algorithm.

Explaining a Places Classifier

At this point, a pivotal query arises: How can we determine the origin of the biases revealed by our system? While we hypothesize that these biases stem from the classifier, it is conceivable that a biased explanation dataset might also produce similar outcomes. To investigate this, we can implement the same analytical task using a different dataset, which would allow us to compare and contrast the findings with those obtained previously.

More specifically, in the previous section (Section 4.4.1), we explored the use of the COCO dataset, which exclusively contains annotated objects without addressing the relationships between these objects in our classifier explanations. This section aims to extend this approach by incorporating the relational information between the objects and investigating the importance of the explanation dataset by comparing the results on the same classifier using a different explanation dataset. More specifically in this experiment, the Visual Genome dataset [161] will be utilized as the "explanation dataset." For this purpose, we have selected the Visual Genome dataset as our cross-checking tool. Visual Genome, like COCO, is among the select few datasets that include annotated images, making it ideal for our comparative study. The Visual Genome dataset is a rich visual resource that offers detailed annotations of images. Unlike the COCO dataset, which primarily focuses on object annotations, Visual Genome includes annotations for both objects and their interrelationships.

Setting For this experiment, we employed the same image classifier trained specifically for scene classification on the PLACES dataset [2], that was used in Chapter 4.4.1. This classifier was then utilized to perform classifications on a selected subset of images from the COCO dataset, which also appear in the Visual Genome dataset. This approach allowed us to examine the variations in explanatory outputs when different algorithms and datasets are used in conjunction.

Each image's object descriptions were linked to WordNet synsets through the NLTK Python library [3]. These synsets served as concept names (CN) and were used in conjunction with the hyponym-hypernym structure of WordNet as a TBox. This structured approach aids in understanding how various contexts and dataset structures affect the explanatory capabilities of our classifier.

Results

The findings obtained from the Visual Genome, as shown in Figure 5.2.5, are set side by side with those from the COCO dataset, depicted in Figure 5.2.4. These results highlight the classifier's remarkably consistent performance across both datasets. Such consistency reinforces the argument that the biases observed are more likely attributes of the classifier's architecture rather than a result of any biased distribution in the datasets employed for these analyses. Nonetheless, significant variations in dataset distribution could lead to differing interpretations, underlining the critical need for meticulous selection of the dataset used for explanations. This point is extensively discussed and analyzed in several works, including [57, 58, 59].

Evaluating the Significance of Roles

In the previous experiment, it was observed that the distinguishing features between classes could often be attributed solely to individual concepts, such as the presence of a bed or a dog. However, there are numerous cases where this approach proves insufficient, and the roles and interactions between elements must be considered. For example, distinguishing between the "driver" and "pedestrian" categories in images containing "motorbike", "bicycle", and "person" cannot be accurately achieved without recognizing the relationships between the person and the vehicle. The roles "rides" or "on" are indicative of the former category, whereas the absence of these roles or the presence of the role "next to" suggests the latter. In this experiment, the effectiveness and relevance of roles were tested, with the classification of driver versus pedestrian serving as the selected task.

Setting The main challenge faced in this experiment was the scarcity of datasets that pair images with their semantic descriptions, a vital component for the functionality of our system. While Visual Genome does

[2] http://places2.csail.mit.edu/index.html
[3] https://www.nltk.org/howto/wordnet.html

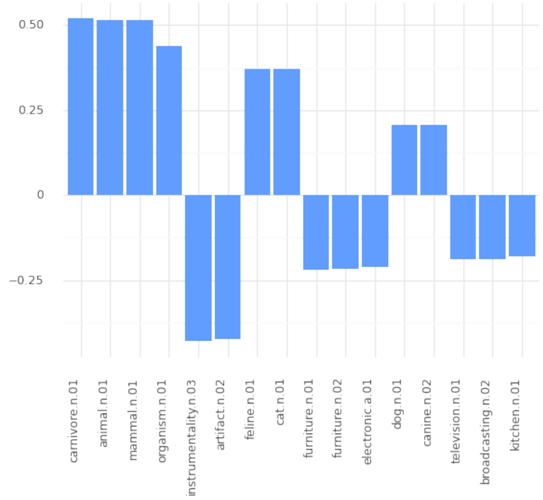


Figure 5.2.4: Global explanation for the subset of COCO which is classified as “bedroom”, with target class “veterinarian”

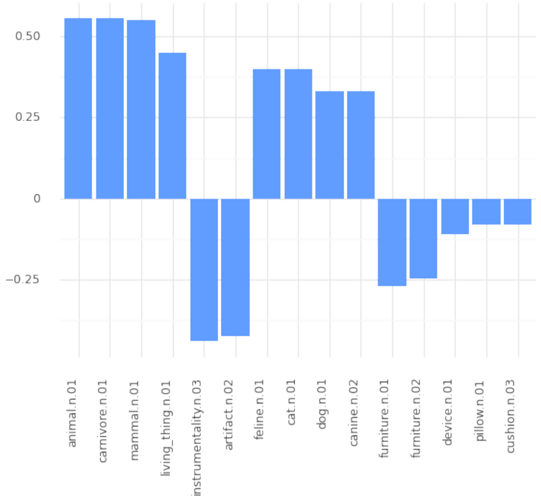


Figure 5.2.5: Global explanation for the subset of Visual Genome which is classified as “bedroom”, with target class “vet”

include roles, their availability is sporadic and inconsistent, appearing in some images but absent in others that are visually similar. Additionally, most real-world scenarios lack accompanying semantic information with their image datasets. To address this issue, we opted to employ a Scene Graph Generator [46] capable of extracting both concepts and roles from images. This integration into our pipeline enables experimentation with any image dataset or the creation of custom datasets using images sourced from the internet, thereby greatly expanding our research capabilities. Following dataset assembly, we extract semantic descriptions using the scene graph generator. The resulting knowledge graphs are generally accurate, although there are occasional discrepancies, such as a person walking a bicycle being mistakenly classified as “riding” it. As our Scene Graph Generator (SGG) we employed “RelTR: Relation Transformer for Scene Graph Generation” [47] and executed it on Google Colab, utilizing the default settings for the model parameters. The model predicted among the 150 entity classes and 50 relationship classes from the Visual Genome dataset. Additionally, a prediction was deemed acceptable if its confidence level exceeded 0.3. An example of the generation of the semantic description is shown in Figure 5.2.6.

The initial phase involves scouring the internet for images that meet our specific criteria, categorizing them into two groups: “**driver**” and “**pedestrian**”. This categorization is specifically applied to individuals on motorbikes and bicycles to prevent the role descriptor from coinciding with the class label, such as in “person driving car”. We utilize search engines like Google, Bing, and Yahoo to compile images based on keywords like “**people**”, “**motorbikes**”, and “**bicycles**”, securing creative commons images which are then manually sorted into two classes:

1. driver class: comprising 63 images of people on bicycles and 127 images of people on motorbikes
2. pedestrian class: including 31 images of people next to parked motorbikes and 38 images of people beside parked bicycles.

With a comprehensive explanation dataset now in hand, containing images and their corresponding knowledge, we proceed to evaluate the dataset through our counterfactual system and analyze the explanations generated for the two defined classes.

Results

Local Explanations In Figure 5.2.7, we observe three instances of local counterfactual explanations applied to a dataset analyzed with the Scene Graph Generator. The images on the left represent the “Pedestrian” class, while those on the right correspond to the “Driver” class. The adjustments suggested for transforming the first image from the “Pedestrian” to the “Driver” class involve adding the concept “ride^bicycle”

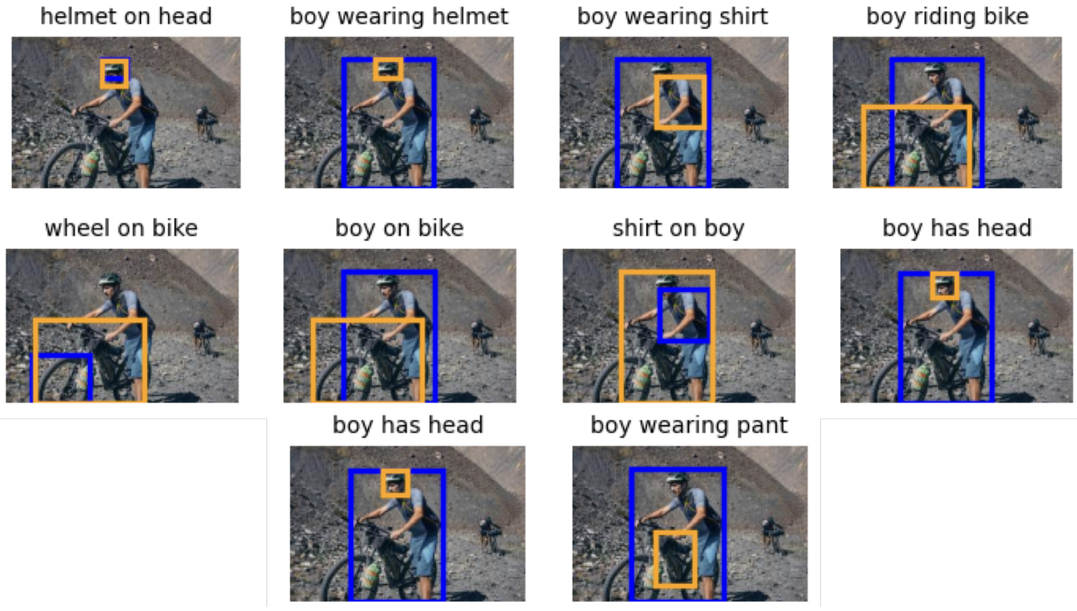


Figure 5.2.6: Example of the Scene graph Generation process for an image from the dataset.

(∃ride.bicycle) to two men in the background, and altering the gender of the individual carrying the bag from female to male. It's important to note that the “ride^bicycle” notation signifies the insertion of this specific role to enhance the classification. The same applies to the transitions illustrated in the second row, where the primary modifications involved altering the gender and incorporating the role “ride^bicycle”. In the transition depicted in the last row, the modification involves changing the entity label from dog to man ($e_{\text{Dog}}\rightarrow$ Man), and including the concept “ride^bicycle” with the entities labeled as man and woman. This change is not only necessary for aligning the image more closely with the “Driver” class but also adds depth to the scene's contextual understanding.

Furthermore, additional adjustments for another “Pedestrian” image include inserting the “riding” role between the person and the bicycle and similarly altering the cyclist's gender. These modifications are strategically chosen to provide minimal yet effective shifts towards the intended classification. The interventions across each pair of images are designed to be both sensible and minimal, maintaining consistency with modifications observed in other datasets. This approach ensures that the counterfactual explanations are not only practical but also maintain a logical connection to the underlying visual elements and their semantic interpretations.

Global Explanations The global counterfactual explanations that facilitate the transition from the “pedestrian” to “driver” classification are illustrated in Figure 5.2.8 through descriptions of concept sets, which include both concepts and their associated roles. The most significant addition, overwhelmingly, is “*ride^wheeled_vehicle*”. This concept serves as an umbrella term, encapsulating both “*ride^bicycle*” and “*ride^motorbike*”, indicating its broader, parent role in the hierarchy of concepts. Subsequent additions include “*wearing^helmet*”, highlighting a critical safety element in driving scenarios. Interestingly, the concept of “helmet”’alone also appears, but less frequently. This may be because, in some images within the driver category, helmets are depicted on the handlebars rather than being worn, indicating a nuanced interpretation of the rider's immediate context and readiness.

Furthermore, the removal of “*wear^hat*” (a subset of “*wear^clothing*”) complements the introduction of “*wear^helmet*”, suggesting a shift from less protective headwear to more safety-focused attire in the driving context. Additionally, “*have^seat*” is eliminated from the descriptions, reflecting the fact that bicycle seats are often not visible when the bikes are in use, thus aligning with the dynamic nature of the “driver” classification.

While other edits are present, they are minimal and their contributions might be less significant, potentially representing noise within the data. Although these could be rationalized, their sparse occurrence suggests they

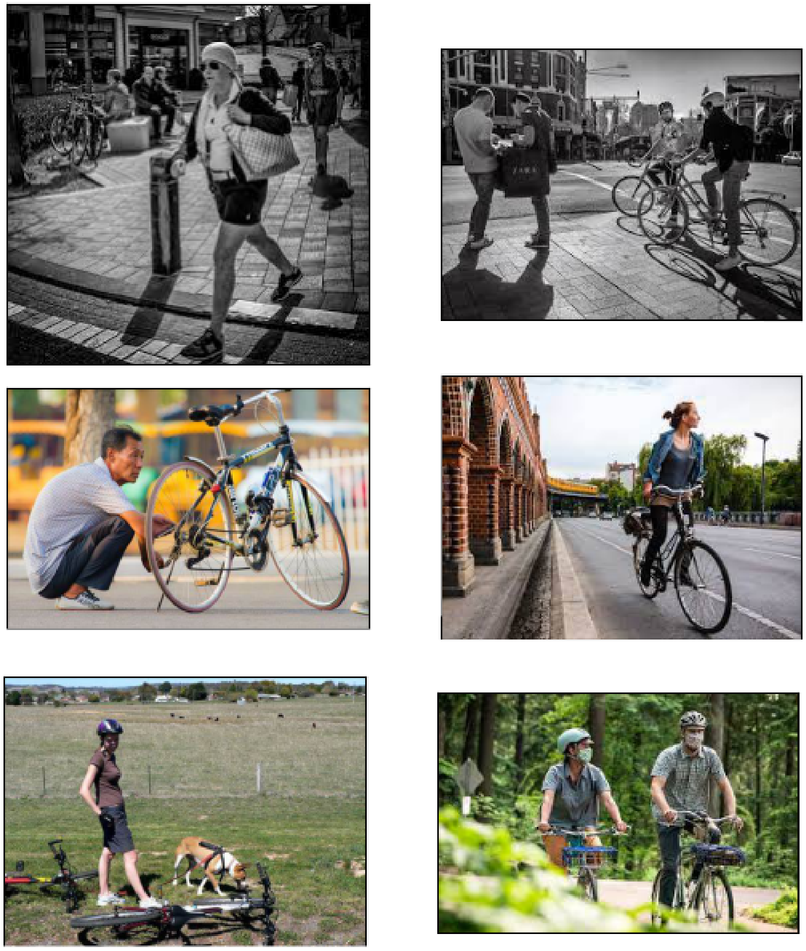


Figure 5.2.7: Three examples, shifting from “pedestrians”’(left) to “drivers”’(right). The main edits are additions of “ride^bicycle”, along with some gender changes and an edit of a dog to a man ($e_{\textsf{Dog}} \rightarrow \textsf{Man}$) in the last row.

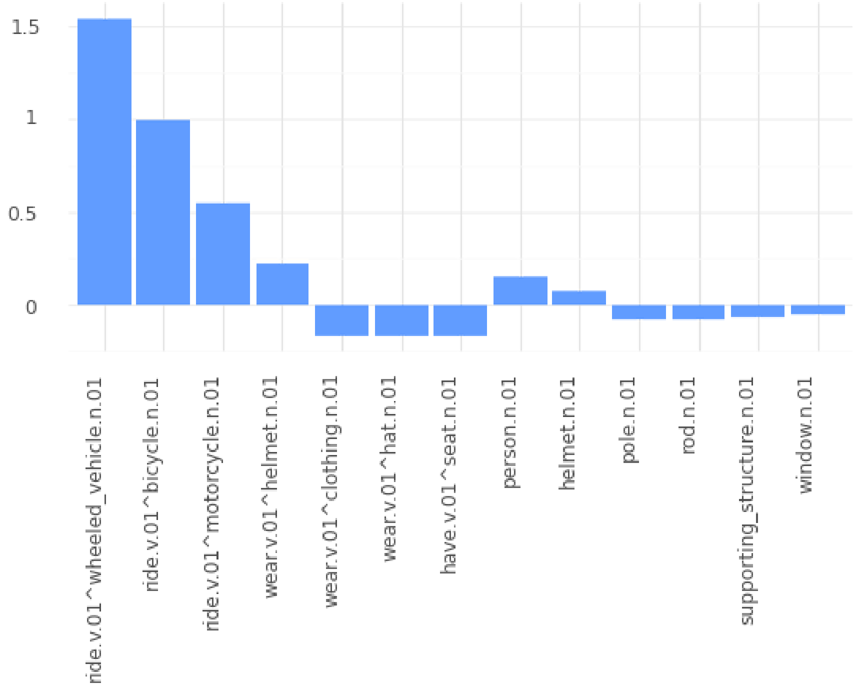


Figure 5.2.8: Flipping class form “pedestrian”’to “driver”, the most important changes are: the addition of “ride^wheeled_vehicle”, “wear^helmet”’and the removal of “wear^hat”.

may not consistently impact the overall classification transition. This analysis underscores the complexity and depth of understanding required to interpret and utilize counterfactual explanations effectively, demonstrating how subtle changes in concept and role descriptions can significantly alter the perceived context of an image.

5.3 Conceptual Counterfactuals using GNNs

While we simplify the problem by aggregating the edges into concepts and tackling a more sophisticated set edit problem to identify the nearest counterfactual instance, some information embedded in the edges, particularly relationships between objects beyond a single hop, continues to be overlooked. Take, for instance, an image that shows a person on a motorbike in a store and another motorbike on the street. A scene graph might easily suggest that the setting is a dealership, with the person testing the motorbike without actually riding it. However, all the previous methods encode this information using labels such as $person, riding\hat{\ }motorbike$, $motorbike, in\hat{\ }store$, and $motorbike, on\hat{\ }road$ might not clearly differentiate which motorbike the person is actually using, potentially leading to inaccurate explanations. Despite these challenges, using graph-based information can significantly enhance our ability to draw precise conclusions, which is particularly vital in areas like Explainable Artificial Intelligence (XAI).

This emphasis on utilizing graph-based methodologies to enhance the precision of interpretations in XAI demonstrates the critical need to advance these technologies. As we transition from theoretical frameworks to practical applications, the connection between conceptual models and graph theory becomes increasingly vital. The process of identifying and comparing instances from different classes using graph methodologies not only illuminates the complexity involved but also underlines the importance of sophisticated tools for managing and interpreting this complexity. As we delve deeper, the integration of Graph Neural Networks (GNNs) offers a promising solution to navigate these challenges efficiently, proving essential in applying these theories to real-world tasks effectively.

Methodology

Given a query instance $I_{(A)}$ from class A, the task involves identifying a different image $I'_{(B)}$ from class B, which is not class A, aiming to minimize the shortest edit path between $I_{(A)}$ and $I'_{(B)}$. Although various metrics exist for measuring distances between images, we adopt a conceptual approach using scene graphs to depict objects and their interactions within images. Consequently, the challenge of image similarity boils down to one of graph similarity.

Graph modifications (such as insertions, deletions, and substitutions) are used to measure deterministically the similarity between two graphs $G_{(A)}$ and $G'_{(B)}$. However, determining these edits is an NP-hard problem. Optimal edit paths can be determined using tree search algorithms, though this approach requires exponential time. In situations where a counterfactual graph to $G_{(A)}$ needs to be identified from a set of N graphs, the graph edit distance (GED) must be computed $N-1$ times.

To reduce computational demands, we employ lightweight Graph Neural Networks (GNNs) that enhance the graph proximity evaluation by mapping all N graphs into a consistent embedding space [65]. Here, the closest instance is determined without losing any information, unlike previous methods. By locating the nearest embedding to $G_{(A)}$ from a different class B, GED calculations are required only once per query during retrieval. Specifically, we address the following optimization problem for semantic graphs derived from any input modality:

$$GED(min|G_{(A)}, G'_{(B)}|),\ \ such\ that\ A \neq B \tag{5.3.1}$$

More specifically, we construct a definitive ground truth by establishing an absolute similarity metric between graph pairs using GED, despite its computational intensity. To enhance the efficiency of GED calculations, we employ a suboptimal algorithm that utilizes a bipartite heuristic to speed up an existing LSAP-based algorithm [140, 74]. Additionally, we refine the graph edits by assigning operation costs based on conceptual distances within the WordNet hierarchy. It is important to note that our methodology is not tied exclusively to the ground truth, and other metrics can be utilized within the current framework. However, we have chosen GED as the primary ground truth metric mainly because previous methods [84, 58] also attempt to approximate GED.

Training "For efficient graph comparison, we deploy a Siamese Graph Neural Network (GNN) architecture that extracts graph embeddings through a combination of different GNN layers. More specifically, for embedding generation, we use stacked GNN layers, described by either GCN [154], GAT [352], or GIN [384]. These embeddings are pooled to generate global graph embeddings, formalized by the equation:

$$h_G = \frac{1}{n}\sum_{i=1}^{n}(u_i^{K-1} + \sum_{j\in\mathcal{N}(i)} u_j^{K-1}) \tag{5.3.2}$$

where u_i is the representation of node i, $N(i)$ is the neighborhood of i, n is the number of nodes for G and K is the number of GCN layers. The embeddings undergo dimensionality reduction to ensure consistency and the model is trained to minimize the loss function:

$$\mathcal{L} = \mathbb{E}(\left\|(h_{G_{(C_x)}} - h_{G'_{(C_y)}})\right\|_2^2 - GED(G_{(C_x)}, G'_{(C_y)})) \tag{5.3.3}$$

.

Upon generating embeddings, they are compared using cosine similarity to rank and retrieve the most suitable counterfactual graph instance, ensuring the instance selected from class B differs from class A and optimizes the graph edit distance, thereby enhancing the explanatory power of these instances for more precise AI explanations.

Once graph embeddings have been extracted, they are compared using cosine similarity to produce rankings. For each query image $I_{(A)}$ and subsequently its scene graph $G_{(A)}$, we obtain the instance $G'_{(B)}$ with the highest rank given the constraint that $I'_{(B)}$ is classified in $B \neq A$. $I'_{(B)}$ is proposed as a CE of $I_{(A)}$ since it constitutes the instance with the minimum graph edit path from it, classified in a different target category B. Specifically, we retrieve a scene graph $G'_{(B)}$ as:

$$G'_{(B)} = G^i_{(B)}, \ \arg\max_i \left(\frac{h_{G^i_{(B)}} \cdot h_{G_{(A)}}}{\left\| h_{G^i_{(B)}} \right\| \left\| h_{G_{(A)}} \right\|} \right) \ \textit{if } B \neq A \tag{5.3.4}$$

where $i = 1, ..., N$. Selecting target class B is correlated with the characteristics of the dataset in use and the goal of the explanation itself. Precisely, if the data instances have ground truth labels, the target class can be defined as the most commonly confused compared to the source image class [348]. Another valid choice is to arbitrarily pick B to facilitate a particular application, i.e. explanation of classifier mistakes, in which case B is the true class of the query image [2]. We choose the first approach when ground truth class labels are available; otherwise, we define the target class as the one with the most highly ranked instance not classified as A.

5.3.1 Evaluation

Our evaluation approach encompasses both quantitative metrics and human-centered experiments to ensure a comprehensive analysis of our methods. The quantitative aspect of our evaluation employs a detailed comparison between the rankings derived from our graph embeddings and the established ground truth rankings provided by GED. We utilize several metrics to gauge the effectiveness and accuracy of our embeddings:

1. *Average Precision@k (P@k)*: This metric considers all results within the top-k ranks retrieved by GED as relevant, offering a broad measure of precision.

2. *Binary P@k* and *Binary NDCG@k*: These focus on the precision and ranking quality of the topmost GED result, emphasizing its position within the retrieved ranks through Normalized Discounted Cumulative Gain (NDCG).

3. *Average number of edits*: We calculate the average number of node and edge modifications—insertions, deletions, and substitutions with different concepts. These calculations are performed post-hoc using GED to ensure a fair and consistent basis for comparison.

The choice of GED as the benchmark for evaluation draws on prior research and is justified by several of its intrinsic features:

- Its semantic richness, which allows for a deep understanding of the conceptual changes between graphs,

- Its completeness in representing distances owing to its reliance on graphs that accurately encapsulate both objects and their relationships,

- Its deterministic nature, making it a reliable standard across various modalities and levels of granularity within evaluated techniques. This universality is critical, particularly when baseline methods may interpret units of information differently, such as pixel-based techniques that consider significant rectangular areas versus those that focus on abstract concepts.

In addition to quantitative metrics, our evaluation includes human-in-the-loop experiments designed to validate and enrich our understanding of the generated counterfactual explanations.

Human evaluators participate in a test designed to assess the effectiveness of our counterfactual explanations. They are involved in a direct comparison task, similar to the one presented in Section 5.2.1 in which they select between two counterfactual explanations for a specific query image: one created using a Graph Neural Network (GNN) and the other using one of two algorithms. The first algorithm is the set-based counterfactuals (SC) approach for calculating semantic counterfactuals, detailed in Section 5.2 and referred to as SC. The second is those proposed by [348], referred to as SVE (Semantic Visual Counterfactuals). The first comparative analysis aims to explore the impact of information loss on the quality of explanations, particularly focusing on how the removal of relationships between objects that are distanced by two or more edges affects the quality of the generated counterfactuals. The second study aims to investigate human preferences between the results from the GNN and the state-of-the-art visual counterfactual that also necessitates white-box access to the classifier.

The results presented in this section primarily involve the utilization of approximately $p \sim N/2$ training graph pairs. Unless specified otherwise, these results are achieved using the Graph Convolutional Network (GCN) variant [155]. The selection of training pairs and the GCN model choice are pivotal in ensuring that our analysis is both robust and representative of the average performance of our frameworks under typical usage conditions. The use of half of the available N graph pairs provides a substantial but manageable dataset, facilitating detailed statistical analysis and model training while maintaining computational efficiency.

Human Evaluation on the CUB Dataset

Setting The experimental setup described here mirrors the framework initially outlined in Section 5.2.1. More specifically, we utilize the same two pre-trained neural network models: VGG-16 [324] and ResNet-50 [114]. These models are employed to process and generate predictions on the test set of the CUB dataset, which we have designated as our *explanation dataset* after incorporating image annotations into a DL knowledge base. Referencing the approach detailed by [348], a subset of bird images from the CUB dataset was selected for analysis. Each chosen image was paired with a counterfactual counterpart from the same dataset, ensuring that the counterpart did not share the same avian species (or label) as the original image.

Building on this established framework, our research integrates these methodologies into a human evaluation survey. In this survey, following the procedure outlined in Section 5.2.1, evaluators were tasked with identifying which of two counterfactual images bore a closer semantic resemblance to the original bird depicted. This assessment explicitly required evaluators to disregard the bird's posture or background elements, focusing solely on semantic similarities. The procedure followed by the annotators remained consistent, including the consent form and the structured layout of the page. The only difference lay in the methods that were compared. In each instance, one of the counterfactual images was generated using the GNN-based method, while the second counterfactual image was randomly selected between the SC and the CVE method.

Results From the analysis of the data in the comparative human survey (Table 5.2), it becomes evident that counterfactual explanations that effectively utilize the complete informational content of the graph edges, are significantly more preferable to humans compared to the other two methods under review. Our method showed nearly double the preference rate compared to the CVE approach, which operates at the pixel level and requires white-box access to the model. Even against with SE, which aggregates edge data into sets thereby losing detailed connectivity information, the GNN approach was favored 2.6 times more frequently, despite a higher number of undecided responses.

The distinct advantage of the GNN method stems from its comprehensive use of graph data—capturing the full spectrum of node and edge relationships—enabling more nuanced and accessible explanations. This fundamental difference in data utilization makes our approach intrinsically more intuitive to users.

Moreover, a chi-square test was conducted to statistically analyze the differences in user preferences. This test highlighted significant disparities, indicating a robust preference for the GNN method over both the SE (p-value = 0.003), and over the SVE techniques (p-value = 9.21e-08). These results not only illustrate a notable deviation from the expected response distribution but also strongly affirm the superior interpretability of our graph-based counterfactual explanations. This statistical validation confirms that our methodology's emphasis on maintaining the integrity of the entire graph structure—utilizing all available information on the edges—significantly enhances the clarity and effectiveness of the generated explanations.

Ours	Win%	Lose%	Tie%
SC	**48.86**	19.32	31.82
CVE	**48.42**	26.27	25.31

Table 5.2: This table shows the percentages reflecting human preferences: Win% indicates the percentage GNN method was favored, Lose% represents the opposite, and Tie% denotes instances of no preference. **Bold** highlights the method with the highest preference rate.

GED-based Quantitative Analysis The agreement between the counterfactuals $I'_{(B)}$ retrieved by each method (CVE, SC, and their approach) and the ground truth GED is examined. It is observed that their method surpasses CVE across all ranking metrics (referenced in Table 5.3). Regarding SC, metrics are applicable only for $k = 1$, as this method generates a single CE rather than a ranked list. Consequently, the precision at 1 (P@1) for SC is recorded at 0.02, significantly lower than that achieved by their method.

Furthermore, it is noted that their approach results in the fewest overall edits. As detailed in Table 5.4, their method produces approximately 1 and 2 fewer edits on average compared to SC and CVE respectively, reinforcing the assertion that their counterfactual explanations (CEs) involve the minimum number of edits necessary.

Additionally, their CEs are characterized by minimal-cost edits; specifically, the resulting Graph Edit Distance (GED) between the query and the retrieved counterfactual scene graph shows lower GED scores when compared to both CVE and SC.

	P@k↑		P@k (binary)↑		NCDG@k (bin.)↑	
	k=1	k=4	k=1	k=4	k=1	k=4
CVE	0.02	0.10	0.02	0.11	0.11	0.26
Ours	**0.19**	**0.34**	**0.19**	**0.49**	**0.23**	**0.36**

Table 5.3: Comparative analysis of counterfactual retrieval outcomes against the benchmark GED rankings on CUB. **Bold** indicates the highest-performing results.

	Node ↓	Edge ↓	Total ↓
CVE	8.43	4.70	13.13
SC	8.07	**3.66**	11.73
Ours	**6.16**	4.34	**10.5**

Table 5.4: Mean edits for nodes, edges, and overall on CUB. **Bolded** values indicate the best outcomes (minimum edits).

Local Explanations Local explanations for the CUB dataset are showcased in Figure 5.3.1, where three images from the class A (Rusty Blackbird) are analyzed. Each image is explored in terms of the necessary edits and Graph Edit Distance (GED) required to transition them to class B (Brewer Blackbird). Notably, our methodology results in the smallest number of concept edits compared to competing approaches.

The SE method demonstrates some significant drawbacks, as seen in the examples where it either introduces additional birds that do not belong to the initial class (SC, left) or displays only a partial view of the bird (SC, middle). These errors lead to costly and unnecessary deletions and additions of elements, detracting from the efficiency and relevance of the explanations. On the other hand, our approach employs a graph-based framework where each concept instance is distinctly linked to graph nodes. This unique linkage, along with the strong interconnections between nodes, robustly guides the graph similarity assessments conducted through GED. Consequently, our method provides a more precise and expressive measurement of distance, offering clear advantages over approaches that rely on simpler, unstructured data sets.

The CVE technique, while avoiding some of the overt errors seen in SC's results, also struggles to find counterfactuals conceptually similar to the original query $I_{(A)}$. This is evidenced by the higher GED and increased number of edits required. Although CVE considers visual features such as zoom, which helps mitigate some mistakes by focusing on finer image details, it lacks the semantic depth provided by our GED-based approach. This semantic depth is crucial as it ensures that the explanations not only visually resemble the query but are also conceptually coherent, preserving the underlying biological and categorical characteristics that define each bird class.

In essence, our GNN-based approach integrates a comprehensive understanding of both the visual and structural aspects of data. By mapping each bird to a graph where nodes represent significant features and edges define the relationships between these features, our model achieves a balance of visual accuracy and semantic richness. This integration enables more intuitive and contextually appropriate transformations, which are essential for producing practical and informative counterfactual explanations in real-world applications.

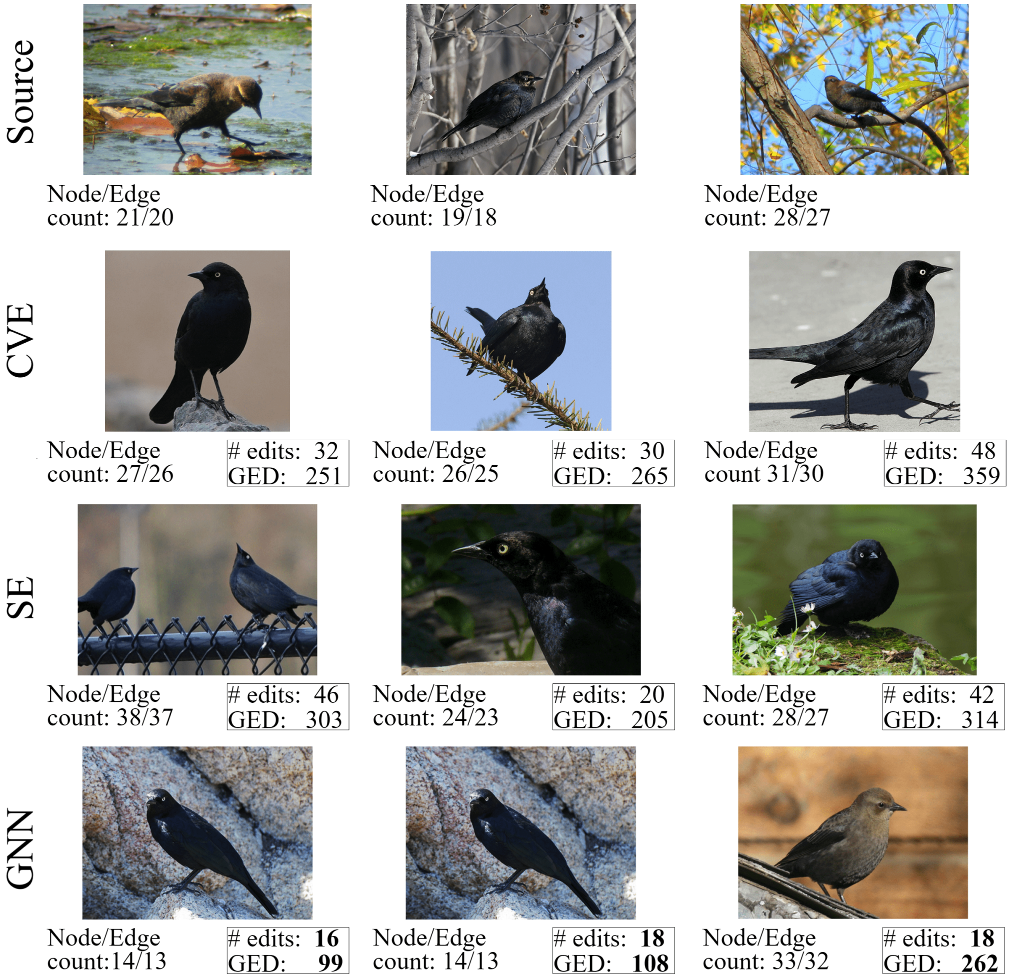
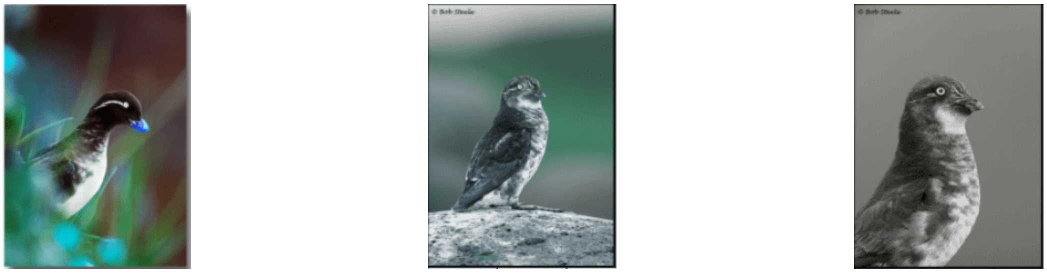


Figure 5.3.1: The results for transitioning from Rusty Blackbird to Brewer Blackbird are presented as follows: The first row displays the original image. The second row showcases the results from CVE method. The third row features the explanations generated by CE. Lastly, the explanations produced by the GNN approach are displayed in the final row. **Bold** denotes best results (lowest number of edits and GED scores).



(a) Source image (b) Top-1 retrieved by CVE [348]. (c) Top-1 retrieved using GNN

Figure 5.3.2: A counterfactual explanation example.

The GNN based algorithm is capable to retrieve counterfactuals that respects not only the semantics of nodes and edges but also the overall geometry of the graph. This capability is manifested in its precision in focusing on semantic details pertinent to bird species, while effectively minimizing distractions caused by irrelevant features such as the background. Such an attribute is a promising aspect of the counterfactuals provided by the framework, contributing towards the development of more robust explanations, despite this particular element not being extensively analyzed within the current study. Initially, a qualitative example is presented to substantiate this claim. In Figure 5.3.2, the most similar image to 5.3.2a is sought using both the CVE method and the framework's own method. It becomes apparent from Figures 5.3.2b and 5.3.2c that both counterfactual images bear a visual resemblance.

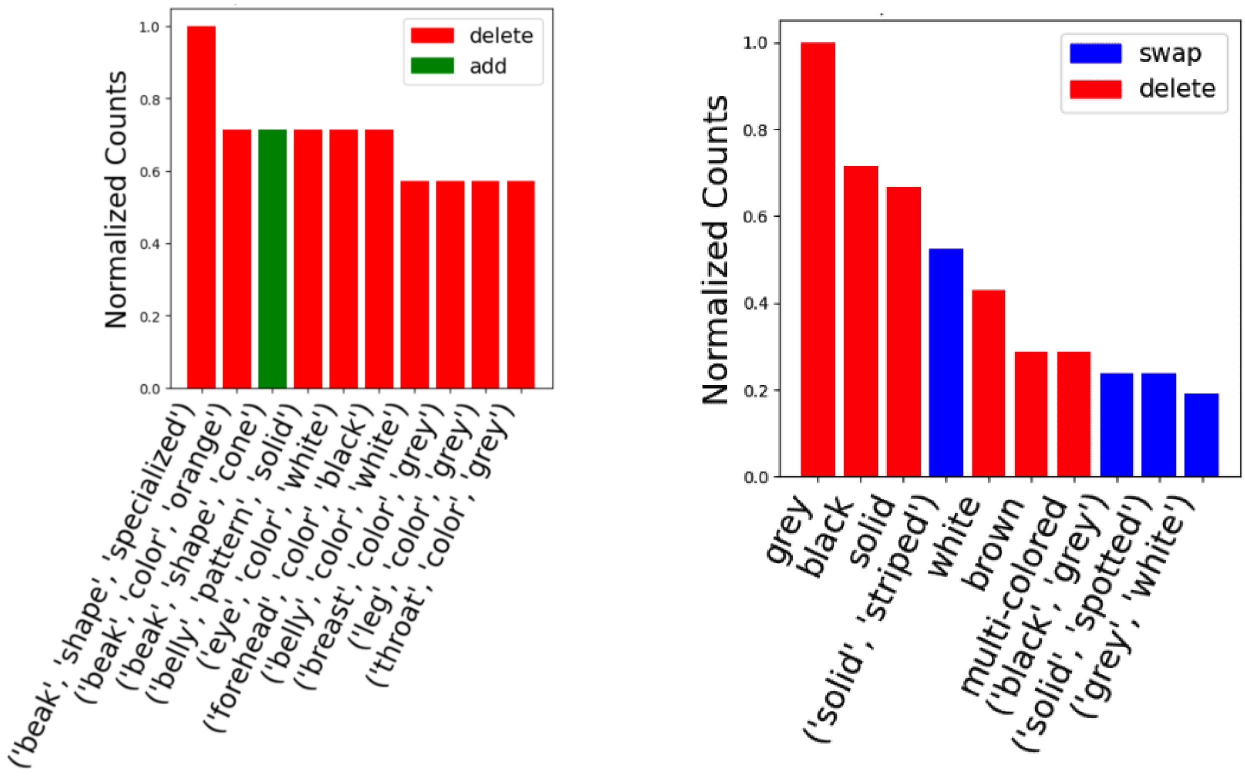
The effectiveness of scene graphs in representing data is highlighted effectively in this context. It is observed that the most similar scene graphs, according to different methods, show distinct characteristics in how they represent the data. Particularly, the method developed by the framework is adept at retrieving graphs that more accurately respect the geometric configuration of the original image's scene graph.

Additionally, it is noted that the framework's approach manages to retrieve an image that excludes certain concepts such as "leg"'or "tail." This exclusion results in a representation that more closely mirrors the original source image. This structural similarity, therefore, leads to better semantic consistency. This aspect emphasizes the framework's ability to deliver precise and meaningful counterfactual explanations, showcasing its potential to provide deeper insights and more reliable interpretations in the analysis of visual data.

Global Counterfactuals Global Counterfactual Explanations entail adjustments within a structured framework using standardized units. In our analysis, these units are primarily graph triples formatted as *(concept-edge-concept)* or simpler *concept edits* within these triples. Both approaches aim to compile local edits to craft a comprehensive explanation of the classifier's behavior from a broader, macro perspective. Utilizing the CUB dataset as a case study, it becomes evident that global CEs align closely with human perceptual understanding.

For example, during the classification shift from Parakeet Auklet to Least Auklet, notable features such as the triplet *("beak", "shape", "specialized")* are removed to de-emphasize characteristics of the original class. Concurrently, features representative of the target class, such as the triplet *("beak", "shape", "cone")*, are introduced to mirror the new class accurately. This method of aggregating edits from multiple images across the dataset allows for the extraction of *global edits.* These edits collectively delineate the necessary modifications across the dataset to elucidate the transition between classes. While these modifications are most effectively represented as graph triples, it is also feasible to detail changes in concepts or relationships.

In support of this, Figure 5.3.3a illustrates the specific triple edits that facilitate the counterfactual transition from Parakeet Auklet to Least Auklet. Additionally, Figure 5.3.3b displays the global edits related to concepts observed in the CUB dataset images. The correspondence of these results with human perception further validates the effectiveness and relevance of global Counterfactual Explanations in understanding and interpreting AI classifier decisions within a complex dataset.



(a)

(b)

Figure 5.3.3: Edits involving triples (a) and concepts (b) (insertions, deletions, substitutions) necessary for transitioning from Parakeet Auklet to Least Auklet.

Explaining a Places Classifier

In this section, we have directed our focus towards conceptual counterfactuals, particularly in light of the previous sections which highlighted their clear advantages over the state-of-the-art (SotA) pixel-level method employed by CVE. Following the experimental of Section 5.2.1, we utilize the Visual Genome (VG) dataset [161], which comprises over 108,000 human-annotated scene graphs. These graphs intricately detail scenes featuring multiple objects and their interactions.

Setting To facilitate manageable experimentation, we have constructed two subsets of 500 scene graphs each, yielding approximately 125,000 potential training graph pairs for our Graph Neural Networks (GNNs). The first subset, referred to as **VG-RANDOM**, consists of randomly selected scene graphs. The second subset, termed **VG-DENSE**, is specifically curated to include graphs with higher densities and fewer isolated nodes, emphasizing the significance of object interconnections.

In Table 5.5, we provide supplementary statistical information about these two datasets, detailing both the maximum and minimum nodes. The datasets VG-DENSE and VG-RANDOM each comprise 500 graphs in total.

When analyzing the results in the experimental section, it is crucial to take into account the size and density of the input data, as these factors can significantly influence the outcomes of the study.

A notable challenge within the VG dataset is the absence of ground truth classification labels, which presents a unique opportunity to evaluate our counterfactual retrieval method in scenarios devoid of predefined target classes. To address this, we employ the same pre-trained Places365 classifier [419], using a ResNet50 architecture. This classifier helps us determine counterfactual classes based on the closest rankings, thus allowing an effective evaluation of our method's capability in identifying relevant counterfactual explanations across

		VG-DENSE	VG-RANDOM
Mean	density	0.20	0.06
	edges	9.04	8.77
	nodes	7.25	14.57
	isolated nodes	0.47	3.37
Max	density	0.47	0.67
	edges	36	27
	nodes	15	20
	isolated nodes	3	12
Min	density	0.14	0.01
	edges	5	5
	nodes	6	4
	isolated nodes	0	0

Table 5.5: Detailed statistical data for the VG-DENSE and VG-RANDOM graph datasets.

diverse and complex visual scenes.

Results Initially, we assessed the average number of edits between our method and the SC approach, as documented in Table 5.6. At first glance, the numerical outcomes from both methods appear comparable. However, a detailed analysis, coupled with the average GED outcomes from Table 5.7, clearly showcases the superiority of the GNN approach. It is important to note that the VG dataset features a much wider diversity of concepts compared to CUB, and despite the stringent knowledge-based constraints applied during GED computation, a higher edit distance between concepts is anticipated. Interestingly, this increase in edit distance does not apply to mean GED, as CUB records a higher number of average edits.

Furthermore, the GNN method consistently achieves lower GED across all scenarios, even in cases where the number of edits is higher, such as with VG-RANDOM.

	VG-DENSE			VG-RANDOM		
	Node↓	Edge↓	Total↓	Node↓	Edge↓	Total↓
SC	**4.91**	7.29	12.2	**12.15**	**7.52**	**19.67**
Ours	4.95	**7.15**	**12.11**	12.18	7.54	19.72

Table 5.6: Average number of node, edge & total edits on VG. **Bold** denotes best results (lowest number of edits).

	VG-DENSE ↓	VG-RANDOM ↓
SC	128.67	186.77
Ours	**122.41**	**180.67**

Table 5.7: Average top-1 GED (VG) for CEs when methods disagree. **Bold** for best (lowest) GED scores for each dataset split.

Local Explanations By analyzing the counterfactual images retrieved for VG-DENSE as shown in Figure 5.3.4 (left), it becomes evident that our method yields results that are significantly more detail-oriented. For instance, in the first column, not only does our approach successfully retrieve an image incorporating the concepts "man', "board', and "water', but it also captures the intricate relation of "man on board'. In contrast, in the third column, while the GNN manages to retrieve a pizza by considering the specific toppings involved, the SC approach merely retrieves an image featuring similar but less connected concepts, such as "bun" and "bread" or "meat" and "sausage'.

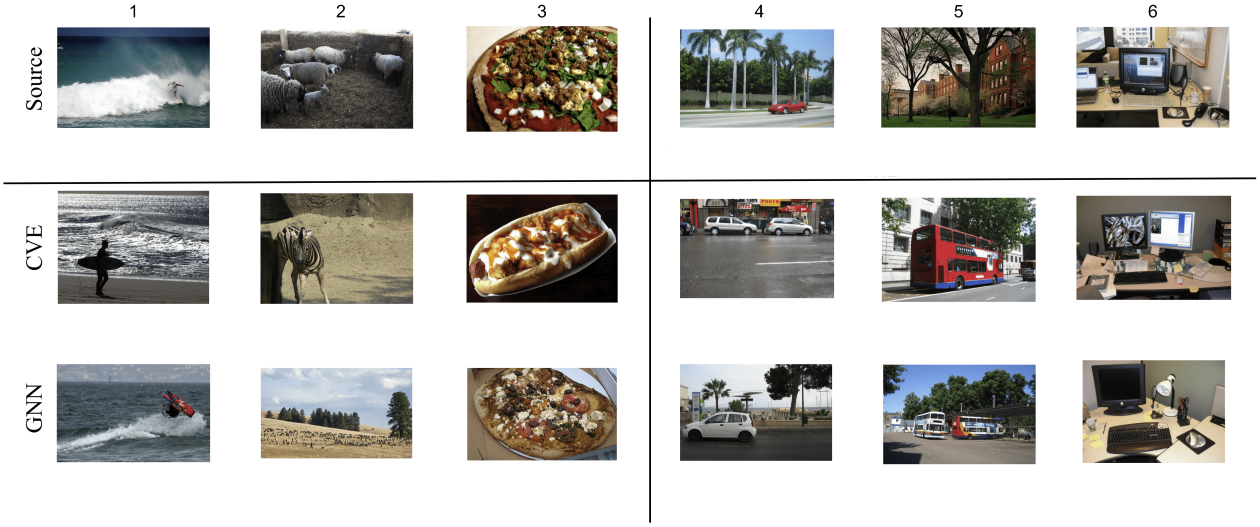


Figure 5.3.4: Qualitative outcomes (optimal metrics highlighted in **bold**): VG-DENSE (first three columns on the left) and VG-RANDOM (last three columns on the right).

Similarly, the results for VG-RANDOM depicted in Figure 5.3.4 (right) adhere to this same nuanced approach. In columns four and five, GNN emphasizes the relational dynamics, retrieving images that focus on the interactions between trees and other objects. However, given the sparsity of the underlying graphs in some instances, as observed in the sixth column, the prominence of certain concepts occasionally overshadows the structural connections. This discrepancy is reflected in the increased number of edits associated with our method for VG-RANDOM. Despite this, the Graph Edit Distance (GED) does not always align with this increase, highlighting yet again the critical role of semantic context in evaluating these images.

Evaluating the Significance of Roles

We have undertaken a replication of the experiment presented in 5.2.1, which involves explaining the classification of web-crawled creative-commons images into "driver" and "pedestrian" categories. In this experiment, the images were manually classified, leading to the use of a non-neural classifier to explain the classifications. Utilizing the state-of-the-art scene graph generator (SGG) by [47], we extracted global edits from the generated graphs to facilitate the transition from "pedestrian" to "driver", as depicted in Figure 5.3.5 (left).

The relevance of these edits is corroborated by intuitive reasoning; for instance, the addition of relationships such as (helmet, on, head) and (man, on, bike) align with the common understanding that people wear helmets when riding bikes. Similarly, the deletion of (seat, on, bike) reflects the observation that the bike seat is obscured when a person is riding. These intuitive edits illustrate the practical application of common sense in refining scene graph outputs.

To assess the robustness of our methodology across different annotation techniques, we substituted the original SGG with a combined pipeline involving image captioning using BLIP [181] followed by graph parsing through Unified VSE [373]. This replacement was tested to see if the semantic integrity of the edits is maintained across different technological approaches. The results, shown in Figure 5.3.5 (right), confirm that the edits generated through this new pipeline closely resemble those produced by the original SGG method.

The comparison reveals that more precise local edits can be achieved by carefully considering the multiplicity of objects and their interrelations. However, it is also evident that generic triple edits can occur due to inaccuracies within the automatic annotation pipeline, underscoring the necessity for diligent curation of explanation datasets. This curation is crucial to minimize errors and enhance the quality of explanations provided by automated systems.

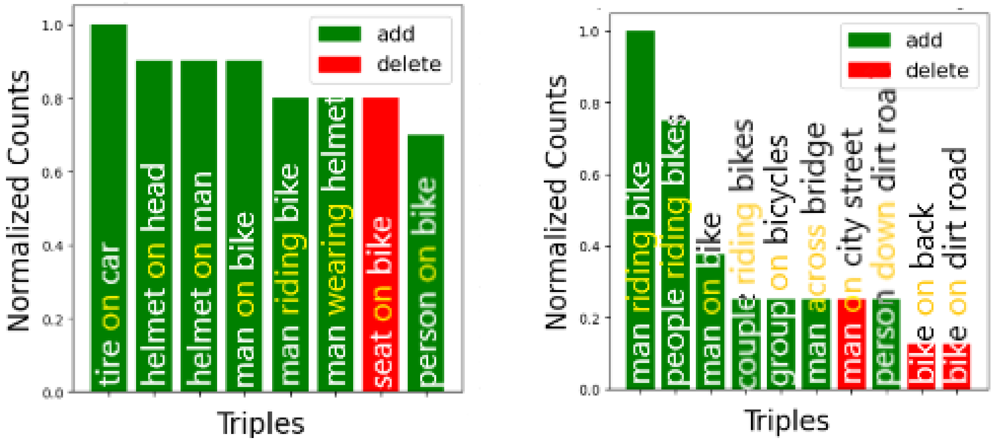


Figure 5.3.5: Modifications of graph triples (insertions/deletions) to transition from "pedestrian" to "driver". Edge and node labels within a triple are highlighted in yellow for clarity.

Evaluation in Audio - COVID-19 Classification

As the algorithms presented in the previous sections, the method described here is also model-agnostic and modality-agnostic. Thus, we can use the same framework to explain a classifier across different modalities. In this instance, we utilized the Smarty4covid dataset [398], notable for its use in the IEEE COVID-19 sensor informatics competition [4], which identifies COVID-19 from cough sounds. This extension into audio classification aligns our findings with those derived from the SC method, particularly in highlighting frequent concept edits related to respiratory symptoms and the exposure of an existing gender bias within the data.

This dataset, characterized primarily by its conceptual nature and minimal interconnections, did not yield new insights beyond those previously established, reaffirming the importance of the nature and density of annotations. Nonetheless, it confirmed that our method performs comparably to SC, even in this less conventional application.

The methodology for generating Smarty4covid graphs for this dataset involved several adaptations from our standard procedures. Each user or patient was directly linked to their symptoms and characteristics, which were discernible audibly to some degree. The analysis of symptoms occasionally required categorizing certain symptoms as sub-symptoms based on the hierarchical structure outlined in [398, 58] Smarty4covid hierarchy, deviating from our usual practice of using WordNet [243] to calculate node edit costs. Due to the simplicity of edge types within this dataset, the strategy for modifying edges was streamlined, treating edge swaps and the addition or deletion of edges as significant alterations.

To refine the accuracy of these adaptations, we incorporated custom BioBert embeddings [170] for the GNN similarity component, recognizing the unique linguistic characteristics of the medical field. This choice marked a departure from our previous reliance on simpler Glove embeddings [275], aiming to better capture the specific semantic nuances of medical terminology.

Comprehensive details of global edits are documented in Table 5.8, which includes triple edits encompassing edge edits and adjacent concepts. For clarity, the structure of the triples has been simplified in the table, omitting the head and predicate where all heads are labeled as the"User"concept, and predicates represent symptoms or sub-symptoms. The latter part of Table 5.8 focuses on node edits independently of the edges. This detailed examination not only confirms previous findings but also reveals additional insights, such as the reported gender bias and an emerging correlation between COVID-19 positivity and younger demographics, expanding our understanding of the dataset's complexities.

[4] https://healthcaresummit.ieee.org/data-hackathon/ieee-covid-19-sensor-informatics-challenge/

Concept Edits	Normalized Counts	Triple Edits	Normalized Counts
“Sneezing”	1.0	“Sneezing”	1.0
“RunnyNose”	0.78	“RunnyNose”	0.73
“DryThroat”	0.35	(’Male’,“Female”)	0.68
“Fever”	0.34	“DryThroat”	0.36
“Dizziness”	0.31	“Fever”	0.35
“Fatigue”	0.22	“Dizziness”	0.31
“Respiratory”	0.22	(“Fourties”, “Twenties”)	0.29
“DryCough”	0.21	“DryCough”	0.23
“TasteLoss”	0.21	“Fatigue”	0.23
“Cough”	0.16	“Respiratory”	0.23

Table 5.8: Comprehensive edits for transitioning from COVID-19 Negative to Positive status, displayed through concepts and triples.

5.4 Assessing the Importance of Conceptual Explanations

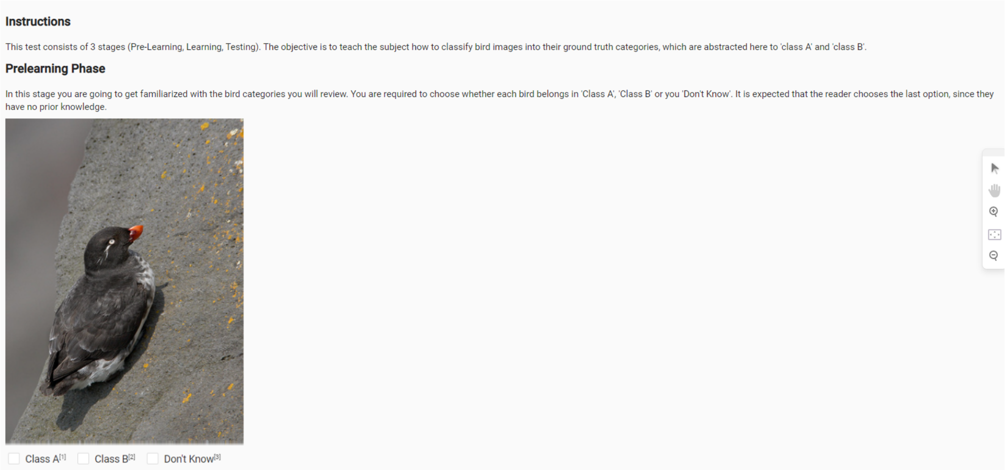
In the previous Sections, we explored various algorithms for calculating Semantic Counterfactual Explanations. Initially, the incorporation of semantics appears to be a logical approach to unravel the decision-making processes of opaque systems, often referred to as “black boxes.” This perspective is supported by recent scholarly works which suggest that the primary distinction between counterfactual explanations and adversarial attacks lies in the presence of semantic coherence [27]. However, the validity of this assumption requires empirical verification. Specifically, our research aims to test the following hypothesis: “Does the use of appropriate semantics actually aid users in comprehending the decision-making process of a black box?” Here, the term “appropriate semantics” implies that the explanations provided are congruent with the semantic decision-making process inherent to the system being analyzed. For instance, in scenarios such as the pedestrian versus driver classification, it would be misguided to analyze the decision-making process without considering the relational data provided by edges within the graph. If the semantic level employed by the black box is known, then the highest level of relevant information should be utilized. In such cases, a Graph Neural Network (GNN) approach may be more appropriate than a set-based counterfactual explanation, particularly if the impact of relationships between edges on the classifier’s decisions is unclear.

To empirically test this hypothesis, we plan to conduct a series of human surveys incorporating elements of machine teaching. Specifically, we will adapt the CVE’s machine-teaching experiment outlined [348], but restructured to incorporate our graph-based explanations. This experiment will be divided into three phases: pre-learning, learning, and testing, with participants split into two distinct groups to experience different learning conditions. The first group will participate in a “visually-informed” session, where they will be presented with both images and their corresponding scene graphs. The second group, referred to as the “blind” group, will receive only the scene graph pairs and edits, without any visual context.

This dual-method approach will enable us to evaluate how effectively humans can grasp and utilize graph-based concepts when visual aids are absent, introducing a new application of this evaluation technique. We anticipate that this study will illuminate the relative influence of visual versus conceptual information on human understanding and decision-making in complex tasks like image classification, thereby providing deeper insights into the efficacy of semantic counterfactual explanations.

5.4.1 Setting of the Experiment

This experiment aimed to delve deeper into the understandability of both the traditional and our novel methodologies by replicating the machine-teaching human experiment described in [348], using the CUB dataset with the same classes and procedure as reported in Section 5.2.1, with the only modifications to incorporate the graph-based explanations. We maintained the structured stages of pre-learning, learning, and testing, and divided our annotators into two separate groups to follow different learning protocols: “visually-informed” and “blind”. Unique to our study, the “blind” variant provided annotators solely with scene graph pairs and graph edits, omitting any visual images. This innovative approach was designed



Instructions

This test consists of 3 stages (Pre-Learning, Learning, Testing). The objective is to teach the subject how to classify bird images into their ground truth categories, which are abstracted here to 'class A' and 'class B'.

Prelearning Phase

In this stage you are going to get familiarized with the bird categories you will review. You are required to choose whether each bird belongs in 'Class A', 'Class B' or you 'Don't Know'. It is expected that the reader chooses the last option, since they have no prior knowledge.

□ Class A[1] □ Class B[2] □ Don't Know[3]

Figure 5.4.1: Initial instructions for the CUB machine teaching experiment during the Pre-Learning phase. Participants can select from “Class A”, “Class B”, or “I don’t know”.

to uniquely assess the extent to which individuals rely on graph-based concepts over visual imagery to comprehend the reasoning behind classifications. This method, offers valuable insights into the cognitive processes involved in understanding complex data representations and enhances our understanding of the effectiveness of explanatory models in AI.

The same platform used for the prior human experiment is employed once again. However, in this iteration, each annotator is restricted to evaluating only **one** single sample. This limitation is imposed to more clearly assess the contribution of the learning phase, avoiding scenarios where an annotator might become more “competent” after multiple exposures to the learning phase. The experimental workflow, as outlined by [348], is adopted, thus incorporating all three stages: pre-learning, learning, and testing.

Pre-learning stage During the pre-learning stage, unlabeled images from the test set are presented to the users so that they can become acquainted with the types of images they will be required to classify later. Figure 5.4.1 serves as an illustration of the pre-learning screen. It is made clear to the annotators that classification into the anonymized classes A and B cannot be performed without progressing through the learning stage, and thus, selecting "I don’t know" is the anticipated response. In Figure 5.4.1, the three available options for image classification are explicitly displayed: “Class A”, “Class B”, or “I don’t know”. It is stipulated that only one option can be selected at any given time, consistent with the procedures outlined in [348].

Learning stage The learning stage constitutes the core of this human experiment. As detailed in the main paper, two variants are conducted to assess the extent to which concepts influence human perception. It is stipulated that a participant engages in either the “visually-informed” or the “blind” experiment, but not both, to preclude the evaluation of the same data sample in both experiments and thereby prevent any potential knowledge transfer between the two variants. Participants are divided into equal subgroups, with seventeen in the “visually-informed” variant and sixteen in the “blind” one.

In the **visually-informed** variant, training images from anonymized classes A and B are presented to annotators, accompanied by their scene graphs, as depicted in Figure 5.4.2. To ensure no overlap between training and test images, various tools such as “zoom-in”/“zoom-out”, “pan”, and “move” are provided to annotators to facilitate navigation within the images and the corresponding scene graphs.

In the described setup, images positioned on the left are invariably assigned to class A, whereas those on the right are categorized under class B. Scene graphs displayed on the right elucidate the edits required for the $A \rightarrow B$ transition; green nodes symbolize concept additions, blue nodes indicate concept substitutions (showing both the source and target concepts), and red nodes mark concept deletions. Nodes of other colors suggest that the associated concepts remain unchanged across the two classes.

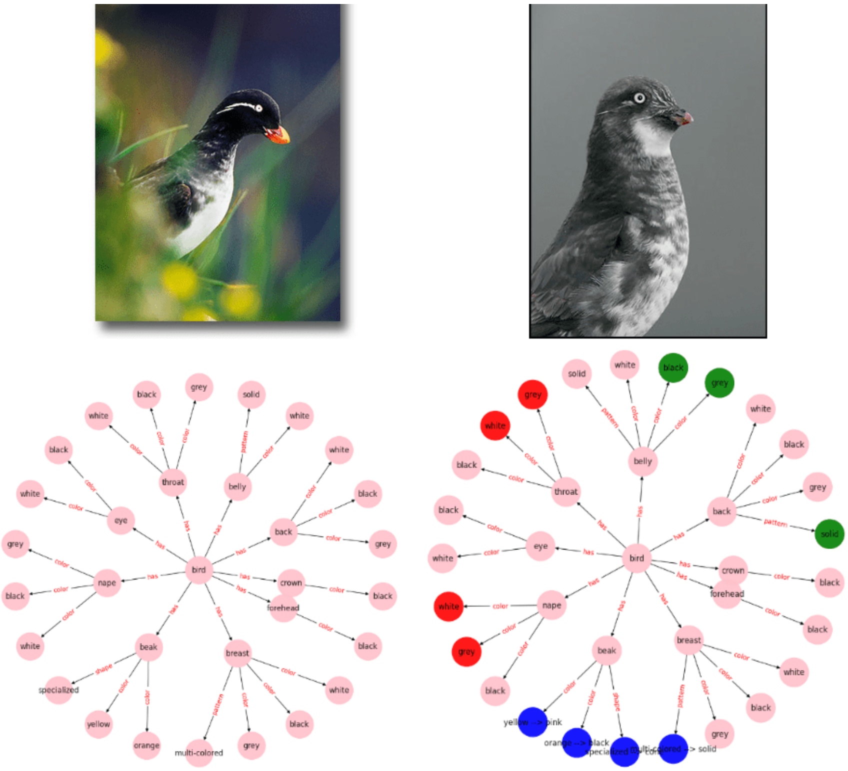


Figure 5.4.2: Example of the visually-informed learning stage.

During the training phase, it is observed that a user's attention is naturally drawn to the most frequent insertions, substitutions, and deletions. This focus aids in identifying the discriminative features between class A and class B. The association of such concepts with corresponding images facilitates the mapping of graph edits to visual differences, enabling users to distinguish between classes both visually and conceptually.

In the "blind" variant of the learning stage, participants are provided only with scene graphs, devoid of any training images, and the graph edits are demonstrated through colored nodes. This approach mirrors the machine-teaching learning stage presented in [348], where only discriminative regions of the images are highlighted, and the rest of the image is obscured. Thus, annotators are required to learn from these explanations alone, mentally associating the explained concepts with visual regions in the test images. In this scenario, the explanations are linked to graph edits, and annotators must recognize the discriminative concepts that are added, substituted, or deleted for the $A \rightarrow B$ transition.

However, given that this learning environment lacks visual cues, it is considered to be more challenging than the learning stage implemented by Vandenhende et al. Here, annotators must bridge concepts with image regions, engaging in cross-modal grounding to identify discriminative features. Throughout this blind learning stage, the extent to which annotators rely on concepts over pixels to classify images from unknown classes is measured. This experiment is crucial in demonstrating how conceptual explanations can significantly aid humans in approximating a zero-shot classification setting, highlighting the importance and informativeness of such explanations.

Testing Stage In the testing stage, the same images as in the pre-learning stage are provided to the users, but no scene graphs are included. Based on the knowledge acquired in the previous stage, annotators are expected to have grasped the visual and conceptual differences between the classes; hence, they are required to assign an appropriate class to each test image by choosing either "class A" or "class B" for each one. In contrast to the pre-learning stage, the option "I don't know" is not available. Following this stage, an accuracy score for each user is calculated based on their correct choices during the testing stage.

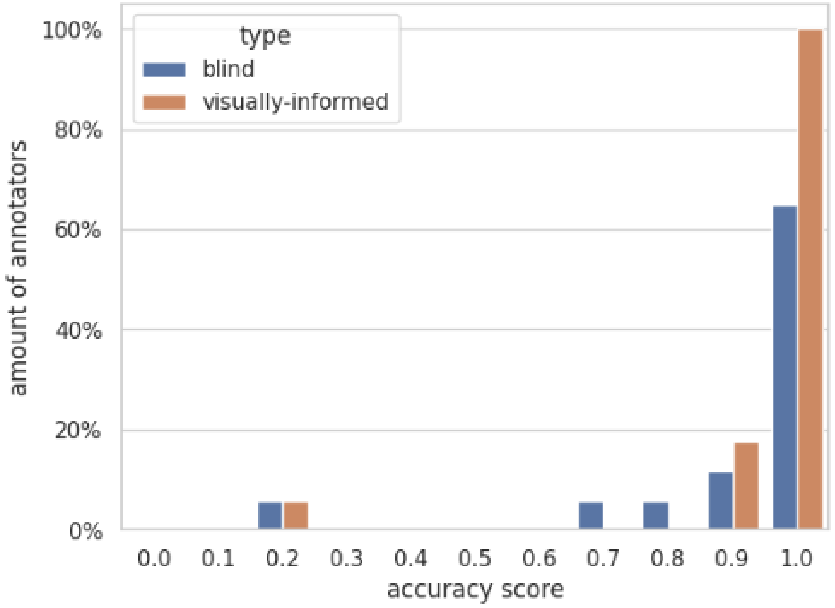


Figure 5.4.3: Variability in test accuracy across human evaluation experiments in machine teaching.

5.4.2 Results

The accuracy of the GNN approach in the visually-informed trials significantly exceeds the scores reported in CVE, underscoring the enhanced effectiveness of semantic counterfactual explanations in leading humans to comprehend the distinguishing concepts between classes, as opposed to the more basic pixel-level CEs that lack conceptual depth. The results from the “blind” experiment reveal a predictable decline in accuracy compared to the visually-informed outcomes, yet they still surpass the performance noted in CVE. This greater accuracy in concept-based explanations as opposed to purely visual ones confirms the importance that humans attribute to higher-level features in classification tasks.

Human experiment	Test accuracy %↑
GNN - visually-informed	**93.88**
GNN - blind	89.28
CVE	82.1

Table 5.9: Accuracy scores of human participants for accurately classifying samples into classes A and B. The highest score is highlighted in **bold**.

The average accuracy for the visually-informed experiment stands at 93.88%, suggesting that users are generally highly adept at identifying the key concepts that distinguish the two bird classes and associating them with visual information. The average accuracy for the blind experiment is noted at 89.28%. This figure, being quite close to that of the visually-informed experiment, allows us to conclude that concepts alone are sufficiently robust for teaching discriminative characteristics to humans, even in the absence of direct visual context. The accuracy scores for both the visually-informed and blind experiments significantly surpass those reported in CVE, indicating that conceptual explanations are more meaningful and informative to humans compared to pixel-level explanations.

Figure 5.4.3 offers a detailed breakdown of the accuracy scores attained by participants in the testing phase of the machine teaching experiment. It is evident that the scores predominantly reach highs of 0.9 and 1.0, indicating that the explanations generated by our method are highly interpretable for humans and advantageous for executing classification tasks. A comparison of the “visually-informed” and “blind” results indicates a gradual reduction in test accuracy when visual aids are absent.

Applicability of Machine-Teaching Experiment

The machine-teaching experiment is deliberately conducted using only the CUB dataset. This decision serves to underscore the advantages of the learning phase: since annotators initially lack knowledge about bird species, they stand to greatly benefit from acquiring distinctive bird characteristics during the learning phase, which they can then utilize in the testing phase. For instance, before the experiment, none of the annotators can distinguish between a Parakeet Auklet and a Least Auklet. However, after participating in the learning stage, they gain the ability to identify key discriminative features, aiding them in accurately classifying birds during the test phase.

Conversely, the Visual Genome dataset, which comprises images of common everyday scenes, does not lend itself well to a similar experiment. For example, most people already understand the fundamental differences between a kitchen and a bedroom, making a learning phase unnecessary in these contexts, even if the scene labels are hidden. This situation can be likened to an instance of data leakage.

Additionally, there is a potential issue with misleading concepts. In some instances, certain concepts might lead visual classifiers to develop biases, a problem typically not encountered by humans. Take, for instance, the presence of a TV, which could be found in both kitchens and bedrooms. If, hypothetically, the selected images of bedrooms all featured TVs while those of kitchens did not, the explanatory graphs might overly emphasize the addition of "add TV" nodes. Consequently, a human might be expected to classify any image with a TV as a bedroom and any without as a kitchen, mirroring the potential bias of a visual classifier trained on such data. Yet, when faced with actual test images, humans are unlikely to be swayed by the presence or absence of TVs, instead relying on common sense for classification. Therefore, not only would the learning stage prove superfluous, but any overt bias, such as "add TV," would fail to influence human judgment in the final classification, rendering the counterfactual explanation largely irrelevant to human users.

5.5 Conclusion

In this chpater we have developed a novel explainability framework that leverages the robustness of knowledge graphs for generating counterfactual explanations using the relational information between the objects. This framework ensures that the explanations are not only valid and feasible—always reflecting edits towards real data points—but also minimal, due to the incorporation of edit distance computations. Furthermore, the explanations are actionable, thanks to the manual assignment of edit costs. Our human study indicates that these counterfactual explanations are understandable and meet the satisfaction of end-users.

The framework, however, relies heavily on the dataset used for explanations, which should ideally be curated by domain experts. In critical fields like medicine, the investment in expert curation is justified by the benefits. For less critical applications, we have demonstrated that utilizing semantically rich datasets such as the Visual Genome, or employing automatic knowledge extraction methods like scene graph generation, can also yield valuable explanations.

Additionally, we introduced a model-agnostic approach for computing counterfactuals using the expressive capabilities of semantic graphs. This involved the innovative use of a GNN-based similarity model to facilitate the GED calculation, which accelerates the typically NP-hard process of retrieving counterfactuals across all input graph pairs. Our evaluations indicate that our model not only ensures minimal and actionable edits but also enhances human interpretability, particularly in scenarios where concept interactions are densely packed. Additionally, it outperformed a state-of-the-art algorithm in calculating semantic counterfactuals for images in a white-box manner, as evidenced by user preferences and human understandability in a series of human surveys.

Looking ahead, there is considerable potential for advancing this research. We plan to enrich the framework with more comprehensive knowledge and incorporate theoretical insights from description logics and reasoning. Furthermore, we are exploring the use of generative models capable of applying semantic edits to a data sample to produce new samples that can be evaluated by the classifier. We also aim to address potential limitations related to the robustness of counterfactual explanation methods and the impact of low-quality annotations. Enhancing efficiency through unsupervised GNN methods represents another promising avenue for future work. This comprehensive approach will continue to refine the applicability and effectiveness of our framework across various datasets and conditions. Finally, as part of our next steps, we plan to extend the human-machine teaching experiment to different datasets, such as those in the audio domain, in order to further validate the hypothesis that semantic explanations enhance human understanding.

Chapter 6

Optimal and Efficient Text Counterfactuals using GNN

6.1 Introduction

In the preceding chapters, a framework and algorithms for retrieving counterfactual explanations in a modality-aware manner within a black box setting have been demonstrated. Emphasis has been placed on image and audio classifiers, yet the prevalence of classifiers in the text domain underscores the pressing necessity to modify the proposed framework to better accommodate textual modalities.

A significant challenge has been identified due to the presence of highly advanced language models capable of generating high-quality counterfactual explanations with ease. The existing framework, which does not generate new instances but rather searches within a dataset to find the semantically closest instance, is notably limited. This limitation is further exacerbated in the text modality for several reasons. First, the transformation of text into a graph format poses a significant challenge [339]. Furthermore, the meaning of a word in text is highly dependent on its surrounding words, a constraint not applicable to scene graphs, where the interpretation of objects remains constant regardless of proximity to other items.

Inspired by the shortcomings of the current framework, in this chapter we propose an efficient algorithm for generating optimal text counterfactuals using GNNs [222]. This method can be specifically targeted to a classifier or employed in general-purpose scenarios without any classifier, using only a dataset. The results from the experiments have indicated that this method outperforms state-of-the-art classifiers in four critical metrics—minimality, fluency, closeness, and runtime—across two distinct datasets. Remarkably, it achieves these results in less than 2% and 20% of the time required by its two competitors, demonstrating both superior efficacy and efficiency.

6.2 Realated work

Since the introduction of the Transformer model [350], the field of NLP has witnessed a significant expansion in its capabilities, addressing a wide array of linguistic tasks. Interest in explainability [9, 53] and interpretability [230] has surged, focusing on identifying biases and spurious correlations that affect the generalization of state-of-the-art models. Additionally, adversarial attacks [407] have shed light on the inner workings of these models, enhancing post-hoc interpretability by triggering alternate outcomes.

A number of studies have explored adversarially perturbed inputs aimed at label flipping [240, 252, 182, 303], while others have attempted more generalized approaches to perturbation [305, 374]. These methods, despite producing linguistically promising results, are often computationally intensive and slow, with some requiring over 47 hours to process 1,000 samples (see Table 6.1) [303]. Furthermore, the transition from generalized text manipulation to targeted interpretability necessitates a more controlled generation process, as the opacity of general-purpose editing tools based on large language models (LLMs) frequently results in suboptimal substitutions [80].

Research on exposing vulnerabilities in state-of-the-art models through adversarial or counterfactual inputs remains robust, with perturbations ranging from the character [69] to the word [91, 295] and sentence levels [128]. Our project focuses on semantic changes at the word level, adhering to established norms for word-level perturbations.

The manual and automated creation of adversarial examples has been pursued [90, 147, 254], with early methods using paraphrasing [127] and more recent approaches employing masked language modeling techniques [177, 303, 182]. Techniques leveraging similarity-driven substitutions based on word embedding distances [132, 424] optimize local accuracy for classification tasks while ensuring the controllability of adversarials [252]. These model-specific methods partially align with our approach but are limited in scope.

General-purpose counterfactual generators that fine-tune LLMs to offer diverse perturbations applicable at multiple levels of granularity [374, 96, 305] open new avenues for textual counterfactuals. However, these methods often compromise on explainability due to the unpredictable nature of LLM decision-making [42, 309]. Conversely, recent advances in graph-related optimization strategies [396, 220] showcase promising results, balancing performance, explainability, and computational efficiency in linguistic interventions.

To enhance the transparency and utility of adversarial examples, integrating hybrid approaches that combine the interpretive strengths of graph-based methods with the generative capabilities of LLMs could offer a more nuanced and effective means of generating adversarial texts. This innovative direction could lead to more reliable and comprehensible adversarial inputs, narrowing the gap between current limitations and the ideal of fully transparent NLP models.

In the current study, we explore the impact of altering specific words on the performance of textual classifiers through what we refer to as word-level counterfactual interventions. Our methodology is defined by a structured framework with key attributes aimed at optimizing these interventions. Each substitution should achieve or closely approximate the best possible outcome while maintaining a predefined measure of semantic closeness, ensuring optimality. Additionally, there should be at least one semantic input modification in every dataset instance to maintain controllability. Lastly, the ideal solution should be obtained through streamlined search methods rather than exhaustive exploration of all possible alternatives, enhancing efficiency.

To address these principles, we treat counterfactual interventions as a problem of combinatorial optimization. This challenge is tackled using graph assignment techniques derived from graph theory [386]. Moreover, to augment our strategy, we incorporate Graph Neural Networks (GNNs) [378] as a more rapid, albeit approximate, alternative to traditional graph-based methods [393].

Our innovative approach is designed to be versatile, suitable for both specific model applications and broader general uses, without necessarily altering the final classification output. This flexibility enables the adaptations not only for tasks such as label-flipping but also for assessing semantic similarity [220] and generating content without specific targets [374]. Although our focus here is primarily on classification tasks to allow direct comparisons with existing methodologies, we evaluate our system against two state-of-the-art text editing frameworks [374, 303] on metrics like label-flipping accuracy, linguistic fluency, and semantic proximity. This comprehensive analysis aims to establish a robust baseline for the efficacy of counterfactual interventions in text classification scenarios.

6.3 Algorithm for Generating Text Counterfactuals Using GNNs

6.3.1 Problem Formulation

The approach is based on a graph-based framework in which words from sentences are mapped onto nodes, while the costs of substituting one word for another are assigned to the edges connecting these nodes. A bipartite graph, denoted as $G = (V, E)$, is considered, where the set of nodes V is divided into two distinct groups: the source nodes S with $|S|$ and the target nodes T with $|T|$, ensuring that $S \cup T = V$ and $S \cap T = \emptyset$. This setup is essential for addressing the discrete optimization challenge of finding the most efficient connections between nodes within G. The focus is on establishing a minimum weight matching $M \subseteq E$ where the sum of the edge weights, $\sum w_e, w_e > 0$ in W, is minimized for edges $e \in E$ that cover the smaller of the two node sets $\min(|S|, |T|)$. It is ensured that if $|S| \leq |T|$, a connection or substitution from each node in S to a different node in T is possible.

The mathematical formulation of this optimization problem is expressed as follows:

$$\min \sum w_e, \text{ subject to } s \neq t \text{ if } \exists e_{s \to t}$$

Traditionally, the problem could be tackled by an exhaustive search, where all possible permutations of (s, t) pairs are considered, and every permutation of T is evaluated until the minimal sum $\min \sum w_e$ is reached. However, this method results in an exponential complexity of $O(m^n)$, assuming a complete graph where each s is connected to every t, hence $E = S \times T$ with $|E| = nm$.

An efficient approach is achieved by treating the problem as a variant of the rectangular linear assignment problem (RLAP), in which n source nodes are assigned to $m \geq n$ target nodes to minimize the total weight of the connections. The RLAP not only provides a framework for finding optimal solutions but also allows flexibility through multiple possible matchings for each source node s. By employing algorithms adapted from foundational literature, this problem is addressed with significantly improved efficiency, reducing the complexity to $O(mn \log n)$, a marked improvement over the exponential complexity of more naive methods. These algorithms have been continually refined and adapted to efficiently meet the specific requirements of the RLAP.

6.3.2 Graph neural network for RLAP

Graph Neural Networks (GNNs) [317] have become an indispensable tool for analyzing and learning from data that exhibits intrinsic graph structures, effectively encapsulating the relationships among diverse entities. These networks are particularly effective in scenarios where the representation of data as graphs is natural. For example, in the domain of linear assignment problems, GNNs are utilized to address the linear sum assignment problem (LSAP), where n agents must be matched to n jobs under unique pairing constraints, with the goal of minimizing the total cost [29].

Building on this premise, we have refined the model by implementing a Graph Convolutional Network (GCN) [156] to deal with the Relaxed Linear Assignment Problem (RLAP). This innovation represents a pioneering use of GNN frameworks to resolve RLAP, filling a gap in existing literature. The architecture of our tailored GCN model comprises three integral parts: an encoder, a convolution module, and a decoder. This structure facilitates iterative updates of node attributes through multiple phases, thereby improving the model's effectiveness and precision in solving assignment challenges [205].

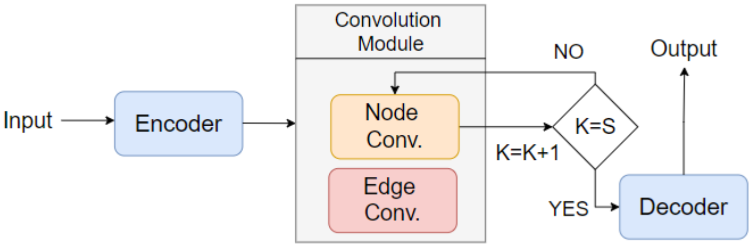


Figure 6.3.1: Overview of the GNN architecture used. Attributes at each node are updated over $S \geq 2$ iterations in the node convolution layer.

Encoder/Decoder

Given a bipartite graph G, the encoder employs a Multi-Layer Perceptron (MLP) applied to every edge, converting raw attributes into latent embedding features. Initially, each edge $i \to j$ is represented by $e_{ij} = w_{ij}$, where w_{ij} is the edge's weight. Meanwhile, nodes start with zero-valued attribute vectors. After encoding, the transformed graph is fed into the convolution module for attribute updates.

On the decoder side, we retrieve the updated edge attributes from the output of the convolution module and use another MLP-based transformation to predict edge labels through a sigmoid function. In other words, each edge's final attribute is passed through a learnable update function that produces a binary label corresponding to whether or not the edge is part of the solution.

The Convolution Module

This module consists of two distinct layers: a *node convolution* layer and an *edge convolution* layer. The node convolution layer updates a node's attributes by collecting information from its connected edges and immediate neighbors with learnable aggregation weights. The edge convolution layer, in turn, refines the attribute vector of each edge by combining the attributes of the two nodes it links.

In a bipartite graph with sets S and T (see Section 6.3.1), each node in S connects to all nodes in T, so messages from one node can propagate to every other node after two iterations of convolution. Consequently, although the convolution layer technically considers only first-order neighborhoods, the structure of a bipartite graph ensures the entire network becomes reachable within two passes.

Edge Convolution. For an edge $i \to j$, we first gather information from the two endpoints via:

$$\overline{e}_{ij} = [\, v_i \odot c^u,\ v_j \odot c^u,\ e_{ij} \odot c^e \,], \tag{6.3.1}$$

where v_i and v_j are node attribute vectors for nodes i and j, respectively; c^u and c^e are channel attention vectors for node and edge features (matching the dimensionality of v_i/v_j and e_{ij}). The symbol $\odot$ denotes element-wise multiplication, and $[\cdot,\cdot,\cdot]$ indicates concatenation. Note that $\overline{e}_{ij}$ is an intermediary vector unifying node and edge features; it is then passed to an MLP-based update function $\rho^e(\cdot)$, giving $e_{ij} \leftarrow \rho^e(\overline{e}_{ij})$.

Node Convolution. For a node v_i, we aggregate features from its incident edges and adjacent nodes:

$$\overline{v}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \rho_1^v\big([\, e_{ij} \odot c^e,\ w_{ij}(v_j \odot c^u)]\big), \tag{6.3.2}$$

$$e_{ij} \in \mathcal{E}_i, \quad v_j \in \mathcal{V}_i,$$

where ρ_1^v is an MLP that generates embedding features, $\mathcal{E}_i$ is the set of edge attributes for edges incident to node v_i, and $\mathcal{V}_i$ is the set of its first-order neighbors. The term w_{ij} weighs the importance of neighbor v_j when gathering features and is itself computed by another MLP, $\tau([v_i, v_j])$. After computing $\overline{v}_i$, we concatenate it with the original v_i and update:

$$v_i \leftarrow \rho_2^u\big([\overline{v}_i,\ v_i]\big),$$

where ρ_2^u is an MLP. All functions ρ_1^v, ρ_2^u, and τ are implemented as MLPs with distinct architectures and parameters.[1]

6.4 Counterfactual generation overview

The methodology described in the study encompasses three distinct phases (Figure 6.4.1). Initially, a textual corpus designated as D is used, from which words are extracted based on their grammatical category, forming the foundational node set S. The corresponding target node set T either mirrors S or is derived from an external linguistic resource like WordNet [243], aggregating all potential replacement candidates for the elements in S. Together, the node sets S and T construct a bipartite graph G, as discussed in Section 6.3.1. The edges within this graph are designed to signify the semantic proximity between words in the source and target sets.

Proceeding to the second phase, the bipartite graph G is fed into a trained Graph Convolutional Network (GCN). This network processes the graph and delivers an approximate solution to the Relaxed Linear Assignment Problem (RLAP), represented by a series of candidate word pairings. Each pairing consists of a source word from set S and a suggested substitute from set T.

[1] For comprehensive details, see [205], which describes the underlying model hyperparameters in depth.

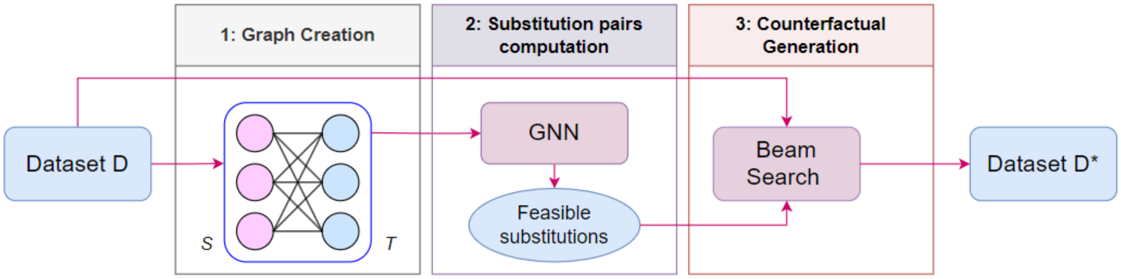


Figure 6.4.1: An overview of our approach. First, we build a bipartite graph whose nodes represent individual words. Next, we apply a Graph Neural Network (GNN) to find plausible substitutions that effectively approximate the RLAP. Finally, by running beam search on the original dataset, we selectively replace certain words to generate a new counterfactual dataset.

In the culmination of the process, the third phase involves the employment of a beam search algorithm. This algorithm utilizes a heuristic function to meticulously select the most appropriate substitutions from the array provided by the GCN. The chosen substitutions are applied, with words from S replaced by their counterparts in T, resulting in a modified dataset, denoted as $D*$. This counterfactual dataset serves as the output of the method, presenting a systematically altered version of the original corpus based on the semantic relationships and substitutions identified through the structured workflow.

6.4.1 Construction of Bipartite Graph

In the development of the bipartite graph G, we initiate by extracting words from the document D focusing on their parts of speech (POS). This process is pivotal for examining the adaptability of our model across different settings. We employ two distinct methods for word extraction: POS-specific and POS-agnostic approaches. In the POS-specific method, word selection for potential modification is confined to words that fall under certain POS categories such as adjectives, nouns, and verbs. On the other hand, the POS-agnostic method considers all words equally, regardless of their POS classification.

For assigning weights to the edges of the graph, we explore two contrasting methodologies, each differing in their level of transparency and methodological approach. Initially, we use a straightforward method by leveraging a lexical hierarchy to calculate distances. Specifically, the edge weight between two words is determined by their semantic proximity, as gauged by the similarity value provided in WordNet.[2] For our second approach, we employ various large language models (LLMs) to produce word embeddings, including AnglE[3], GISTEmbed[4], GinaAI[5], and MUG[6]. Here, the weight of an edge is set based on the cosine similarity between the embedding vectors of the two words.

The design ensures that lower similarity scores—which correspond to lighter edges—are favored, thereby forming *contrastive word pairs* for substitutions. This selection criteria is instrumental in identifying prime candidates for word substitutions in M.

To maintain syntactic integrity, particularly in the POS-agnostic method, we implement an edge filtering system. This system involves setting a predetermined, significantly larger weight to edges—approximately ten times greater than the normal weights derived from WordNet path similarity or cosine similarity of embeddings. This strategy effectively prevents inappropriate POS substitutions by excluding heavily weighted edges from selection in M. In contrast, the POS-specific method does not require this filtering mechanism as all words under consideration already share the same POS, ensuring syntactic consistency without additional constraints.

[2] This utilizes the ‘path_similarity‘ function between synsets corresponding to the words, as detailed here: https://www.nltk.org/howto/wordnet.html.
[3] Details on this model can be found at: mixedbread-ai/mxbai-embed-large-v1
[4] Further information is available at: avsolatorio/GIST-Embedding-v0
[5] More on GinaAI embeddings can be found here: https://jina.ai/embeddings/
[6] For more details, refer to: Labib11/MUG-B-1.6

6.4.2 Substitution pairs computation

To identify suitable substitution pairs, we address the Rectangular Linear Assignment Problem (RLAP) on a specifically constructed graph G. As discussed earlier (see Section 6.3.1), conventional deterministic methods can solve this with a complexity of $O(mn \log n)$. Although these techniques ensure the most accurate solution, their performance diminishes as the size of the dataset—and consequently, the graph—increases. To generate substitution pairs in a time-consistent manner irrespective of dataset size, we employ a Graph Neural Network (GNN) model (refer to Section 6.3.2). This model offers an approximation to the optimal solution traditionally obtained by deterministic methods but does so with markedly improved speed. This approach ensures **efficiency** by speeding up the computation process.

The GNN model tackles the RLAP by focusing on minimizing the sum of $\sum w_e$, effectively identifying the most *dissimilar* $s \rightarrow t$ pairs. This process achieves an *approximate* **optimality** in the substitution of concepts within G, thus generating useful contrastive substitution pairs. Additionally, **controllability** is *partially* maintained because the graph G is characterized by its density—there are no isolated s nodes, and the condition $|S| \leq |T|$ holds true, with T mirroring S or being derived from S using antonyms sourced from WordNet (each word might correspond to multiple antonyms). The use of the term *"partially"* highlights the inherent trade-off between *controllability* and *minimality*—the latter referring to the minimal number of word changes needed (see Section 6.5.2. This trade-off arises from employing a beam search strategy during the generation of counterfactuals. It is important to note that in certain cases, this *controllability* may be compromised if a source concept does not align well with the definitions in WordNet.

6.4.3 Counterfactual Generation

As a result of solving RLAP, a specific subset of matches, denoted as $M \subset E$, is derived. This matching is crucial as it represents the optimal substitutions for n source concepts within a dataset. The cumulative weight of these matches, represented as W_n^M, plays a significant role in the selection process that follows. Essentially, W_n^M includes the total weight of all substitutions that involve the n identified source concepts.

Selection Process Following the identification of the optimal matches, the next critical step involves the selection of which conceptual substitutions from M will be implemented on the dataset D. This selection is executed via a beam search strategy, a method well-suited for sifting through a multitude of options and narrowing them down to the most pertinent substitutions. The criteria for this selection are meticulously set to ensure that only minimal textual alterations are made. The aim is to adjust as few words as possible in each instance, thereby causing only slight perturbations to the input data. Such minimalistic changes are preferred because they help maintain the clarity and intelligibility of the explanations provided, as suggested by prior studies like Alvarez-Melis and Miller (2019).

Setting Limits on Substitutions An integral part of this process involves setting limits on the number of substitutions permissible for each text instance within the dataset. This is done in two distinct ways: one approach involves fixing a maximum number of substitutions per instance, while the other adopts a dynamic strategy where the limit is proportional to the text length. Specifically, in the dynamic method, the upper limit for substitutions is set at 20% of the total word count of each instance. This approach ensures a balanced modification of the text, preventing excessive alterations that could compromise the original context or meaning.

Termination Criteria The termination of the search and selection process is contingent upon achieving one of two outcomes: either the model's prediction is altered (flipped), or the predefined upper limit of substitutions is reached. This termination protocol is crucial as it ensures the edits remain within a manageable scope, thereby preserving the essential characteristics of the original text while still introducing the necessary conceptual shifts. This methodical limitation of edits is fundamental in maintaining the effectiveness and efficiency of the counterfactual generation process.

6.5 Experiments

In this section, the presentation of experiments is conducted along with the corresponding results. These outcomes demonstrate that the proposed framework is capable of producing fluent, minimal edits while achieving a high percentage of label-flipping in a significantly shorter duration when compared to alternative editing frameworks. The experimental suite was executed on a uniform system setup, which included a *16 GB GPU*, an *Intel i7 CPU*, and *16 GB of RAM.*

6.5.1 Experimental Setup

Datasets The evaluation of the framework was conducted against other editors documented in the literature, utilizing two datasets in the English language: the IMDB dataset, which comprises movie reviews for binary sentiment classification [226], and a six-class variant of the 20 Newsgroups dataset used for topic classification [168]. Owing to the substantial computational requirements imposed by the methods being compared, a sample of 1K instances from each dataset was selected for evaluation. The execution of MiCE on merely 1K samples necessitated over 47 hours (refer to Table 6.1), rendering full dataset experiments unfeasible. This sample size was determined to be double that utilized in comparable studies, which examined the same methods on identical datasets [80].

Predictors In the research conducted, the performance of certain edits is assessed using predictive models aligned with the methodologies described by [303]. These models, which are built upon the foundations of RoBERTa$_{LARGE}$ [210], demonstrate a test accuracy of 95.9% and 85.3% on the IMDB and Newsgroups datasets respectively. The evaluation of these models has been executed passively, with the same predictor models being employed across each dataset under investigation.

Editors As for the editing frameworks, a comparison was drawn between the existing framework and two state-of-the-art editors, MiCE [303] and Polyjuice [374]. It was observed that MiCE tailored its edits towards minimal adjustments aimed specifically at label-flipping, whereas Polyjuice provided edits that were not limited to any singular task and were more generalized. The framework being assessed utilized a deterministic RLAP solution as a baseline, against which the GNN RLAP optimization was compared. To further explore the generalization capabilities of the framework, both POS-restricted and POS-unrestricted substitutions were employed. Further information and analysis of these editors can be found in Chapter 8, Section 7.4.

Metrics The effectiveness of various editors was gauged through several metrics inspired by MiCE. These metrics include:

1. **Flip-rate:** This metric is quantified as the proportion of instances where an edit leads to a change in the model's prediction, thereby causing label-flipping. The calculation of this rate was done passively.
2. **Minimality:** Defined as the "size" of an edit, this is measured using the word-level Levenshtein distance between the original input and the edited version. This distance is then normalized on a scale from 0 to 1, calculated as the ratio of the Levenshtein distance to the number of words in the original input.
3. **Closeness:** The semantic similarity between the original and edited input is measured using the BERTscore [405].
4. **Fluency:** This is evaluated by comparing how the distribution of the edited input aligns with that of the original. Initially, a pretrained T5-BASE model [286] is used to compute the loss value for both the original and edited input. Subsequently, the $loss\_ratio$ (i.e., $edited/original$) is reported. An ideal fluency score is aimed at achieving a $loss\_ratio$ of 1.0, which would indicate equivalent losses for both texts, thereby defining the fluency metric as $|1 - loss\_ratio|$.

Further information and analysis of these metrics can be found in Chapter 8, Section 7.2.

6.5.2 Results

In the provided document, the experimental outcomes are depicted in Table 6.1, encompassing data from both the IMDB and Newsgroups datasets. Comprehensive analyses are made accessible in Section, 6.5.2.

				IMDB		
	Editor	**Fluency ↓**	**Closeness ↑**	**Flip Rate ↑**	**Minimality ↓**	**Runtime ↓**
Wordnet	Deterministic w. fluency	0.14	0.969	0.892	0.08	4:09:41
	GNN w. fluency	0.07	0.986	0.861	0.12	3:17:51
	GNN w. fluency & dynamic thresh	0.057	0.986	0.851	0.146	4:18:34
	GNN w. fluency & POS_filter	0.08	0.992	0.862	0.123	**0:32:05**
	GNN w. fluency & edge filter	0.105	0.993	0.845	0.149	3:00:38
	GNN w. fluency_contrastive	0.112	**0.999**	0.914	0.014	2:12:06
	GNN w. contrastive	0.048	0.996	0.927	**0.01**	2:00:15
Emb.	GNN w. AnglE & contrastive	0.063	0.995	0.944	0.011	0:45:38
	GNN w. GIST & contrastive	0.037	0.995	0.882	0.016	0:58:14
	GNN w. Jina & contrastive	0.047	0.995	0.928	0.017	1:00:56
	GNN w. MUG & contrastive	**0.036**	0.996	0.889	0.013	0:52:19
	Polyjuice	0.394	0.787	0.782	0.705	5:01:58
	MiCE	0.201	0.949	**1.000**	0.173	48:37:56

				Newsgroups		
	Editor	**Fluency ↓**	**Closeness ↑**	**Flip Rate ↑**	**Minimality ↓**	**Runtime ↓**
Wordnet	Deterministic w. fluency	0.182	0.951	0.870	0.135	4:20:52
	GNN w. fluency	0.074	0.985	0.826	0.151	3:48:37
	GNN w. fluency & dynamic thresh	0.043	0.984	0.823	0.148	4:47:14
	GNN w. fluency & POS filter	0.044	0.989	0.841	0.143	1:19:57
	GNN w. fluency & edge filter	0.12	0.989	0.834	0.151	3:05:08
	GNN w. fluency_contrastive	0.088	0.979	0.875	0.033	2:45:31
	GNN w. contrastive	0.033	0.989	0.920	0.033	2:02:34
Emb.	GNN w. AnglE & contrastive	0.005	0.995	0.904	0.027	1:09:13
	GNN w. GIST & contrastive	**0.001**	0.995	0.898	0.02	1:02:55
	GNN w. jina & contrastive	0.013	0.993	0.882	0.025	0:57:31
	GNN w. MUG & contrastive	0.005	**0.996**	0.900	**0.016**	**0:53:04**
	Polyjuice	1.153	0.667	0.8	0.997	6:00:10
	MiCE	0.152	0.922	**0.992**	0.261	47:23:35

Table 6.1: Experimental results of counterfactual generation. We evaluate different versions of our framework using the metrics described on subsection 6.5.1, and we compare it with MiCE and Polyjuice. For each metric (column) the best value is highlighted in **bold**. Reported runtimes refer to inference.

It has been found that the proposed editors—equipped with deterministic and GNN mechanisms—surpass MiCE and Polyjuice in three of the four evaluated metrics, namely **minimality**, **fluency**, and **closeness**. In terms of the flip-rate metric, it is reported that MiCE secures the highest results, achieving between 99% and 100% across the two datasets. In contrast, the optimal editor from the current study demonstrates slightly lower yet significant flip-rate values of 94.4% for IMDB and 92% for Newsgroups. The superior performance of MiCE in this metric is anticipated due to its white-box access to the classifier, allowing it to craft edits that significantly influence the classifier's response, irrespective of the input text's content.

Further, results indicate a greater degree of minimality in edits when the graph construction relies on embedding models rather than WordNet. It is observed that the usage of WordNet leads to modifications involving approximately 10% of the original tokens, whereas embedding models necessitate changes to merely 1% of the tokens. This distinction is attributed to the state-of-the-art (SoTA) embedding models' superior capability to accurately reflect concept distances, thereby facilitating higher quality substitutions and generating more contrastive pairs. Such outcomes imply that fewer embedding-based substitutions are needed to achieve the same level of impact on the classifier's output compared to those based on WordNet. However, the adoption of embedding models slightly diminishes the transparency of the method. Despite these minor variances, all variants of the current framework consistently surpass previous techniques across all metrics for Polyjuice and three metrics for MiCE. Furthermore, even the general-purpose variant of the framework, which lacks direct access to the classifier, still manages to produce better outcomes than the white-box MiCE, achieving this in just 2% of the time.

Regarding runtime, a marked improvement is noted for the editors proposed in this study compared to MiCE and Polyjuice. The deterministic editor, serving as a baseline, requires roughly 4 hours for processing each dataset. Editors incorporating the GNN discussed in Section 6.3.2 demonstrate enhanced performance, averaging between 2 to 4 hours. This efficiency is significantly boosted through the integration of embedding models, where processing times are reduced to under an hour (52 minutes to 1 hour for IMDB, and between 53 minutes to 1 hour and 9 minutes for Newsgroups). This substantial enhancement in speed represents a major advantage of the proposed framework over the two SoTA editors, demonstrating speed improvements of approximately 97% and 83% in comparison to MiCE and Polyjuice, respectively.

Static vs. Dynamic Threshold It is observed that to maintain a low count of modifications, a mechanism is required to restrict the number of changes per data instance, even if it results in a decrease in flip-rate. Two distinct strategies are employed for this purpose. A static cap on substitutions is imposed in the first strategy, allowing a maximum of 10 substitutions per textual input irrespective of its length, as determined through empirical testing. The second strategy involves dynamically calculating an optimal cap on substitutions, which is defined as 20% of the total word count of the text after several trials. The outcomes indicate a negligible enhancement in metrics with the application of a dynamic threshold, although the runtime extends by about one hour per dataset. This increase in time is anticipated as the dynamic threshold introduces additional linear complexity for each text instance, compared to the constant time complexity ($O(1)$) in the static method. Unless specified otherwise, the static threshold remains the default approach.

POS-restricted vs. Unrestricted Substitutions An evaluation is conducted to ascertain the capability of the editing tool in recognizing which parts of speech (POS) predominantly influence a specific dataset when substitutions of related words are made. Restrictions are imposed on which POS can be candidates for substitutions, and the results are compared with those of an unrestricted version of the framework. For sentiment classification using the IMDB dataset, it is presumed that adjectives and adverbs primarily influence the sentiment label of each instance, leading to restrictions being placed on altering only these two POS. Conversely, for the Newsgroups dataset, which falls under topic classification, nouns are deemed crucial in deducing the topic, prompting instructions for the editor to consider only nouns for substitutions. Results documented in Table 6.1 reveal that editors, both with and without POS filtering, exhibit remarkably similar outcomes for both the IMDB and Newsgroups datasets. This similarity suggests that the lack of significant differences is not contingent upon specific POS restrictions. A notable disparity is observed in runtime, where restricted editors require 32 to 60 minutes, whereas unrestricted editors take 2 to 4 hours. This difference is expected, as focusing on specific POS at any one time reduces the number of words considered as substitution candidates, significantly decreasing the number of graph nodes and edges, thus reducing the time required for graph construction and GNN inference.

Edge Filtering In the interest of preserving the POS during each substitution, a penalty mechanism is applied when computing edge weights in the graph. This mechanism assigns a weight approximately 10× greater than the normal weights, as determined by WordNet path similarity or embedding cosine similarity, to edges that connect words of different POS. Consequently, edges bearing high weights are nearly impossible to select in the search for a minimum weight matching, rendering substitutions involving different POS highly improbable. Examination of results with and without edge filtering suggests minimal differences, leading to the supposition that the utility of such a mechanism might be redundant, given that the functionalities appear to be subsumed by the GNN's solution to the graph assignment problem.

Contrastive vs. Fluent Contrastive Edits The behavior of the editing tool is investigated under conditions optimized for label-flipping scenarios versus general-purpose editing. The heuristic function of the beam search in the final stage of the framework is modified accordingly (as illustrated in Figure 6.4.1). For general-purpose edits, the heuristic is based on the metric of fluency discussed in Subsection 6.5.1, which aids in producing fluent edits. For label-flipping, the heuristic employs contrastive probability, which evaluates the alteration in model prediction for the original label, to identify the most effective edits (as indicated by *GNN w. contrastive* in Table 6.1). Additionally, an average of fluency and contrastive probability is utilized as the heuristic, resulting in fluent edits that exhibit a high flip-rate (as shown by *GNN w. fluency_ contrastive* in Table 6.1). While general-purpose edits achieve the lowest flip-rate, they still surpass Polyjuice, another general-purpose editor, in all evaluated metrics. This demonstrates that the framework can also serve as a versatile, untargeted editor producing high-quality edits; further extensive testing on this assertion is deferred to future studies. Conversely, the label-flipping optimized edits display superior results in terms of fluency, closeness, and minimality when compared to MiCE, a state-of-the-art white-box editor optimized for label-flipping. In terms of flip-rate, MiCE shows better performance by 7%, albeit at a considerable cost of a 20-fold increase in execution time.

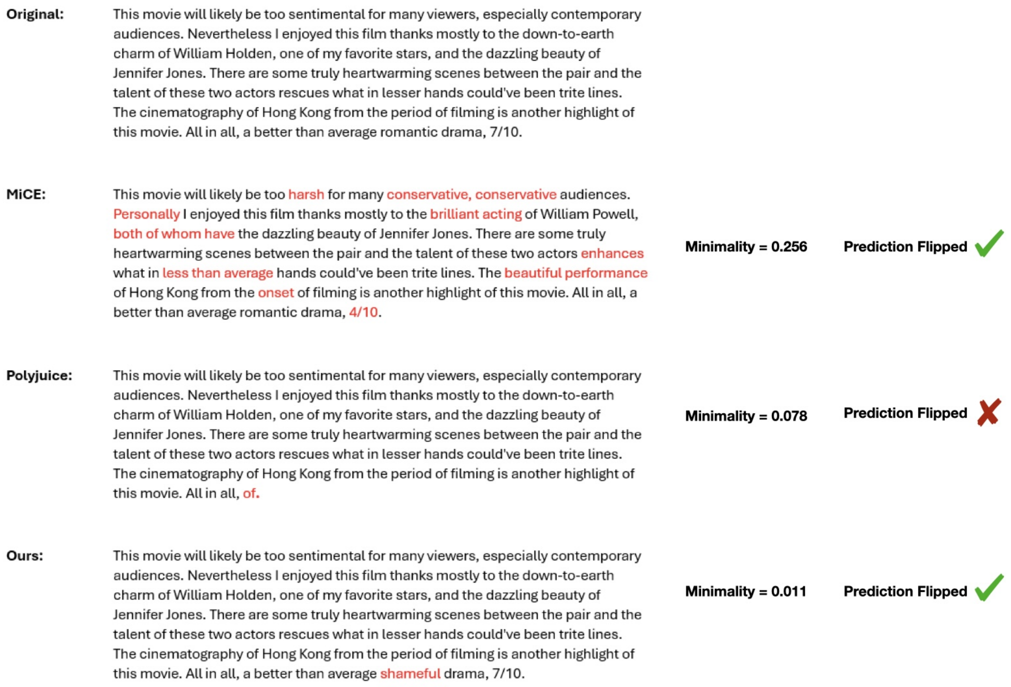
WordNet vs. Embeddings The impact of substituting WordNet path similarity with cosine similarity of embeddings is explored when determining the weight of specific edges in the bipartite graph G. On one side, deterministic hierarchies offer more explainable relationships between concepts, fully justifying the causal pathways of substitutions. On the other side, recent embedding models are likely to capture the relationship and similarity between two words more effectively than WordNet. To maintain a manageable framework size, the top four best-performing models from an embedding benchmark competition are employed, provided their size does not exceed 1.25 GB. These models, which rank highly in the competition, do not show significant performance improvements with an increase in size. The results support the hypothesis, with variants utilizing embedding models outperforming those based on WordNet in all metrics. Concerning GPU inference, the embedding models also demonstrate faster performance than WordNet, which requires API calls for each word/graph node, significantly slowing down the graph creation process.

Edits Comparison Between Editors

In order to showcase our editor's advantages in terms of minimal edits and successful label flipping, we perform a qualitative comparison against two other text-editing approaches—Polyjuice and MiCE. To that end, we select a sample from the IMDB dataset that is initially predicted as *positive* by the classification model. We then generate revised versions of this sample using our own editor, as well as the two baseline editors. Since Polyjuice is specifically designed to invert the sentiment from *positive* to *negative*, we employ its `[negation]` control code, which instructs the system to introduce negation cues into the text, thereby prompting a label shift.

Figure 6.5.1 presents the original text alongside the modified outputs, with changes highlighted in red. By examining these snippets, we can assess the number and nature of the edits introduced by each editor. This visual check allows us to measure how each system balances minimality of edits against semantic preservation.

As illustrated in the figure, MiCE makes the largest number of modifications, including at least two phrases that conflict with the intended semantics of the passage: *"conservative, conservative"* and *"both of whom have"*. Furthermore, MiCE frequently opts for multi-word replacements rather than concise single-word edits. These choices undermine the principle of *minimality*, which emphasizes making only the smallest necessary changes.



Original: This movie will likely be too sentimental for many viewers, especially contemporary audiences. Nevertheless I enjoyed this film thanks mostly to the down-to-earth charm of William Holden, one of my favorite stars, and the dazzling beauty of Jennifer Jones. There are some truly heartwarming scenes between the pair and the talent of these two actors rescues what in lesser hands could've been trite lines. The cinematography of Hong Kong from the period of filming is another highlight of this movie. All in all, a better than average romantic drama, 7/10.

MiCE: This movie will likely be too harsh for many conservative, conservative audiences. Personally I enjoyed this film thanks mostly to the brilliant acting of William Powell, both of whom have the dazzling beauty of Jennifer Jones. There are some truly heartwarming scenes between the pair and the talent of these two actors enhances what in less than average hands could've been trite lines. The beautiful performance of Hong Kong from the onset of filming is another highlight of this movie. All in all, a better than average romantic drama, 4/10.

Minimality = 0.256 **Prediction Flipped** ✓

Polyjuice: This movie will likely be too sentimental for many viewers, especially contemporary audiences. Nevertheless I enjoyed this film thanks mostly to the down-to-earth charm of William Holden, one of my favorite stars, and the dazzling beauty of Jennifer Jones. There are some truly heartwarming scenes between the pair and the talent of these two actors rescues what in lesser hands could've been trite lines. The cinematography of Hong Kong from the period of filming is another highlight of this movie. All in all, of.

Minimality = 0.078 **Prediction Flipped** ✗

Ours: This movie will likely be too sentimental for many viewers, especially contemporary audiences. Nevertheless I enjoyed this film thanks mostly to the down-to-earth charm of William Holden, one of my favorite stars, and the dazzling beauty of Jennifer Jones. There are some truly heartwarming scenes between the pair and the talent of these two actors rescues what in lesser hands could've been trite lines. The cinematography of Hong Kong from the period of filming is another highlight of this movie. All in all, a better than average shameful drama, 7/10.

Minimality = 0.011 **Prediction Flipped** ✓

Figure 6.5.1: Original input and edited inputs from different editors. The changes that each editor performed are highlighted in red color.

On the other hand, Polyjuice makes a single alteration at the end of the text. While this satisfies a minimal change requirement, the substituted token itself does not carry meaningful semantic content, instead functioning merely as a forced trigger to flip the classification label. Such edits can be conspicuous, signaling the involvement of a neural model or an automated counterfactual editor, which stands in contrast with the ideal of "imperceptible edits" in counterfactual scenarios.

Our editor offers the most balanced and precise result among the three. By modifying only one word, it shifts the sentiment from *positive* to *negative* without introducing unnatural or semantically problematic text. This single, contextually coherent edit meets the primary counterfactual objective while preserving the overall structure and readability of the original instance, thus exemplifying both *minimality* and *semantic fidelity*.

Edits	Minimality ↓	Prediction Flipped
Polyjuice	0.078	False
MiCE	0.256	**True**
Ours	**0.011**	**True**

Table 6.2: Comparison of the performance metrics for the edits displayed in Figure 6.5.1. Each column illustrates a particular property, and the best-performing value for each is shown in **bold**.

We present the numerical outcomes for the examples in Figure 6.5.1, focusing on *minimality* and *label-flipping*, in Table 6.2. Since the experiment involves only a single text sample, we employ the term *prediction flipped* (rather than *flip-rate*) to indicate whether the edited version successfully alters the classifier's initial prediction. Notably, Polyjuice is unable to achieve a prediction flip, whereas both MiCE and our approach succeed in doing so. Furthermore, our editor demonstrates the strongest performance in terms of minimality, with Polyjuice placing second and MiCE being the weakest among the three.

6.6 Trade-offs

Given the highly customizable nature of our editor, numerous trade-offs must be navigated during the generation of counterfactuals.

Controllability vs. Minimality Our approach to controllable interventions involves the potential alteration of any semantic element necessary to achieve a specific outcome, such as label-flipping. To achieve this, we could theoretically modify as many words as needed. However, to maintain minimal edits, we impose a cap on the number of word substitutions allowed per textual input and employ beam search to identify the most suitable changes. This strategy inevitably leads to a compromise on absolute controllability, as it does not ensure that every possible word substitution is made. Yet, our framework guarantees that at least one word in each input will be modified, thus upholding a basic level of controllability. In our experiments (refer to Table 6.1), we accept this compromise, prioritizing minimality over extensive controllability. While it is feasible to achieve full controllability by adjusting the constraints previously mentioned (i.e., maximum number of substitutions and beam search utilization), such modifications typically degrade performance in terms of minimality.

Optimality vs. Execution Speed In our framework, we adopt both a deterministic method (refer to *Deterministic w. fluency* in Table 6.1) and a GNN-based strategy (refer to *GNN w. fluency* in Table 6.1) to address the RLAP. The deterministic approach guarantees optimality, as established graph matching algorithms are known to secure the best solution (cite in [162, 142]). However, these algorithms, with a computational complexity of $O(mn \log n)$, tend to slow down as the graph, and consequently the dataset size, increases—this size correlates with the number of words that need substitution. By substituting these deterministic algorithms with a trained GNN (detailed in Section 6.3.2), we achieve a substantial increase in processing speed at the expense of achieving only an approximate solution, rather than an optimal one.

Explainability vs. Execution Speed In our work, we utilize WordNet as the default way of computing edge weights between nodes, where each edge weight is based on the path that connects a source word s with target word t in WordNet. By mapping each concept to WordNet synsets, a deterministic concept position is assigned to each word, providing a fully transparent concept mapping to a well-crafted lexical structure. The utilization of word embeddings casts a shadow on word mapping, since we transit to a vector representation of an uninterpretable multi-dimensional space via black-box models. Similarity in the embedding space translates to semantic similarity of physical concepts, acting as our guarantee towards employing embedding models.

In combination with the deterministic solution to RLAP, WordNet mapping guarantees *explainability* of edits, since all paths $s \rightarrow t$ are tractable, and the choice of edges is fully transparent due to the deterministic selection process of graph matching algorithms [21]. By obtaining the resulting matching M we gain full access to the set of edits to perform $S \rightarrow T$ transition. A sacrifice in explainability is imposed when using the GNN instead of the deterministic graph assignment algorithms: the GNN introduces an uncertainty to the edge selection, since we cannot be entirely sure *why* a specific edge was chosen. Although we have trained the GNN to output the RLAP solution, the model itself still remains a black-box structure that hides the exact criteria which decide whether an edge will be selected or not. Still, in some applications the speedup offered by the GNN outweighs this drop in explainability, while the opposite may hold in cases where trustworthiness is of utmost importance.

Overall, as observed from our experiments (see Table 6.1), leveraging embedding models to compute edge weights and the GNN to solve RLAP showcases major improvements in *fluency*, *flip-rate* and *minimality*, while also being considerably faster. Someone could argue that this approach is clearly better that the fully deterministic one, since it produces higher quality edits. Despite that, we need to point out that these improvements come at a significant cost on explainability, since, due to the GNN, the edge selection process is no longer transparent and edge weight computation depends on black-box embedding models.

6.7 Methodological and Technical Details

6.7.1 GNN Training

Our approach to training the Graph Neural Network (GNN) embedded within our framework starts by building upon the pretrained model proposed in [205]. We then refine this model for our RLAP task, adhering closely to the procedure described by the original authors, apart from a slight modification in the loss function.

First, we construct a synthetic dataset of M samples[7], where each sample includes a cost matrix C whose elements are drawn from a uniform distribution in $(0, 1)$. We also derive the corresponding optimal assignment solution by applying the Hungarian algorithm [162]. Treating RLAP as a binary classification problem, we split the ground-truth assignment matrix Y^{gt}[8] into positive and negative labels. Since each node has at most one positive edge connected to it (with all other edges being negative), we employ the *Balanced Cross Entropy* loss to mitigate the influence of numerous negative labels:

$$\begin{aligned} L = -\sum_{i=1}^{n}\sum_{j=1}^{m}\Big(& w \times y_{ij}^{gt}\log(y_{ij}) + (1-w)\times \\ & (1-y_{ij}^{gt})\log(1-y_{ij})\Big), \end{aligned} \tag{6.7.1}$$

where y_{ij} is the predicted label for the edge linking source node i to target node j, y_{ij}^{gt} is the associated ground-truth value (positive or negative), and w is a balancing weight that offsets the dominance of negative labels during training. Additionally, n and m represent the cardinalities of the source and target node sets, respectively, implying $|S| = n$ and $|T| = m$.

Following [205], we train the GNN for a total of 20 epochs. The initial learning rate is set to 0.003 and is gradually reduced by 5% every 5 epochs.

6.7.2 Proof of Naive Graph Matching Complexity

We now demonstrate why the naive approach to solving adversarial s–t matchings has an exponential time complexity of $O\big(|T|^{|S|}\big)$. Consider the illustrative graph in Figure 6.7.1, which has a source set $S = \{A, B, C\}$ with cardinality $|S| = 3$, and a target set $T = \{1, 2, 3, 4\}$ with cardinality $|T| = 4$.

Let us observe the possible ways in which each source node can be connected to the target set:

- For source node A, there are $|T| = 4$ possible matches: $A-1$, $A-2$, $A-3$, or $A-4$.
- Independently, source node B also has 4 possible matches: $B-1$, $B-2$, $B-3$, or $B-4$.
- Similarly, source node C can be matched with any of the 4 targets: $C-1$, $C-2$, $C-3$, or $C-4$.

Since each source node selects among $|T|$ options without regard to the others, the total number of combinations for $|S| = 3$ becomes $4 \times 4 \times 4 = 4^3$. In the general case of $|S|$ source nodes matched to $|T|$ targets, this number scales to $|T|^{|S|}$, underscoring the exponential complexity $O\big(|T|^{|S|}\big)$ of the naive solution.

6.8 Conclusion

In this chapter, we present a framework developed based on the one introduced in Chapter 4. The essential concept of this framework relies on the use of a bipartite graph. Rather than identifying the closest instance in the explanation dataset, this approach utilizes bipartite graphs to generate a new instance. Specifically, it facilitates the generation of optimal and controllable word-level counterfactuals through graph-based substitutions. The evaluation of this framework was carried out on two classification tasks. A novel approach involving Graph Neural Networks (GNN) was introduced to augment the previously proposed baseline deterministic graph assignment algorithm, which resulted in a significant acceleration of the overall process.

[7] Each sample is a weighted bipartite graph.

[8] Y^{gt} is a matrix in which $y_{ij}^{gt} = 1$ if edge $i \to j$ appears in the minimum matching, and -1 otherwise.

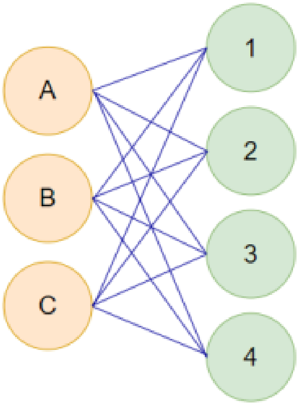


Figure 6.7.1: An example graph illustrating a bipartite structure with $S = \{A, B, C\}$ and $T = \{1, 2, 3, 4\}$.

Comparisons were made between the outcomes of this method and those achieved by two state-of-the-art (SoTA) editors. It was demonstrated that the proposed method not only surpasses these editors in most of the evaluated metrics but also does so with considerable swiftness. Additionally, some trade-offs that users must consider before implementing the proposed method were presented. Looking forward, it is contemplated that the integration of additional external lexical sources, such as ConceptNet, might be explored to broaden the array of potential substitution candidates. Furthermore, enhancements to the performance of the GNN model, which is employed to solve the Relaxed Linear Assignment Problem (RLAP), are also being considered in order to more closely approximate the deterministic optimal solutions.

Chapter 7

Evaluation of Counterfactual Explanations

7.1 Introduction

In the pursuit of more interpretable and accountable artificial intelligence, the generation and evaluation of counterfactual explanations have emerged as a pivotal area of research. Counterfactuals — hypothetical alternatives to real-world events or decisions — provide insights into machine learning models by illustrating how slight modifications to input data can lead to different predictions. This not only enhances transparency but also aids in debugging and improving model robustness.

Counterfactual editors or simply *editors*, aim to make minimal modifications to a given input in order to alter the prediction of a classifier. We present a classification of these systems, review related literature, and provide an overview of their evaluation methods.

The methodologies and intended use-cases of these editors vary [189]. For instance, systems like MiCE [302] and DoCoGen [31] are text counterfactual editors that are optimizing their edits based on the output of a specific predictor, $g()$, by pseudo-randomly masking words in the text and optimizing the proposed replacement to change g's output. Another approach, named CounterfactualGAN [296], combines a conditional GAN (Generative Adversarial Network) with embeddings from a pretrained BERT encoder [61] to model-agnostically generate realistic natural language text counterfactuals. In contrast, text editors like Polyjuice [375] aim to identify general text perturbations that can alter the semantics of a sentence without targeting a specific predictor. They refer to this as *general purpose counterfactuals*, which can be used for a variety of purposes, from data augmentation to generating counterfactual explanations or conditioning to a specific task/dataset.

A significant group of editors focus on generating *adversarial examples* to expose a classifier's vulnerabilities. These adversarial models may differ from other counterfactual editors as they do not necessarily aim to generate a minimal or fluent edit of the original input, and the edits might include noise addition, etc. A suite of adversarial example generators for NLP, including TextFooler [133] and Bert-Attack [133], is implemented in the TextAttack framework [251]. The simpler form of such methods involves using gradient descent on the instance to generate examples that alter the predictor's class while simultaneously optimizing one or more metrics [253]. Instead of attempting random permutations to generate counterexamples, other editors only alter the important features of each text. This importance is calculated in various ways, such as training a classifier to extract the correlation of each term with the task [367], measuring the effect of a feature deletion on the prediction of the classifier [133], or using the predictor's attention [302]. Then, the important terms can be replaced with synonyms, antonyms, important terms from other tasks, or using pre-trained seq2seq models [42, 229, 304, 302, 375, 79].

The article [120] proposes an editor that operates at the image level, generating adversarial examples through gradient shielding within a restricted area. Drawing inspiration from techniques used to identify crucial regions in object detection tasks, a system was developed for creating region-specific adversarial examples for image classification. This system utilizes a new technique called gradient mask, designed to produce adversarial examples that are highly effective in their attacks and cause minimal disruption. Meanwhile, other techniques like AdvProp [381] aims to improve robustness and reduce overfitting through an adversarial training approach, which incorporates adversarial examples as additional training data. Central to this approach is the use of a separate auxiliary batch normalization specifically for adversarial examples, acknowledging their distinct statistical distributions compared to standard examples. CoCoX [8], utilizes "fault-lines"—key semantic-level features critical to alternate predictions—to explain model outcomes. Specifically, for an input image I, where CNN model M predicts class c_{pred}, CoCoX identifies the minimal explainable concepts necessary to add or remove from I to change M's classification to a different target class c_{alt}.

To effectively measure the quality and impact of these counterfactual explanations, a variety of metrics have been developed. These metrics are designed to evaluate counterfactuals across multiple dimensions, such as their ability to achieve the desired outcome (validity), the minimalism of the changes they suggest (sparsity), and their closeness to plausible real-world alternatives (proximity). Each metric provides a different lens through which the effectiveness and utility of counterfactual methods can be assessed, making them crucial for researchers and practitioners alike. Moreover, specific considerations are required when dealing with natural language processing (NLP) tasks, where the nuances of human language demand specialized metrics like fluency, fidelity, and naturalness. These metrics ensure that the generated textual counterfactuals are not only effective in altering model decisions but also remain coherent, relevant, and realistic to human users.

The following section outlines the various general and NLP-specific metrics used to evaluate counterfactual methods. By understanding and applying these metrics, we can better gauge the strengths and limitations of different approaches to generating counterfactual explanations, ultimately leading to more interpretable and trustworthy AI systems.

7.2 Metrics for Counterfactual Explanations

7.2.1 Domain Agnostic Metrics

Flip Rate [304, 302, 43] is an essential metric in the evaluation of counterfactual methods, primarily used to measure the efficacy of modifications made to input data. Specifically, it quantifies the proportion of instances where an applied edit successfully shifts the outcome to a contrasting label, indicating a change in the decision or prediction made by a model. This metric is pivotal for assessing whether the interventions suggested by a counterfactual are meaningful and effective in altering outcomes. A higher flip rate suggests that the counterfactuals generated are not only pertinent but also potent enough to influence the model's behavior significantly. This makes the flip rate an invaluable metric for researchers and developers who aim to enhance model transparency and understand the decision boundaries of their predictive algorithms. By focusing on the flip rate, one can gauge the practical impact of counterfactual explanations in real-world applications, ensuring that these hypothetical alternatives fulfill their intended purpose of illustrating potential decision changes. Flip rate is divided into two distinct sub-metrics:

- **Label Flip Rate** (LFP) calculates the percentage of new examples that flip the original label to the target label.
- **Soft Label Flip Rate** (SLFR) calculates the percentage of new examples whose label differs from the original example's label. SLFR measures how often LLMs generate valid counterfactuals independent of whether the new label is right.

For a dataset with K examples, we calculate FLR and SFLR as follows:

$$LFP = \frac{1}{K} \sum_{k=1}^{K} \nVdash (\tilde{l_k} = l'_k) \tag{7.2.1}$$

$$SLFR = \frac{1}{K} \sum_{k=1}^{K} \left| \tilde{l_k} \neq l \right| \tag{7.2.2}$$

where $\tilde{l_k}$ is the annotated label, l'_k is the target label, and l_k is the original label.

Minimality or Proximity [302, 375, 84, 150, 354] evaluates the closeness between the original input and its counterfactual counterpart. It measures the distance—often in terms of feature space or some domain-specific metric—to assess how minimal the changes are that lead to a different outcome. The underlying premise of using proximity as a metric is to ensure that the suggested modifications are subtle yet effective, promoting counterfactuals that are not only plausible but also closely aligned with the original data point. This minimal divergence is vital as it enhances the likelihood of the counterfactual being perceived as realistic and actionable. Moreover, proximity is especially valuable in scenarios where the goal is to provide users with practical and achievable steps for altering outcomes, such as in loan approval or medical diagnosis scenarios, where slight and realistic changes are preferable. By prioritizing proximity, developers can create more grounded and accessible counterfactual explanations, facilitating better understanding and easier implementation of suggested changes by end-users.

Sparsity [150] measures the minimalism of the changes suggested by a counterfactual. This metric assesses how few features are altered in the transition from the original instance to its counterfactual counterpart, emphasizing the importance of simplicity and clarity in making these hypothetical alterations. The rationale behind valuing sparsity lies in its direct correlation with the comprehensibility and psychological acceptability of counterfactuals to human users. Sparse modifications are easier for individuals to understand and implement, thereby increasing the practical utility of counterfactual explanations. In scenarios where decision-making processes need to be transparent and actionable, sparsity ensures that the explanations provided are accessible and feasible for users to act upon. As such, sparsity not only enhances the user-friendliness of counterfactual explanations but also supports their effectiveness in providing clear, concise, and impactful insights into AI-driven decisions.

Coverage [353, 150] measures the proportion of "good" counterfactuals generated for a given dataset. A "good" counterfactual, in this context, is typically defined by specific criteria such as a limited number of feature changes—often no more than three—ensuring that the suggested modifications remain practical and manageable. This metric provides a global estimate of a method's adequacy by quantifying how many of the generated counterfactuals meet a predefined standard of quality across different test sets. High coverage indicates that a method consistently produces counterfactuals that are likely to be useful and relevant in real-world scenarios, thus enhancing the method's reliability and applicability. Coverage not only highlights the effectiveness of a counterfactual generation method but also its ability to produce actionable and understandable alternatives, which are crucial for end-users who need to make informed decisions based on the model's outputs.

Relative Distance [150] evaluates the quality of counterfactual explanations by comparing the mean distance between generated counterfactuals and the original test instances to the mean distance of naturally occurring counterfactuals within a dataset, often termed as "native counterfactuals." This metric is particularly insightful as it provides a benchmark for how closely the machine-generated counterfactuals mimic real-world scenarios where slight variations lead to different outcomes. By measuring this relative distance, researchers can assess the psychological validity of the counterfactuals: the closer the generated counterfactuals are to the native ones, the more likely they are to be perceived as plausible and understandable by humans. This metric thus serves not only to gauge the realism and relevance of the counterfactuals but also to ensure that they provide intuitive and actionable insights that can effectively aid users in interpreting and trusting machine learning decisions.

Diversity assesses the breadth and variety of alternatives generated by a model. This metric addresses the need for counterfactual explanations to provide a range of different outcomes from a single input, offering users multiple pathways or options for achieving a desired result. In practice, diversity is often quantified using measures such as self-BLEU scores [425] for text inputs or other similarity indices to evaluate how distinct each generated counterfactual is from others within the same set. A high diversity score indicates that the model can produce varied counterfactuals, which is crucial for avoiding bias and ensuring robustness in decision-making processes. Lexical diversity, apart from SelfBleu, can be assessed through the Distinct-n metric [179], which quantifies the diversity of generated CFEs by calculating the ratio of unique n-grams to the total number of n-grams. When it comes to semantic diversity, the $\text{dist}(\cdot)$ function can utilize various measures such as SBERT embedding similarity [293], BERTScore [406], or semantic uncertainty [163].

By fostering a diverse set of plausible counterfactuals, developers and users can explore a wider landscape of potential changes, thereby gaining deeper insights into the model's behavior and increasing the likelihood of identifying genuinely actionable and effective interventions.

Actionability [277, 237] evaluates whether the changes recommended by a counterfactual can be feasibly implemented in a real-world context, ensuring that the proposed alterations are within the realm of possibility for the end user. Actionability is paramount because it directly impacts the usefulness of counterfactuals in operational settings, such as policy-making, clinical decisions, or customer service enhancements. A counterfactual that suggests realistic and achievable changes is more likely to be used to make informed decisions and drive effective interventions. This metric is particularly significant in scenarios where the cost, ethical considerations, and practical constraints of implementing changes are critical. Hence, assessing actionability helps ensure that counterfactual explanations do more than just fulfill theoretical criteria; they provide genuine, executable insights that can lead to tangible improvements and informed decision-making processes.

Causal Constraint Satisfaction (Feasibility) [322, 237] assesses whether the modifications suggested by a counterfactual respect the underlying causal relationships inherent in the dataset. In essence, it measures how feasible the suggested changes are within the context of what is realistically possible, ensuring that the counterfactuals do not merely represent abstract mathematical solutions but are actionable and plausible changes that could occur in the real world. For instance, in scenarios involving sequential or dependent data, such as time-series forecasts or patient treatment records, it is vital that the counterfactuals adhere to logical sequences or medically feasible interventions. By prioritizing causal constraint satisfaction, developers and researchers can generate more meaningful and applicable counterfactuals that enhance user trust and compliance, particularly in critical domains such as healthcare, finance, and policy making.

Complexity [354, 353] quantifies the time or the computational complexity that a counterfactual editor need for producing an outcome. In practical scenarios, where timely decision-making is crucial, the speed at which counterfactuals are generated can significantly impact their utility. For instance, in dynamic environments such as real-time trading or emergency response systems, a slower generation time may render counterfactual explanations less useful. This metric is particularly valuable for comparing different methods or algorithms, identifying those that not only provide high-quality and effective counterfactuals but do so in an expedient manner. Optimizing for counterfactual generation time without compromising the quality of the explanations ensures that these tools are not only theoretically valuable but also practically applicable in fast-paced or resource-constrained settings.

7.2.2 Assessing Counterfactual Explanations in NLP

Minimality [302, 81] measures the "size" of an edit on textual data by quantifying the word-level Levenshtein distance between the original and edited input. The Levenshtein distance calculates the minimum number of deletions, insertions, or substitutions required to transform one text into another, making it an ideal measure of how minimal an intervention is. The principle behind Minimality is that smaller, more subtle edits are preferable, as they are less likely to distort the original meaning or intent of the text while still achieving the desired change in output. This is particularly important in NLP applications where maintaining the coherence and context of the original text is crucial. By emphasizing minimalism, natural language processing counterfactual methods can guarantee that modifications are both impactful and subtle, making it the predominant and most frequently employed metric that editors strive to minimize.

Fluency [349] measures how well the output text $f(x)$ aligns with the expected distribution of texts, $\mathcal{L}$. Ensuring that this text is fluent and within distribution poses a significant challenge, primarily because the true distribution $\mathcal{L}$ may be inaccessible, and assessing fluency systematically is often difficult. To approximate fluency, the token-level perplexity (PPL) of a large language model is commonly utilized [138, 356, 349]. This method involves using a model $\mathcal{M}_{\mathcal{D}}$ trained on an extensive dataset $\mathcal{D}$, calculating the average perplexity for a text sequence $x = x_1, x_2, ..., x_T$ as follows:

$$\mathrm{PPL}(x) = \exp\left\{-\frac{1}{T}\sum_{t=1}^{T}\log p_{\mathcal{M}_{\mathcal{D}}}(x_t|x_{1:t-1})\right\}. \tag{7.2.3}$$

Without fine-tuning $\mathcal{M}_{\mathcal{D}}$ on $\mathcal{L}$, this formula serves as a baseline for fluency measurement. However, if $\mathcal{L}$ is accessible, $\mathcal{M}_{\mathcal{D}}$ can be fine-tuned to adapt to this specific distribution, creating $\mathcal{M}_{\mathcal{L}}$ which can then more accurately detect out-of-distribution (OOD) cases using the same PPL formula. This allows for a comparison between the perplexity scores of $\mathcal{M}_{\mathcal{D}}$ and $\mathcal{M}_{\mathcal{L}}$, effectively highlighting any deviations from the normal distribution and decreases in fluency.

Content Preservation [33] assesses the quantity of the original content remains intact in the counterfactual output, ensuring that the essential meaning and information are preserved despite modifications made to achieve a different model outcome. Content preservation is typically measured using cosine similarity between the embeddings of the original and the counterfactual text, which are often derived embedding models like BERT [183]. High scores in content preservation indicate that the counterfactual maintains a strong semantic alignment with the original input, thereby supporting the usability and coherence of the generated explanations. This metric is particularly important in applications where the fidelity of the information is crucial, such as in legal or healthcare settings, where altering the meaning or omitting critical details could lead to misinterpretations or errors in decision-making based on the counterfactuals.

Fidelity [297] assesses how accurately the counterfactuals reflect the true behavior of the underlying black-box model being explained. In essence, fidelity measures the degree to which the modifications proposed by the counterfactuals would lead to the predicted changes if those modifications were actually implemented. This metric is vital for ensuring that the counterfactual explanations are not only theoretically sound but also practically reliable in predicting model responses to hypothetical inputs. High fidelity in generated counterfactuals boosts user trust in the explanations provided, as it confirms that the explanations are grounded in the operational logic of the model. Therefore, fidelity serves not just as a measure of accuracy, but also as a benchmark for the utility and reliability of counterfactual explanations in helping users understand and interact with AI systems more effectively.

Perceptibility [297] focuses on measuring the semantic similarity between the original text and its counterfactual version to assess how perceivable the changes are to end users. Perceptibility is quantified by employing advanced semantic models like the Universal Sentence Encoder (USE), which calculates the semantic distance between the two texts. A crucial aspect of this metric is ensuring that the modifications, while noticeable, remain subtle enough to maintain the integrity and contextual relevance of the original message. This balance is vital in applications such as sentiment analysis or content recommendation systems, where slight nuances in text can lead to significantly different outcomes. High perceptibility in counterfactuals ensures that the edits are understandable and meaningful, providing clear insights into how specific changes to the input affect the outputs of NLP models, thus aiding in better interpretation and trust in AI decision-making processes.

Expanding on the metrics used to evaluate counterfactual methods, classical text generation metrics also play an essential role. These metrics assess **grammatical correctness** and **coherence**, concentrating on the quality of the generated text with respect to its adherence to linguistic standards and overall readability. **Grammatical correctness** evaluates whether the text aligns with established rules of syntax and usage, ensuring the content is free from errors that could undermine its clarity and credibility. **Coherence** measures how logically the sentences connect and whether the overall text presents a cohesive and comprehensible narrative, thereby ensuring a smooth and seamless reading experience.

7.3 Inconsistency of Counterfactual Explanations

When delving into the realm of counterfactual explanations, it becomes evident that while existing metrics offer avenues for comparing and contrasting different methodologies, a glaring challenge persists: the absence of a definitive ground truth. This void complicates the assessment of a singular explainer's efficacy when considered in isolation, as it becomes challenging to ascertain how closely its output aligns with an ideal explanation that, in theory, could be achieved. In the pursuit of enhancing the evaluation process, an approach rooted in comparison emerges as a promising avenue. By scrutinizing the performance of a counterfactual system against its own outputs, we aim to shed light on its effectiveness in generating explanations that hold merit within the context of its own workings.

Drawing inspiration from the concept of back-translation, which has demonstrated its utility in evaluating and refining machine translation systems, we propose a methodological framework that builds upon iterative feedback loops. This framework involves feeding the system's output back into itself—a counterfactual of a counterfactual. Through this iterative process, we anticipate that the resulting explanation should, at the very least, match the quality of the original input. This expectation is grounded in the understanding that the original input, being both existent and actionable, serves as a tangible benchmark in other works a "lower bound" against which the generated edit can be measured. In essence, the original input acts as a proxy for ground truth, providing a reference point for assessing the efficacy of the generated explanation.

This approach can be employed to establish a baseline for various metrics; our focus is its application to the concept of *minimality*, which stands out as a primary criterion that many editors strive to minimize [106]. Among the desirable attributes of counterfactual explanations is their ability to effect minimal changes to the input sample, with minimality serving as the metric to gauge the disparity between the original and edited samples.

The absence of an ideal standard for explanation complicates the determination of an optimal value for minimality—whether a specific value is deemed advantageous or disadvantageous remains uncertain. In scenarios where a ground truth explanation exists, calculating the optimal minimality becomes feasible; however, in its absence, the comparison of minimality values across various edits and editors emerges as the sole viable option. In essence, while striving to achieve minimal changes is a fundamental objective of counterfactual explanations, without a definitive benchmark, the evaluation of minimality becomes inherently comparative rather than absolute.

In this manner, a new methodology is introduced which utilizes a novel metric termed *inconsistency*. This metric employs the editor's past outputs as benchmarks to gauge the editor's ability to make minimal edits. The procedure involves re-entering the editor's output back into the system to create a subsequent edit. The expectation is that this subsequent edit should be at least as good as its predecessor. For instance, in Figure 7.3.1, which depicts the stages of the feedback loop approach, when the initial edit ("This movie was awful!") is fed back into the counterfactual system, the anticipated result is that the generated edit should at least equal the original text ("This movie was fantastic!"). However, it is observed that the editor introduces an unnecessary whitespace in the produced edit (as illustrated in 2:second edit in Figure 7.3.1). This suggests a superior output that the system failed to recognize, thereby confirming that the system did not produce the optimal output. It is critical to acknowledge that a counterfactual system with a non-zero inconsistency value is certainly sub-optimal. Yet, a zero inconsistency value does not necessarily mean that the system is optimal. This approach sets a lower limit for the editor, though establishing an upper limit might be challenging, if not impossible, to automate. The remainder of the paper provides an extensive explanation of the proposed methodology and innovative metric and demonstrates its application on several commonly used editors with various characteristics.

Back-translation for analyzing editors We formalise our problem as follows. We assume access to a classifier g such that $g : \mathcal{L} \to [0,1]^C$, where $\mathcal{L}$ the set of text for a specific language and C is the number of different classes. We then consider the counterfactual editors for g as functions $f : \mathcal{L} \to \mathcal{L}$, and we assume that the goal of the editor f is threefold:

1. The edited text is classified to a different class $\arg\max g(f(x)) \neq \arg\max g(x)$.
2. The edits are minimal with respect to some distance metric d: $f = \arg\min_{h \in \mathcal{F}} d(x, h(x))$, where $\mathcal{F}$ is the set of functions for which $\arg\max g(f(x)) \neq \arg\max g(x)$.
3. The edited text $f(x)$ is fluent and within the distribution of $\mathcal{L}$.

To assess the extent to which specific criteria are adhered to, the analysis focuses on the behavior of editors under conditions of iterative feedback. This involves studying the function $f(f(\ldots f(x)))$ over n iterations, with the aim of evaluating the three criteria after multiple applications of the editor. Initially, a novel

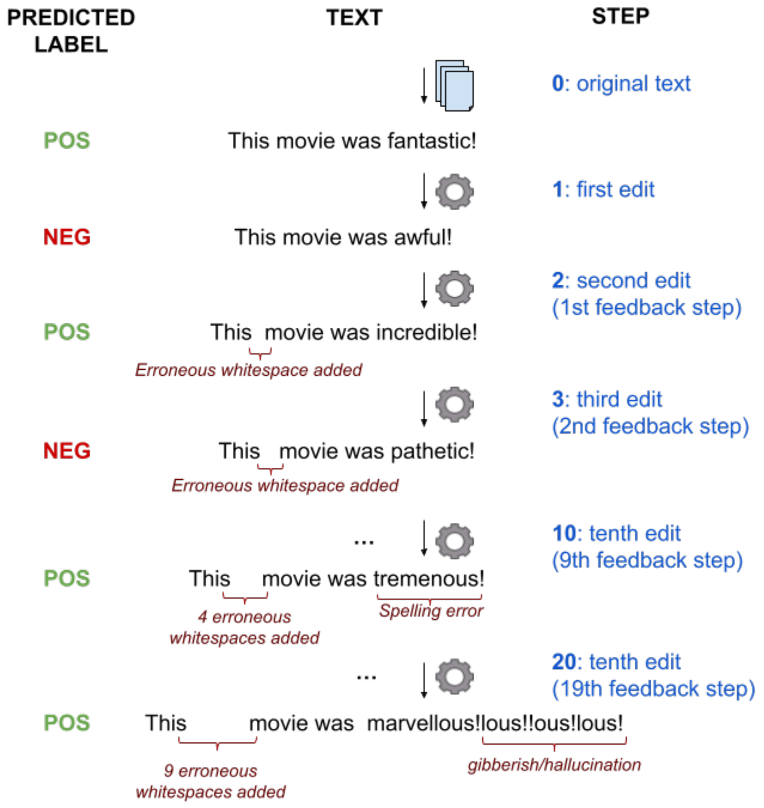

PREDICTED LABEL
TEXT
STEP
0: original text
POS
This movie was fantastic!
1: first edit
NEG
This movie was awful!
2: second edit
(1st feedback step)
POS
This movie was incredible!
Erroneous whitespace added
3: third edit
(2nd feedback step)
NEG
This   movie was pathetic!
Erroneous whitespace added
...
10: tenth edit
(9th feedback step)
POS
This   movie was tremenous!
4 erroneous whitespaces added
Spelling error
...
20: tenth edit
(19th feedback step)
POS
This   movie was marvellous!lous!!ous!lous!
9 erroneous whitespaces added
gibberish/hallucination

Figure 7.3.1: Using the back-translation framework to feed back the edited text to MiCE: We see the evolution of edits (centre) and predicted labels (left) through multiple feedback steps (right). As feedback steps increase, we observe an amplification of erroneous edits.

evaluation metric is defined to quantify the second criterion using the iterative feedback approach. Subsequent discussions elaborate on how the first and third criteria are thoroughly verified by measuring performance metrics at the n-th feedback step, denoted as $\mathsf{metric}@n$. The performance across various metrics at $@n$, indicating the metric value after n applications of the editor f, is examined. The subsequent sections detail the metrics used to evaluate each criterion and outline the underlying assumptions.

7.3.1 Inconsistency of Minimality

Intuitively, since the edits are ideally minimal, if a sentence A is edited into sentence B and their distance is $d(A,B)$, then feeding back sentence B to the editor should yield a sentence C for which $d(B,C) \leq d(A,B)$, otherwise C is not the result of a minimal edit [80]. This inequality holds based on that (a) we know that A exists, (b) we assume all textual edits to be reversible, hence A is reachable from B and (c) d is symmetric, meaning $d(A,B) = d(B,A)$. Thus, in this case, A can be used as a proxy to a ground truth, to be compared with C. Given a distance metric d (such as Levenshtein distance, embedding cosine similarity, etc.), we can measure how consistent the counterfactual editor is w.r.t d by iteratively feeding back the edited text to the editor and measuring the change in the value of d. Specifically, given an editor $f : \mathcal{L} \rightarrow \mathcal{L}$, a text $x \in \mathcal{L}$ and a distance $d : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}^+$ we define the *inconsistency of f with respect to d, for x as:*.

$$\mathsf{inc}(f,x) = \mathsf{relu}[d(f(f(x)), f(x)) - d(f(x), x)] \tag{7.3.1}$$

The difference $d(f(f(x)), f(x)) - d(f(x), x)$ shows how much the distance d changes between consecutive applications of the editor f and the relu function allows to take into account only the increase of the distance

[80]. This is important, because a decrease in the distance, which would correspond to a negative difference, is not necessarily an indicator of a good set of edits. It could, for example, indicate that not enough changes were made, and there is no way to know if that is the case, or if a better, more minimal set of edits was found. Contrarily, when the value is positive, we have a *guarantee* that a better set of edits exists, namely, the one of the previous feedback step. Equation 7.3.1 counts the difference in d after a single feedback iteration through the editor, but as with other metrics in this work, we can keep feeding back the output of the editor to itself, and compute $\mathsf{inc}(f, f(x))$ to get more information about the editor's inconsistency. When we do this, we measure the average inconsistency after n steps of feedback as [80]:

$$\mathsf{inc@}n(f, x) = \frac{1}{n}\sum_{i=0}^{n-1} \mathsf{inc}(f_{i+1}(x), f_i(x)), \tag{7.3.2}$$

where $f_0(x) = x$ and $f_i(x) = f(f_{i-1}(x))$.

7.4 Experiments

The approach is evaluated on two distinct datasets with classifiers specifically trained for each. A binary classifier designed for sentiment analysis is used on the IMDb dataset [227], and a multi-class classifier for short documents is employed on the Newsgroups dataset [169]. The methodology is applied to three counterfactual editors, and metrics are used to generate and test edits on these classifiers, with edited texts fed back to the editor for $n = 10$ iterations. At each step, the edited text chosen is the one with the minimal minimality that changes the classifier's prediction, if such an output exists; otherwise, the text with the smallest alteration is selected. This process is repeated until the tenth iteration and the behavior of each editor is analyzed across these metrics.

Additionally, the impact of test-set size on the variation in results and the statistical significance of these findings is studied, with detailed results presented in Section 7.7. More specifically, it is determined that when the test set size exceeds 200 texts, results on both datasets converge and yield statistically significant differences. To reduce computational demands, 500 texts are randomly sampled from the IMDb dataset for experimental use. The entire Newsgroups dataset is used, given its smaller size.

Editors

Experiments were conducted using three different editors, each with unique characteristics. Detailed descriptions of these editors and their principal distinctions are provided below.

Polyjuice Polyjuice [375] operates as a general-purpose counterfactual generator, creating perturbations aligned with predefined control types. It utilizes a GPT-2 model, fine-tuned on various datasets containing paired sentences, such as the IMDb dataset. Unlike other systems, Polyjuice does not incorporate classifier predictions in generating counterfactual texts; instead, it emphasizes the diversity of edits, guided by a set of learned control codes including "negation" and "delete". In the experiments, all the control codes were used.

MiCE MiCE [302] represents a two-step method for generating counterfactual edits. Initially, it employs a T5 deep neural network to fill blanks in texts, fine-tuned to align closely with the dataset's distribution. Subsequently, the text is masked either randomly or based on the classifier's attention in a white box approach, and the fine-tuned model is tasked with filling these blanks. This process is designed to identify the minimal edits necessary to modify the classifier's prediction. In the experimental setup, MiCE was used in a white box configuration, leveraging the predictor's outputs for fine-tuning and selecting mask locations based on the classifier's attention, contrasting with Polyjuice, which also uses a deep neural network but operates in a black box manner.

TextFooler TextFooler [133] is designed to generate adversarial examples for black-box classifiers and diverges from the other editors by not relying on a deep neural network like GPT2 or T5 for constructing counterfactuals. Instead, it focuses on identifying key words influential to the predicted class and substitutes them in a deterministic manner. The significance of each word is assessed by evaluating the impact of its removal on the classifier's output. Replacement words are chosen from the closest matches in the embedding space, maintaining independence from the rest of the sentence and the classifier's logic. TextFooler restricts its edits to synonym replacements, ensuring that replacements preserve the original part-of-speech tag.

Both MiCE and Polyjuice employ deep neural networks (T5 and GPT2, respectively) to fill in the blanks of randomly selected masked tokens. These networks can generate subtle and imperceptible perturbations that may alter their outputs. Unlike TextFooler, there are no restrictions on the number or the part-of-speech tagging of the words they can insert. This absence of constraints likely contributes to the greater minimality of edits by these two editors. For instance, if a neural network opts to replace a verb with a noun, additional words might need to be inserted to ensure syntactic correctness. This flexibility allows for increased diversity in the results produced by these editors.

Metrics

The metrics that we use with our methodology are:

Minimality This metric measures the word-level Levenshtein distance between the original text and its edited version, providing a quantitative assessment of the minimal changes made.

Inconsistency (@n) We calculate the inconsistency of the word-level Levenshtein distance using a specific formula (referred to as equation 7.3.2).

Soft Flip Rate Defined as the ratio $\frac{n_{flipped}}{n_{all}}$, where n_{all} represents the total number of samples in the dataset, and $n_{flipped}$ is the count of samples where the prediction changes following the text editing process.

Entropy This measures the entropy of the output from the predictor when applied to an edited input, serving as an indicator of the predictor's confidence.

Perplexity (Base Model) We evaluate the language model perplexity of the base GPT-2 model, a widely-used general-domain language model, as detailed in equation 7.2.3.

Perplexity (Fine-Tuned Models) The language model perplexity of GPT-2 fine-tuned on specific datasets like IMDB and Newsgroups assesses the unpredictability of the edited text in relation to each dataset. These datasets are available at IMDB[1] and Newsgroups[2].

Grammatical errors To assess the number of grammatical errors, character-level grammatical mistakes are measured after each feedback step. Under the assumption that the texts are within distribution, it is expected that these errors should not fluctuate significantly after each application of the editor. In practice, the T5 grammar correction model[3] is employed to produce a corrected version of the text at every feedback stage. The character-level Levenshtein distance between each text and its corrected version is then measured and reported as the number of grammatical errors @n.

Moreover, these metrics are computed after n steps of feedback, with the exception of inc@n, which inherently incorporates these feedback steps.

Datasets

IMDb The IMDb dataset originally includes 50,000 movie reviews, evenly divided into positive and negative categories for binary classification purposes. For our research, we selected a random sample of 500 reviews to create a test set. In this test set, the average review consists of 204 tokens and 1,000 characters, with a variability (standard deviation) of 112 tokens and 562 characters. The composition of the reviews in terms of sentiment is nearly balanced, with 52% categorized as positive and 48% as negative.

[1] https://huggingface.co/lvwerra/gpt2-imdb
[2] https://huggingface.co/QianWeiTech/GPT2-News
[3] https://huggingface.co/vennify/t5-base-grammar-correction

For reviews classified as positive, the average length is 990 characters and 530 tokens, with a standard deviation of 204 and 108, respectively. Reviews with a negative sentiment show a similar pattern, with an average of 1,006 characters and 204 tokens, and a standard deviation of 589 characters and 115 tokens, respectively.

Newsgroups The original Newsgroups dataset contains 20,000 brief documents distributed evenly across 20 different newsgroup categories, which indicate the topic of the documents. For our experiments, we utilized the test-set partition, which includes 7,000 documents from the scikit-learn library [4], as the training set had already been used to fine-tune some models. The dataset has an average of 603 characters and 207 tokens per document, with standard deviations of 495 and 103, respectively.

The dataset encompasses 20 classes, listed as follows: comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, misc.forsale, talk.politics.misc, talk.politics.guns, talk.politics.mideast, talk.religion.misc, alt.atheism, and soc.religion.christian.

Experimental Details

In both experiments, the predictors utilized were those employed by MiCE, which necessitate white box access to the predictor. This was chosen to minimize intervention in the editors' code. These predictors, based on ROBERTA-LARGE, remained unchanged during the evaluation, maintaining an accuracy of 95.9% for IMDb and 85.3% for the Newsgroups as reported in the proposed paper.

The comparison involved three counterfactual editors—MiCE, Polyjuice, and TextFooler—using the same classifier. Changes were made to the text, which was then tested with the classifier and the modified text fed back to the editor ten times. At each feedback stage and for each text input, the editors generated several altered versions. The version with the lowest minimality altering the prediction (the counterfactual goal) was selected if available; otherwise, the version with the lowest minimality was chosen. This approach was adopted to address instances where an editor did not initially change the prediction but did so in subsequent iterations.

For MiCE, a pre-trained T5 model, supplied by the authors, was employed[5] [6]. This model underwent fine-tuning using the identical dataset that was utilized for the predictor. During the generation process, default settings for each dataset were maintained as provided by the authors on their page[7], from which the experimental code was also acquired. An integration of custom data into the code represented the sole modification, enabling the generation of counterfactuals at each step.

Polyjuice is utilized via this module[8]. Throughout the generation procedure, searches were conducted across various control codes—"resemantic", "restructure", "negation", "insert", "lexical", "shuffle", "quantifier", "delete" — with the aim of producing as many perturbations as possible for each instance. This was achieved by setting $num_perturbations = 1000$. In none of the experiments was such an abundance of results returned by Polyjuice.

TextFooler was utilized via the TextAttack module[9]. To ensure a fair comparison, the same parameters as those outlined in the authors' paper were selected. Constraints were applied to prevent the modification of stopwords and words that had already been modified. Furthermore, the threshold for considering two words as synonyms based on word embedding distance was set at 0.5, with enforced replacements based on part-of-speech tagging.

[4] https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html
[5] https://storage.googleapis.com/allennlp-public-models/mice-imdb-predictor.tar.gz
[6] https://storage.googleapis.com/allennlp-public-models/mice-newsgroups-editor.pth
[7] https://github.com/allenai/mice
[8] https://github.com/tongshuangwu/polyjuice
[9] https://textattack.readthedocs.io/en/latest/

	IMDb			Newsgroups		
	MiCE	Polyjuice	TextFooler	MiCE	Polyjuice	TextFooler
inc@1 ↓	0.86	6.21	**0.01**	11.11	0.99	**0.04**
inc@2 ↓	5.95	4.65	**0.33**	7.97	1.29	**0.55**
inc@3 ↓	4.65	3.98	**0.36**	7.89	1.35	**0.46**
inc@5 ↓	4.87	2.90	**0.47**	6.92	1.30	**0.49**
inc@9 ↓	4.73	2.22	**0.49**	6.11	1.21	**0.46**

Table 7.1: Inconsistency (inc@n) computed on the IMDb and Newsgroups datasets.

7.5 Interpreting the inc@n metric

In Table 7.1, we present the outcomes of applying the proposed inc@n metric, which offers an intuitive understanding. The inconsistency value represents the average distance from the editor's previous local minimum to the current state, essentially measuring the mean number of token modifications made by the editor beyond those necessary to generate a valid counterfactual. Various factors, such as the method of identifying key text segments, the generation process, or the criteria for optimal edits, can influence these inconsistencies.

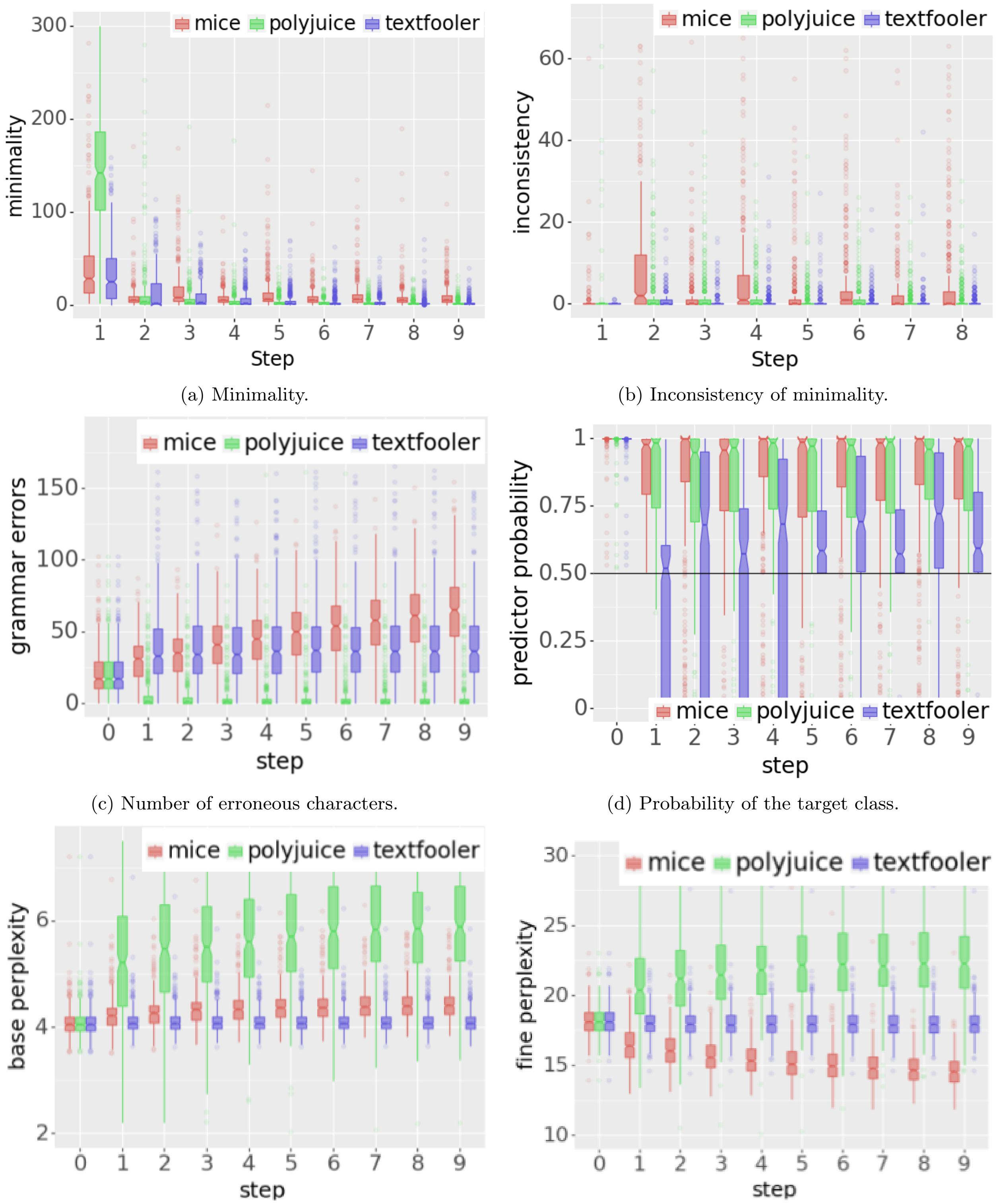
We observe notable variations in the inc@n metric among different editors, indicative of their diverse methodologies. TextFooler emerges as the most consistent, exhibiting low inc@n values that suggest minimal incremental changes. This consistency likely stems from TextFooler's systematic approach to selecting textual replacements, thereby facilitating the production of stable explanations. In contrast, MiCE and Polyjuice display less consistency, possibly due to their reliance on large language models for text generation. These models are prone to significant fluctuations from minor disturbances that might alter their outputs [133]. Specifically, Polyjuice often has to infer change points in the text without direct guidance from the predictor, leading it to make more drastic modifications to achieve desired outcomes.

Particularly in the initial stages for the IMDb dataset, Polyjuice's high inconsistency is marked by its need to predict change points blindly. Given the longer texts, the search space for Polyjuice becomes considerably larger, necessitating more aggressive edits. For instance, in the first two editing phases on the IMDb dataset, Polyjuice deleted over 70% of the original text in 83% of the cases where the changes did not shift the classification, thus were not adopted. This pattern of 'extreme erasure' diminishes in subsequent steps as the input length reduces significantly—original texts average 204 tokens, while those edited by Polyjuice drop to just 29.

Conversely, Polyjuice achieves greater consistency with shorter texts, where the reduced search space curbs the need for drastic changes. The variance between the initial and subsequent stages of the inc@n metric for Polyjuice highlights its strategy to shorten lengthy texts initially, which stabilizes its performance in later revisions. This behavior remains consistent across different datasets but is not as apparent in the first stages of results from the Newsgroups dataset, which features texts with 43% fewer tokens on average than those from IMDb.

To delve deeper into how each editor modifies the input text in terms of token count, we introduce Figure 7.5.2. This figure illustrates the average number of tokens in the edited texts compared to the input text's token count. The outputs from MiCE and Textfooler align closely with the input text across the datasets we analyzed. In contrast, Polyjuice often produces shorter texts, though there are instances where it generates longer texts, these are not typical. It is important to note that these variations may stem from Polyjuice's underlying mechanism, such as GPT-2, and the evaluation approach used. Specifically, since Polyjuice is designed for counterfactual generation, our evaluation prioritized texts that achieved a counterfactual objective (i.e., texts with a different label from the original) over texts that were similar to the original in classification [231]. This requirement, coupled with Polyjuice's task-agnostic design, compels it to implement more drastic edits, often cutting down a substantial portion of the original text consistently across the datasets.

In Figure 7.5.1b, it is observed that higher values of inc@n are associated with MiCE when n is even. This



(a) Minimality.

(b) Inconsistency of minimality.

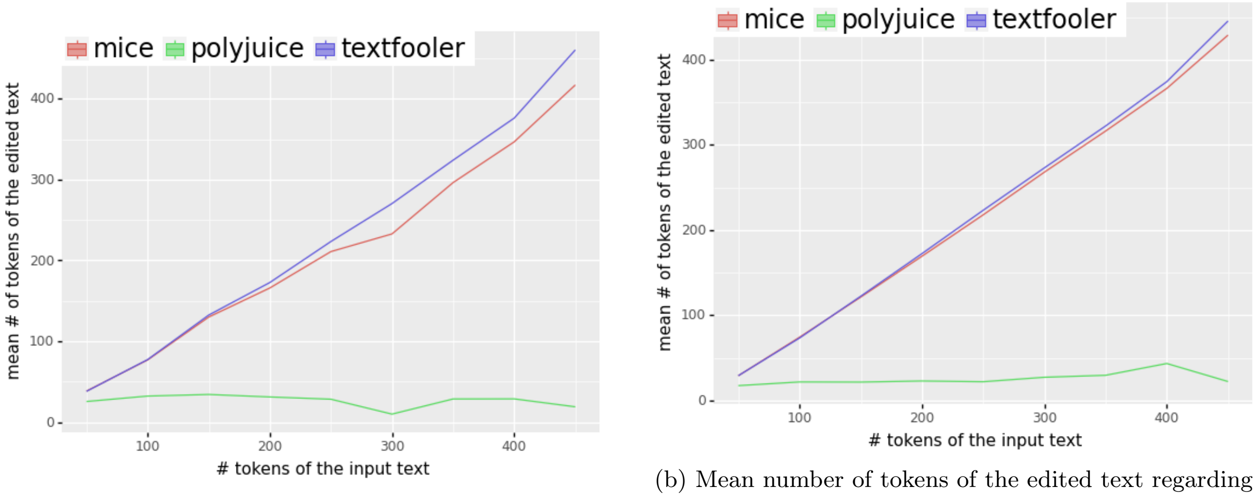
(c) Number of erroneous characters.

(d) Probability of the target class.

(e) PPL of base GPT-2.

(f) PPL of fine GPT-2.

Figure 7.5.1: Minimality, inc@n, and predictor probability, base-ppl and fine-ppl, after each step of feedback and for each editor on the IMDb dataset.



(a) Mean number of tokens of the edited text regarding the number of tokens of the input of the IMDb Dataset.

(b) Mean number of tokens of the edited text regarding the number of tokens of the input of the Newsgroups Dataset.

Figure 7.5.2: Mean number of tokens of the edited text regarding the number of tokens of the input.

phenomenon indicates an easier transition to the original class than from it, which could be due to the influence of residual elements from the original input text that drive the classifier towards maintaining its initial prediction, consequently necessitating fewer modifications. Higher inc@n values suggest that additional edits are required to progress to the subsequent feedback step compared to returning from the prior one. With MiCE achieving a perfect flip rate at the initial step and a 0.85 flip rate after nine feedback steps, texts from even feedback steps predominantly revert to the original class. This implies that transitioning back to the original class is facilitated by the residual content from the initial text, which steers the classifier towards the initial prediction and reduces the need for further edits. An illustration of this can be seen in Figure 7.5.3, where an edit by MiCE leaves intact certain elements of the original text that exhibit positive sentiment (highlighted in bold). In the same example, MiCE adjusts the word "carrier" to "masterpiece," likely a correction from a mistaken "career," which adds to the inconsistency observed. Particularly in the case of Newsgroups, MiCE's performance shows notable inconsistency in the initial steps, possibly stemming from its requirement to target a specific class, unlike other editors.

These observations underscore the necessity for feedforward evaluations of such systems, as the minimality@1 metric exposes only a restricted, dataset-dependent facet of the editors' capabilities and performance. Moreover, the effectiveness of incorporating additional feedback steps is demonstrated, allowing for a more accurate quantification of the differences between samples produced by an editor and obtaining a proxy for a global minimum. In Table 7.1, it is observed that beginning with *inc*@3, the inc@n values start to converge across both datasets.

It is noteworthy that the inconsistency of minimality reveals different facets of the editor compared to minimality alone. A high level of minimality indicates that more edits were made to change the label of the input text. This may be attributed either to limitations of the editor or to the inherent requirements of the input text needing extensive modifications to shift its classification. To rule out the latter possibility, it is essential to identify counterfactual examples demonstrating lower minimality, confirming the existence of more optimal states that were not explored. These states, however, must align with the exact conditions considered by the editor. The analysis of three editors in this study highlights a secondary aim to generate realistic

The biggest heroes, ***is one of the greatest movies ever****.*
A good story, great actors and a brilliant ending *is what*
makes this film the ~~jumping start~~ absolute worst of the
director Thomas Vinterberg's great ~~carrier~~ masterpiece.

Figure 7.5.3: MiCE example of an IMDb dataset sample.

	MiCE	Polyjuice	TextFooler
	IMDb		
Flip Rate@1 ↑	**1.000**	0.8747	0.6195
Flip Rate @9 ↑	0.8561	**0.9675**	0.7865
	Newsgroups		
Flip Rate@1 ↑	**0.87**	0.77	0.79
Flip Rate @9 ↑	0.836	**0.968**	0.89

Table 7.2: Flip-rate after feeding the original text to the editor once (@1), and after 9 steps of feedback (@9) for the IMDb and Newsgroups dataset.

counterexamples; thus, inserting random characters within the text, while possibly effective in achieving a label change with less minimality, does not align with the desired outcomes. Similarly, TextFooler strives to substitute each word with a synonym, making the replacement of a word with its antonym (for example, changing "love" to 'hate") an unacceptable strategy. Until now, there have been no effective or unbiased techniques to discover counterexamples with lower metrics such as minimality, that also satisfy the specific criteria set by the editor. The methodology introduced here addresses this deficiency, and the inconsistency metric allows for the quantification of the editor's limitations concerning the specified metric, in this instance, minimality. Briefly, a positive inconsistency indicates the presence of potential goal states with a lower metric value that were not investigated by the editor.

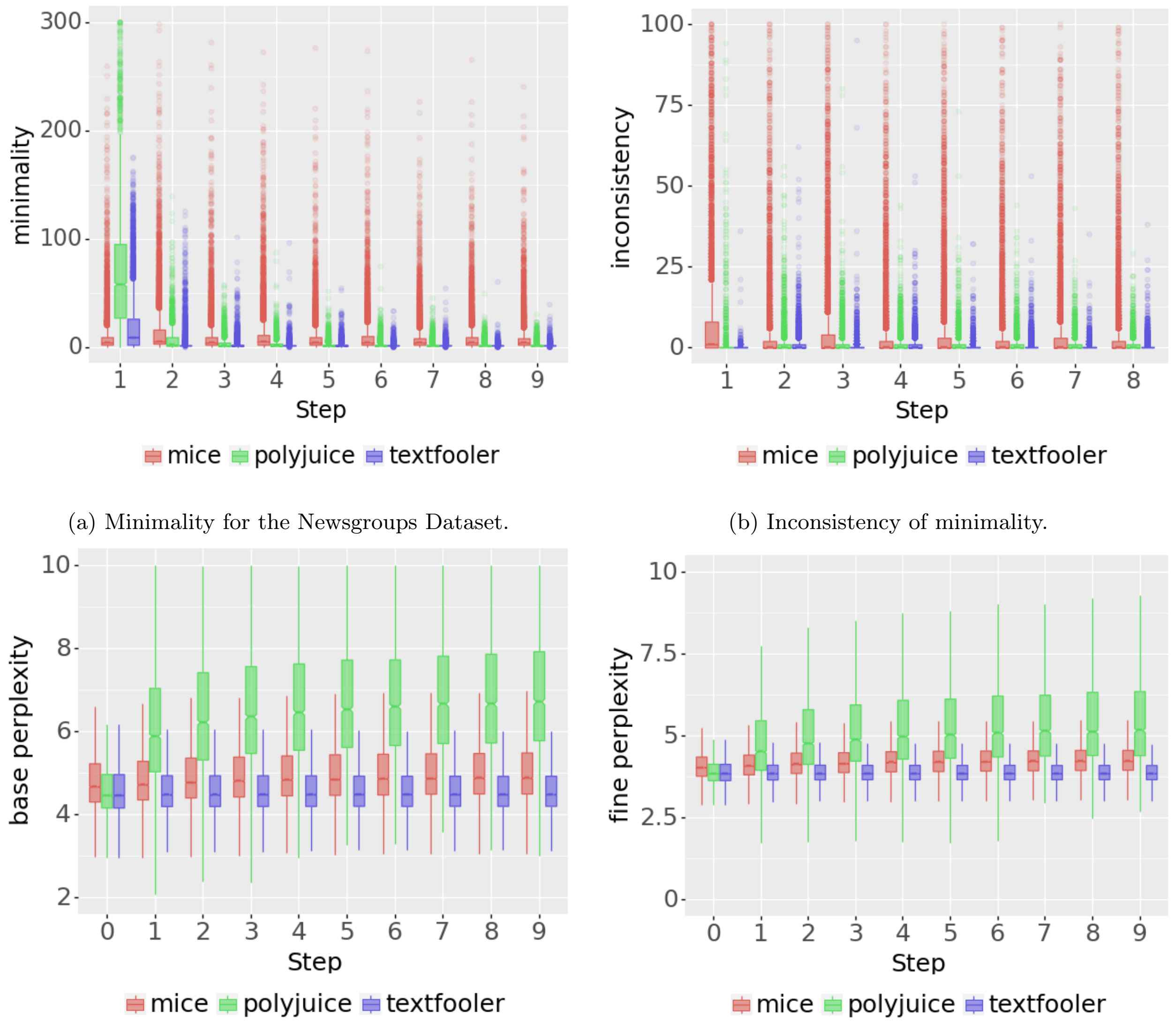
Counterfactuals of Counterfactuals in Multi-class Dataset - Newsgroups

Figure 7.5.4 is shown to depict the minimality, the inconsistency of minimality, the perplexity of base GPT-2, and the perplexity of fine-tuned GPT-2 for each editor using the Newsgroups dataset. Unlike the binary task observed in the IMDb dataset, no pattern is seen between odd and even steps, although a consistent behavior is noted. Since an editor is not required to revert to the original class but can switch to *any* other class at each feedback step, the challenge of label flipping remains similar across both even and odd steps. Isolation of instances where an editor reverts to the original class demonstrates that the patterns noted in the IMDb analysis persist. Moreover, the multi-class nature of the Newsgroups dataset appears to pose greater challenges for MiCE compared to other editors. This is attributed to MiCE's requirement for a specific target class for each edit, aimed at changing the class of the text, whereas editors like Polyjuice and TextFooler provide flexibility to alter the class to any different class. A target class is defined for each step as the second class predicted, adhering to the default approach used by the editor's creators in their research. Consequently, the task performed by MiCE is more demanding than that performed by other editors, as editing a text to shift from class A to class B proves at least as challenging as altering it from class A to any other class. This could explain the higher inconsistency values and differing behaviors observed with MiCE compared to those on the IMDb dataset. The figures suggest that the proposed method delivers consistent results concerning the behavior of each editor, even with fewer steps involved. This finding implies that the computational costs of the method can be significantly reduced, with as few as two or three steps sufficing to draw reliable conclusions.

7.6 Additional Insights from Counterfactuals of Counterfactuals

Besides measuring minimality and inc@n, we also investigated how the feedback approach can give us additional insights for the other desiderata for editors, flip-rate, grammatical errors and fluency.

Soft Flip Rate In Table 7.2, we present the flip-rate obtained by implementing the feedback methodology. Initially, MiCE demonstrates a perfect flip rate; viewing this in isolation could mistakenly suggest that the model consistently changes the classification of any given text. However, this result is specific to the test set and is not universally applicable, as there is a notable decrease in the flip rate in subsequent steps. This indicates that there are scenarios, closer to its distribution as discussed in Section 7.6, where MiCE fails to change the predicted class. On the other hand, the flip-rate for both Polyjuice and TextFooler shows an increase during later feedback stages.



(a) Minimality for the Newsgroups Dataset.

(b) Inconsistency of minimality.

(c) Perplexity of base GPT-2.

(d) Perplexity of fine GPT-2.

Figure 7.5.4: Minimality, Inconsistency of minimality, Perplexity of base GPT-2, and Perplexity of fine GPT-2 for the Newsgroups Dataset.

To investigate this, we examine the predictive probabilities of the target class following the application of various editors and subsequent feedback steps, as illustrated in Figure 7.5.1d. Here, a sample is considered to have 'flipped' if its target prediction probability exceeds 0.5. Specifically, when utilizing MiCE, the effectiveness of our proposed feedback method is evident. Initially, applying the editor once results in a 100% flip rate for this limited evaluation dataset, suggesting that the editor is capable of changing every text's predicted outcome. However, further analysis of subsequent steps reveals that this is not always the case. Reintroducing the edited text into MiCE shows a decrease in the number of texts being flipped, indicating the diminishing impact of the editor over multiple iterations.

Conversely, tools like TextFooler and Polyjuice demonstrate the lowest initial flip rates, yet exhibit improvement with ongoing feedback. By comparing these findings to those in the corresponding figure addressing the inconsistency of minimality (Figure 7.5.1b), similar trends are noticeable. Notably, there are discernible differences in the outcomes between even and odd feedback steps in the case of MiCE. This pattern reinforces the challenges editors face in consistently reverting to the original class after generating counterfactuals, highlighting the complexities involved in text editing for predictive modeling.

Grammatical Errors In fig. 7.5.1c we show the number of erroneous characters detected after n steps of feedback for each editor (step 0 refers to the original sentence from the dataset). We observe that some texts have fewer grammatical errors after the first pass of MiCE or Polyjuice (step 1 compared to step 0). This is because in some cases, when the input text has grammatical errors, these editors will correct them, which is not necessarily the desired behaviour of a counterfactual editor, especially since there seems to be no consistent impact of error correction on the semantics of the text nor the prediction of the classifier. We manually inspected these cases and found that these grammatical error corrections do not flip the prediction nor do they significantly affect the output value of the predictor, hence the claim that error-inducing edits are frequently undesirable, is highly relevant our case. This behaviour, however, does not hold for successive steps of feedback, since after the first edits the grammatical errors consistently increase for MiCE and TextFooler. Using our evaluation methodology we are able to detect such spurious behaviours of editors on text that deviates from the original as the iterative feedback steps amplify editing patterns, including the consistent introduction of grammatical errors (see also fig. 7.3.1). We note that this error amplifying behaviour of both editors seems consistent regardless of the generation strategy. On the other hand, Polyjuice rarely introduces grammatical errors, and seems to be able to handle grammatically incorrect inputs and correct them in the edits.

Fluency We employ two measures to gauge the fluency of the generated texts, as displayed in Table 7.3. The ppl-base metric identifies TextFooler as exhibiting the most fluent output, a value which remains consistent through multiple feedback iterations, underscoring the editor's reliability. Conversely, MiCE shows a slight decline in fluency following feedback, paralleling an increase in grammatical mistakes and inconsistency in comparison to TextFooler. Although Polyjuice registers the fewest grammatical errors, this does not translate into greater fluency in the ppl-base metric, suggesting that the edited text may deviate from the anticipated norm, as further evidenced by the highest ppl-imdb score recorded for Polyjuice.

Additionally, in Figures 7.5.1c, 7.5.1e, and 7.5.1f, we illustrate the progression of these fluency metrics through each stage of feedback. TextFooler maintains consistent fluency across both indicators, reflecting its minimal variability. Meanwhile, despite Polyjuice seldom introducing grammatical errors, it shows a declining fluency trend, with both the base-PPL and fine-PPL metrics worsening over time. A notable discrepancy in the perplexity trends is observed between MiCE and Polyjuice. For the base model, both editors exhibit a steady increase in perplexity; however, for the fine-tuned model, while MiCE's perplexity decreases, Polyjuice's continues to rise. This indicates that both editors negatively affect fluency, yet MiCE, which is specifically trained on IMDb data, aligns more closely with the IMDb style, suggesting a potential overfitting. In contrast, Polyjuice benefits from training on a variety of datasets, resulting in a broader range of editorial modifications.

Entropy In Figure 7.6.1, we display the entropy values from the output of the IMDb predictor, which facilitates easier comprehension due to its binary classification task, after each feedback step. Here, lower entropy values signify greater confidence in the prediction. We observe a recurring pattern in the odd and even feedback steps for both MiCE and TextFooler. During the even feedback steps associated with the original class, the predictor exhibits higher confidence, which diminishes as the feedback process progresses.

	MiCE	Polyjuice	TextFooler
		IMDb	
ppl-base@1 ↓	4.2546	7.4525	**4.1178**
ppl-base@9 ↓	4.4512	7.3825	**4.1161**
ppl-imdb@1 ↓	**16.5315**	33.4798	18.0662
ppl-imdb@9 ↓	**14.6069**	27.8074	17.9917
		Newsgroups	
ppl-base@1 ↓	5.164	8.926	**4.801**
ppl-base@9 ↓	5.36	7.878	**4.776**
ppl-newsgroup@1 ↓	4.27	6.67	**3.99**
ppl-newsgroup@9 ↓	4.4	5.90	**3.98**

Table 7.3: Metrics for measuring fluency computed for three counterfactual editors, of the IMDb and Newsgroups datasets, after feeding the original text to the editor once (@1), and after 8 additional steps of feedback (@9)

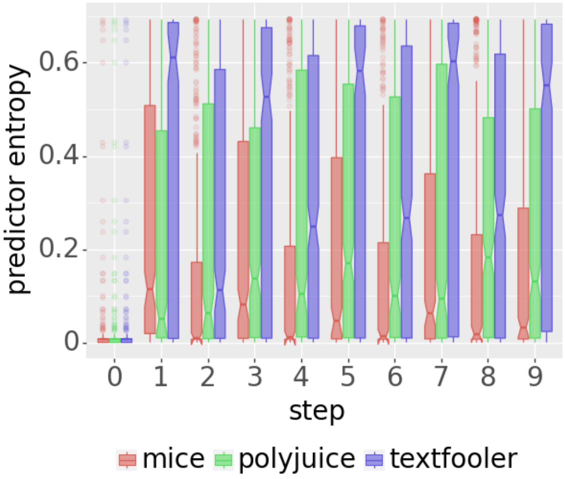


Figure 7.6.1: Entropy of the output of the IMDb predictor for each editor and after each step of feedback.

Conversely, in the odd feedback steps linked to the flipped class, the initial confidence is lower, but there is a gradual increase in confidence as the process continues. This pattern may be explained by the presence of elements from the original sentence that reinforce the original predictions during even steps. Moreover, the increasing confidence seen in even steps and the decreasing confidence in odd steps might be due to the feedback steps relating to the original class only if there has been a consistent flipping of the sample in all preceding feedback steps. As more steps accumulate, the likelihood of maintaining a direct connection to the original class diminishes unless the flipping of samples continues consistently. This trend is evident in MiCE, where the rate of flipping decreases with each step. However, further investigation is warranted, especially since TextFooler displays different behavior in terms of minimality between odd and even steps, though it shows similar patterns in terms of predictor entropy.

7.6.1 A focused Use Case

There is a broad range of use cases for counterfactual editors, that relate to different stakeholders, goals, and priorities, requiring different evaluation metrics to choose a suitable the editor. We demonstrate the potential of the proposed back-translation approach, elaborating the use-case of an AI engineer who uses a counterfactual editor to better understand the behaviour of a predictor.

Choosing an editor. The first step towards "explaining" a predictor for development purposes is to choose a counterfactual editor, based on performance for a set of metrics. They might prioritise *high flip-rate*, to ensure that the generated text is actually a counterfactual. For debugging purposes *low minimality* is also key, since it coincides with more understandable explanations and fewer edits are easier to investigate. The proposed feedback approach could help distinguish between editors that have similarly high performance for flip-rate and minimality using metrics @n. Additionally, the proposed *inc*@n metric could be used to find the most consistent editor regarding minimality.

Assuming access to the editors we analysed, without using the proposed methodology the engineer would probably choose MiCE, since without feedback it has the best flip-rate and low minimality, and there are no indications of biases or inconsistencies. On the other hand, if they used the proposed methodology and considered metrics @n, they would choose TextFooler, which is the most consistent while still maintaining sufficiently good values for metrics @1, and does not have the tendency to introduce grammatical errors, which could be misleading if they do not actually play a part in the prediction as we find in this paper (e.g., MiCE introduces extra whitespaces as shown in fig. 7.3.1, which we find to not affect the prediction flip) .

Inspection through the back-translation approach Having chosen an editor, the proposed feedback approach can also be used as an analysis tool to gain more knowledge about the underlying predictor it allows to get more predictions in the “neighborhood” of a sentence. Additionally, later feedback steps provide more understandable sets of edits (lower minimality; see also to figs. 7.3.1 and 7.5.3). Furthermore, it might be useful to observe samples that are closer to the decision boundary, i.e., high-entropy, low-confidence samples. We can see that TextFooler can generate such samples using the proposed approach, as predictor confidence tends to decrease after each step of feedback (see fig. 7.6.1).

7.7 Impact of Test Set Size

To assess how sample size influences the outcomes of our proposed metric, we carried out a series of t-tests across different sample sizes, feedback steps, and datasets. We specifically chose subsets containing 10, 50, 100, 200 and 500 samples to perform these tests. The goal was to compare the inconsistency scores between every pair of editors and determine at what sample size their scores significantly diverge.

The results, including the p-values, are presented in the tables: Table 7.4, Table 7.5 and Table 7.6 for the IMDb dataset and Table 7.7, Table 7.8 and Table 7.9 for the Newsgroups dataset. In these tables, any p-values below 0.05 across all feedback steps are highlighted in bold. Notably, for the IMDb dataset, p-values were consistently below 0.05 for sample sizes above 100. In contrast, for the Newsgroups dataset, this threshold of significance was observed with sample sizes exceeding 200.

These findings indicate that larger sample sizes tend to show statistically significant differences in the inconsistency scores, affirming the need for adequate sample sizes when evaluating this metric.

7.8 Conclusion

In this work, a methodology was introduced for analyzing various facets of counterfactual editors by acquiring an approximate ground truth through iterative feedback of their outputs. This approach, when combined with evaluation metrics from existing literature, enables a new understanding of the behaviors and performance of counterfactual editors tailored to their specific use cases, thereby assisting in the development of improved editing tools. The metric named inc@n was introduced to assess the consistency of these editors. It was demonstrated how this method could facilitate the diagnosis and analysis of a wide range of existing editors, revealing new insights into their behavior, particularly through the discrepancies observed between the odd and even feedback steps. Notably, this evaluation method illuminates the behavior of editors without requiring external input, such as human assessments or outputs from other editors.

The findings presented permit a more interpretable evaluation of editors, advancing beyond simple comparisons among them. These results encourage additional research in this field, encompassing experiments with new evaluation metrics, different editors, and various tasks. Moreover, plans are underway to enhance the understanding of counterfactuals by assessing their understandability and informativeness through human evaluation.

MiCE								
Sample Size: 10								
Step	1	2	3	4	5	6	7	8
Polyjuice	0.2812	0.588	0.3563	0.3219	0.1093	0.3376	0.3039	0.133
TextFooler	0.4788	0.4853	0.2538	0.2107	0.2014	0.6249	0.0695	0.1658
Sample Size: 50								
Polyjuice	0.0383	0.342	0.0266	0.0073	0.1714	0.0377	0.0852	0.1184
TextFooler	0.2805	1.232e-05	0.2646	0.004	0.03	0.0054	0.1028	0.0063
Sample Size: 100								
Polyjuice	**0.0252**	**0.0168**	**0.0001**	**0.0001**	**0.0048**	**0.0091**	**0.0081**	**0.0003**
TextFooler	**0.0495**	**6e-08**	**0.0104**	**0.0001**	**0.0016**	**0.0003**	**0.0032**	**0.0001**
Sample Size: 200								
Polyjuice	**0.0084**	**0.0036**	**1e-08**	**4.02e-05**	**2.72e-05**	**0.0012**	**0.0007**	**5.66e-06**
TextFooler	**0.0461**	**2.73e-14**	**0.0013**	**1.3e-10**	**0.0006**	**4.89e-07**	**2.2e-05**	**4.2e-08**
Sample Size: 500								
Polyjuice	**0.00043**	**0.0**	**0.036**	**0.0**	**0.0006**	**1e-08**	**0.001**	**0.0**
TextFooler	**0.0368**	**0.0**	**1.76e-06**	**0.0**	**1.86e-06**	**0.0**	**4.2e-05**	**0.0**

Table 7.4: P-value of the inconsistency of different sample sizes of the IMDb dataset for MiCE

Polyjuice								
Sample Size: 10								
Step	1	2	3	4	5	6	7	8
mice	0.2812	0.588	0.3563	0.3219	0.1093	0.3376	0.3039	0.133
textfooler	0.2831	0.3649	0.3056	0.2198	0.073	0.3337	0.1573	0.047
Sample Size: 50								
mice	0.0383	0.342	0.0266	0.0073	0.1714	0.0377	0.0852	0.1184
textfooler	0.0378	0.0015	0.021	0.0011	0.11	0.021	0.0199	0.0261
Sample Size: 100								
mice	**0.0252**	**0.0168**	**0.0001**	**0.0001**	**0.0048**	**0.0091**	**0.0081**	**0.0003**
textfooler	**0.0246**	**4.733e-05**	**4.351e-05**	**9e-06**	**0.0026**	**0.0038**	**0.0003**	**4.258e-05**
Sample Size: 200								
mice	**0.0084**	**0.0036**	**1e-08**	**4.028e-05**	**2.723e-05**	**0.0012**	**0.0007**	**5.66e-06**
textfooler	**0.0082**	**0.0**	**0.0**	**6e-08**	**1.367e-05**	**0.0002**	**1.6e-07**	**1.3e-07**
Sample Size: 500								
mice	**0.0004**	**0.0**	**0.036**	**0.0**	**0.0007**	**1e-08**	**0.0011**	**0.0**
textfooler	**2.2e-05**	**0.0001**	**3.88e-05**	**0.0016**	**0.0058**	**0.0009**	**0.0192**	**0.0007**

Table 7.5: P-value of the inconsistency of different sample sizes of the IMDb dataset for Polyjuice

TextFooler								
				Sample Size: 10				
Step	1	2	3	4	5	6	7	8
polyjuice	0.2831	0.3649	0.3056	0.2198	0.073	0.3337	0.1573	0.047
mice	0.4788	0.4853	0.2538	0.2107	0.2014	0.6249	0.0695	0.1658
				Sample Size: 50				
polyjuice	0.0378	0.0015	0.021	0.0011	0.11	0.021	0.0199	0.0261
mice	0.2805	1.232e-05	0.2646	0.004	0.03	0.0054	0.1028	0.0063
				Sample Size: 100				
polyjuice	**0.0246**	**4.733e-05**	**4.351e-05**	**9e-06**	**0.0026**	**0.0038**	**0.0003**	**4.258e-05**
mice	**0.0495**	**6e-08**	**0.0104**	**0.0001**	**0.0016**	**0.0003**	**0.0032**	**0.0001**
				Sample Size: 200				
polyjuice	**0.0082**	**0.0**	**0.0**	**6e-08**	**1.367e-05**	**0.0002**	**1.6e-07**	**1.3e-07**
mice	**0.0461**	**0.0**	**0.0013**	**0.0**	**0.0006**	**4.9e-07**	**2.225e-05**	**4e-08**
				Sample Size: 500				
polyjuice	**2.28e-05**	**0.0001**	**3.882e-05**	**0.0016**	**0.0058**	**0.0009**	**0.0192**	**0.0007**
mice	**0.0369**	**0.0**	**1.76e-06**	**0.0**	**1.86e-06**	**0.0**	**4.251e-05**	**0.0**

Table 7.6: P-value of the inconsistency of different sample sizes of the IMDb dataset for TextFooler

MiCE								
				Sample Size: 10				
Step	1	2	3	4	5	6	7	8
polyjuice	0.1311	0.0963	0.4378	0.0042	0.1069	0.2044	0.1384	0.1809
textfooler	0.0717	0.2283	0.0924	0.3264	0.165	0.2194	0.5411	0.0453
				Sample Size: 50				
polyjuice	0.1311	0.0963	0.4378	0.0042	0.1069	0.2044	0.1384	0.1809
textfooler	0.0006	0.0064	0.0338	0.1265	0.008	0.1015	0.0995	0.0101
				Sample Size: 100				
polyjuice	0.0033	2.64e-06	0.0177	1e-08	0.0017	2.878e-05	0.081	0.0049
textfooler	**1.29e-06**	**0.0004**	**0.0034**	**0.0001**	**0.0009**	**0.0104**	**0.0344**	**0.0007**
				Sample Size: 200				
polyjuice	**0.0**	**0.0005**	**4.937e-05**	**0.0002**	**2.13e-06**	**0.0003**	**0.006**	**0.0041**
textfooler	**1.513e-05**	**0.0**	**2.8e-07**	**0.0**	**3.5e-07**	**0.0**	**0.0043**	**2e-08**
				Sample Size: 500				
polyjuice	**0.0**	**2.32e-06**	**0.0**	**1.09e-06**	**0.0**	**5e-08**	**0.0**	**1e-08**
textfooler	**0.0**	**3.3e-07**	**0.0**	**3e-08**	**0.0**	**0.0**	**0.0**	**0.0**

Table 7.7: P-value of the inconsistency of different sample sizes of the Newsgroups dataset for MiCE

Polyjuice								
	Sample Size: 10							
	1	2	3	4	5	6	7	8
mice	0.3306	0.0995	0.1407	0.0043	0.7421	0.4028	0.1387	0.2846
textfooler	0.1311	0.0963	0.4378	0.0042	0.1069	0.2044	0.1384	0.1809
	Sample Size: 50							
mice	0.9654	0.0065	0.6376	8e-08	0.7342	0.3373	0.1692	0.9168
textfooler	0.0033	2.64e-06	0.0177	1e-08	0.0017	2.878e-05	0.081	0.0049
	Sample Size: 100							
mice	0.3659	1e-08	0.719	0.0	0.5959	0.173	0.1055	0.7947
textfooler	0.0004	0.0	0.0002	0.0	0.0001	1e-08	0.0358	4.421e-05
	Sample Size: 200							
mice	**0.0468**	**0.0**	**0.5025**	**0.0**	**0.8023**	**0.0176**	**0.0384**	**0.2746**
textfooler	**1.513e-05**	**0.0**	**2.8e-07**	**0.0**	**3.5e-07**	**0.0**	**0.0043**	**2e-08**
	Sample Size: 500							
mice	**0.0**	**2.32e-06**	**0.0**	**1.09e-06**	**0.0**	**5e-08**	**0.0**	**1e-08**
textfooler	**0.0005**	**0.026**	**5.3e-07**	**0.0643**	**1.8e-07**	**0.0037**	**0.0**	**0.0019**

Table 7.8: P-value of the inconsistency of different sample sizes of the Newsgroups dataset for Polyjuice

TextFooler								
	Sample Size: 10							
Step	1	2	3	4	5	6	7	8
polyjuice	0.1311	0.0963	0.4378	0.0042	0.1069	0.2044	0.1384	0.1809
mice	0.0717	0.2283	0.0924	0.3264	0.165	0.2194	0.5411	0.0453
	Sample Size: 50							
polyjuice	0.0033	2.64e-06	0.0177	1e-08	0.0017	2.878e-05	0.081	0.0049
mice	0.0006	0.0064	0.0338	0.1265	0.008	0.1015	0.0995	0.0101
	Sample Size: 100							
polyjuice	**0.0004**	**0.0**	**0.0002**	**0.0**	**0.0001**	**1e-08**	**0.0358**	**4.421e-05**
mice	**1.29e-06**	**0.0004**	**0.0034**	**0.0001**	**0.0009**	**0.0104**	**0.0344**	**0.0007**
	Sample Size: 200							
polyjuice	**1.513e-05**	**0.0**	**2.8e-07**	**0.0**	**3.5e-07**	**0.0**	**0.0043**	**2e-08**
mice	**0.0**	**0.0005**	**4.937e-05**	**0.0002**	**2.13e-06**	**0.0003**	**0.006**	**0.0041**
	Sample Size: 500							
polyjuice	**0.0005**	**0.0213**	**5.3e-07**	**0.0643**	**1.8e-07**	**0.0037**	**0.0**	**0.0019**
mice	**0.0**	**3.3e-07**	**0.0**	**3e-08**	**0.0**	**0.0**	**0.0**	**0.0**

Table 7.9: P-value of the inconsistency of different sample sizes of the Newsgroups dataset for TextFooler.

Furthermore, the scope of experimentation is set to be expanded, with intentions to employ feedback data to automatically rectify the weaknesses and inconsistencies found in editors during their fine-tuning phase, aiming for more robust and interpretable counterfactual edits. In line with these efforts, the exploration of incorporating feedback rationales into the training processes of counterfactual generation algorithms is planned. An objective inspired by back-translation might be considered to mitigate problematic behaviors and enhance performance.

Chapter 8

Explainable Metric for Story Visualization through Counterfactual Explanations

Despite the proliferation of generative architectures, the evaluation of generative models has remained an underrepresented field. Most recent models are assessed using outdated metrics that suffer from robustness issues and fail to evaluate critical aspects of visual quality, such as compositionality and logical coherence of synthesis. Simultaneously, the explainability of generative models remains limited, albeit important, with current approaches often requiring access to the internal mechanisms of these models.

In this chapter, generative models are treated as black boxes, and a novel framework is introduced by adapting the approach presented in Chapters 4 and 5. This adaptation shifts the focus from generating counterfactual explanations to evaluating and explaining generative systems [218]. The framework exploits knowledge-based counterfactual edits to identify which objects or attributes should be inserted, removed, or replaced in generated images to align them more closely with their intended conditioning. By focusing on concepts, more interpretable and meaningful evaluations of generative models are provided. Moreover, global explanations are produced by aggregating local edits, revealing the limitations of a model—specifically, the concepts it is inherently unable to generate. This insight is invaluable for understanding model biases and guiding future improvements.

The effectiveness of the proposed framework is demonstrated by applying it to various models designed for the challenging tasks of Story Visualization. The results validate the power of this concept-based evaluation in a model-agnostic setting, highlighting its potential to advance the field of generative modeling through more robust and explainable evaluation methods.

8.1 Introduction

The domain of image generation has emerged as a pivotal area in deep learning, catalyzing numerous state-of-the-art applications spanning from artistic creation to data augmentation in machine learning workflows [117, 300, 291, 290, 310, 23, 153]. Since the groundbreaking introduction of Generative Adversarial Networks (GANs) by Goodfellow et al. [99], research has predominantly aimed at refining the visual fidelity of generated images to closely match human perception.

Despite these advancements, evaluating the performance of generative models remains a significant challenge due to the absence of ground truth data for direct comparison. Traditional evaluation metrics, such as Inception Score (IS) [312], Fréchet Inception Distance (FID) [115], and Learned Perceptual Image Patch Similarity (LPIPS) [404], have been widely adopted to quantify image quality at the pixel level. However, these metrics have notable limitations, often failing to capture higher-level semantic discrepancies and being sensitive to minor perturbations that are imperceptible to humans. Concerns have been raised about their brittleness leading to inaccurate results [273].

Recent efforts like Clean-FID [273] attempt to address some of these issues by mitigating the impact of visual artifacts. Nonetheless, they still fall short when it comes to evaluating complex aspects such as image compositionality, logical consistency, and fairness in generation [25]. This limitation is particularly problematic in conditional image generation tasks, where the objective is to produce images that accurately reflect specific input conditions or attributes. Current attempts in conditional synthesis evaluation remain limited [325, 18], still facing the shortcomings of their unconditional counterparts upon which they are built.

Explainability in generative models is an emerging area that has not received as much attention as in discriminative models [1, 28]. In discriminative models, techniques such as saliency maps, SHAP values, and LIME have been instrumental in providing insights into model decisions. In contrast, explainability in generative models has been explored only in a limited capacity. Some studies have incorporated explainable feedback mechanisms within GANs [255], or have attempted to interpret the internal workings of these networks [92]. For instance, overfitting in GANs can be addressed by identifying the regions of an image that contribute to a discriminator's decision to classify a sample as real or fake, thus explaining the discriminator's decision [152]. This scarcity of literature hampers the development of robust explainable evaluation methods for generative models, especially when compared to fields like Natural Language Processing, where explainability has gained substantial traction [172, 264, 221].

To overcome these challenges, a paradigm shift from pixel-level evaluation to a concept-based approach is proposed. By focusing on semantic concepts—such as objects, attributes, and relationships—that are present or absent in the generated images, a more interpretable and meaningful assessment of generative models can be achieved. This conceptual framework facilitates the identification of specific capabilities and biases within a model, enabling targeted improvements and fostering transparency.

In this context, the first *explainable evaluation* technique for generative models is introduced, using the methodology presented throughout this thesis, specifically the modelology proposed in Chapters 4 and 5. Specifically, *counterfactual explanations* are leveraged to frame conditional generative evaluation as the answer to the following question: *What minimal changes are needed for a generated image to satisfy certain conditions or resemble a target concept?* Conceptual edits guided by external knowledge sources [**cece**] effectively chart the shortest path to incorporate the desired attributes into the generated image. Existing works that combine explainability with image generation operate on specific models [255, 92, 152] and demand access to their inner structure (white-box techniques). In contrast, the proposed approach treats the generative model as a black box, requiring only the generated outputs and their corresponding conditioning information.

To advance the field of generative modeling and address existing evaluation challenges, this work makes several contributions:

1. **Introduction of a Conceptual Evaluation Framework:** We present a *concept-based* evaluation framework[1] that departs from traditional pixel-level assessment methods. This framework is versatile and applicable to complex tasks such as Scene Generation (SG) and Story Visualization (SV), where capturing semantic content and relationships is crucial. By focusing on high-level concepts rather than low-level image features, our approach provides a more meaningful evaluation of generative models' capabilities.

2. **Development of Explainable Metrics:** Our proposed metrics are inherently *explainable*, designed to reveal which semantic concepts need to be added, removed, or altered in the generated images to align them more closely with the conditioning data or ground truth. Utilizing *counterfactual explanations*, we identify the minimal conceptual changes required, offering clear insights into how and why generated outputs deviate from expectations. These edit operations are conducted in a *model-agnostic* fashion, eliminating the need for access to the internal architecture or parameters of the generative models. This ensures that our evaluation method can be universally applied across different models.

3. **Identification of Generative Blind Spots:** Through comprehensive global explanations, our framework automatically uncovers potential *blind spots* in generative models—that is, specific concepts or elements that a model is intrinsically unable to generate due to limitations such as insufficient training

[1] Framework available at:

data, inherent biases, or architectural constraints. By highlighting these deficiencies, our approach provides valuable feedback for model refinement. This insight is crucial for improving model performance, addressing biases, and guiding future research efforts toward enhancing the generative capabilities of models across diverse concepts.

In essence, our contributions not only introduce a novel way of evaluating generative models but also enhance the transparency and interpretability of their outputs. By shifting the focus from pixels to concepts, we enable a deeper understanding of model behavior, paving the way for the development of more robust, fair, and reliable generative AI systems.

8.2 Related Work

Generative Adversarial Networks (GANs). Generative Adversarial Networks (GANs), introduced by Goodfellow et al. [99], have established themselves as a foundational architecture in the field of generative modeling. A GAN consists of two neural networks in competition: a generator $G(z; \theta_g)$ and a discriminator $D(x; \theta_d)$. The generator G maps a random noise vector z, drawn from a prior distribution $p_z(z)$, to the data space, aiming to produce outputs that resemble real data. The discriminator D assesses input samples x_i and outputs a probability $p_i = D(x_i)$ indicating the likelihood that x_i is a real sample from the data distribution rather than a synthetic one generated by G.

To enhance control over the data generation process, Conditional GANs (cGANs) were proposed [246]. In cGANs, both the generator and discriminator receive an additional input: a conditioning variable y. This allows the generator to produce data conditioned on specific attributes or classes, enabling targeted and more meaningful generation in applications where specific outputs are desired. Significant advancements have been made in cGANs for image generation tasks. Models such as AC-GAN [259] and Projection Discriminator [249] have shown proficiency in generating images with intricate textures and accurate color schemes. Despite these successes, these models often encounter difficulties in generating images with coherent global structures and capturing long-range dependencies. This limitation is primarily due to the inherent constraints of convolutional neural networks (CNNs), which focus on local spatial relationships and may not effectively model global context.

Addressing these limitations, the Self-Attention GAN (SAGAN) was introduced [402]. SAGAN incorporates self-attention mechanisms into both the generator and discriminator networks. The self-attention module enables the model to capture dependencies between widely separated regions of an image, facilitating the generation of more globally coherent and structurally consistent images. Additionally, SAGAN employs spectral normalization [248] to stabilize training dynamics and utilizes the Two-Time Scale Update Rule (TTUR) [116] to balance the learning rates of the generator and discriminator, further enhancing training stability and performance. Significant advancements have been made in cGANs for image generation tasks. Models such as AC-GAN [259] and Projection Discriminator [249] have shown proficiency in generating images with intricate textures and accurate color schemes. Despite these successes, these models often encounter difficulties in generating images with coherent global structures and capturing long-range dependencies. This limitation is primarily due to the inherent constraints of convolutional neural networks (CNNs), which focus on local spatial relationships and may not effectively model global context.

Addressing these limitations, Self-Attention GAN (SAGAN) [402] was introduced. SAGAN incorporates self-attention mechanisms into both the generator and discriminator networks. The self-attention module enables the model to capture dependencies between widely separated regions of an image, facilitating the generation of more globally coherent and structurally consistent images. Additionally, SAGAN employs spectral normalization [248] to stabilize training dynamics and utilizes the Two-Time Scale Update Rule (TTUR) [116] to balance the learning rates of the generator and discriminator, further enhancing training stability and performance. Additionally the StyleGAN series [145, 146, 144] introduced a style-based generator architecture that allows for unprecedented control over image synthesis. StyleGAN models enable fine-grained manipulation of image attributes at various levels of detail, leading to the generation of highly realistic and detailed images, especially in facial synthesis.

Diffusion Models Diffusion models have recently emerged as a groundbreaking approach in conditional image generation, setting new state-of-the-art benchmarks in the field [300, 10, 11]. These models function by gradually adding noise to images and then learning the reverse process to reconstruct the original data. In the past year, there have been significant developments in diffusion-based image synthesis. Stable Diffusion [299] has made high-quality image synthesis accessible even under resource constraints by performing the diffusion process in the latent space of autoencoders instead of directly in the pixel space. This approach reduces computational demands while maintaining impressive image fidelity.

Building upon earlier advancements, DALL-E2 [290] extends its predecessor [291] by integrating text-conditioned image embeddings from CLIP [284] into a diffusion model that acts as a decoder. This results in photorealistic images that accurately represent the input text and enables language-guided manipulation of source images. Imagen [311] takes a further step by utilizing large pre-trained language models like T5 [287] for text encoding, which guides the image synthesis through the diffusion process. This combination allows for generating images that closely align with complex textual descriptions, enhancing the semantic consistency of the output.

DreamBooth [308] builds on the foundation of Imagen by introducing context-aware image synthesis based on textual descriptions. This method allows for the creation of diverse visual subjects while preserving high image quality, enabling personalized and detailed image generation. More recently, models like eDiff-I [**diffusionediffi**] have further advanced diffusion-based image generation by incorporating more efficient training techniques and improved architectural designs. Additionally, works such as [267, 78, 326] have fine-tuned pre-trained Latent Diffusion Models (LDMs)[301] to effectively generate image sequences from textual narratives. Moreover, models like StoryLDM[289] and StoryGPT-V [301] leverage pre-trained LDMs for story visualization, but they approach the task with a modification: repeated character references in captions are replaced with pronouns (e.g., “he,” “she,” “they”). This adjustment challenges the models to maintain character consistency and interpret context despite the reduced explicitness in textual cues. These advancements have significantly enhanced the quality and speed of image synthesis, establishing diffusion models as a dominant force in generative modeling.

Transformers Transformer architectures have significantly advanced various areas of artificial intelligence, particularly in natural language processing and computer vision. In the context of story visualization, transformers have been employed to generate coherent sequences of images from narrative texts, effectively modeling the sequential and contextual dependencies inherent in stories. This has enhanced the generation of temporally consistent and semantically rich visual narratives [187]. Recent transformer-based approaches have further improved story visualization. The VP-CSV model [35] introduces a two-stage process: it begins by predicting visual tokens corresponding to character regions in images and then completes the backgrounds in a subsequent stage. This method enhances character representation while maintaining overall scene consistency. Another notable approach is CMOTA [6], which incorporates memory modules to improve consistency across generated image sequences. It utilizes a bidirectional strategy, performing both text-to-image and image-to-text transformations, allowing for online caption augmentation during training. This bidirectional learning enhances the model's understanding of the relationship between textual narratives and visual outputs, leading to more coherent visual stories.

Generative Evaluation Despite significant progress in image synthesis techniques, the evaluation of generative models has not kept pace and is hindered by reliance on outdated metrics [312, 115, 404, 118]. These traditional metrics are primarily used for benchmarking purposes but fail to address critical issues identified in recent studies [273, 25], such as capturing high-level semantic inconsistencies and being sensitive to minor perturbations that do not affect human perception. Recent advancements have aimed to develop more robust evaluation methods that consider semantic content and alignment with human judgments. For instance, metrics based on the Contrastive Language-Image Pre-training (CLIP) model [284] have been proposed to assess the correspondence between generated images and textual descriptions, providing a more nuanced evaluation of generative models' capabilities.

Explainability in generative modeling offers valuable insights; however, existing approaches are often model-specific [17, 255, 92, 152] or depend on the challenging task of discovering interpretable latent directions [322, 323, 32, 372]. Such methods typically require access to the internal architecture of the models or involve complex analyses of the latent space, limiting their applicability. Our proposed method addresses both the evaluation and explainability of generative models within a unified framework. It is adaptable to any generative model—including those designed for sequential image generation tasks [187, 233, 232, 347]—by focusing solely on the sets of concepts present in the input and output. By abstracting away from model-specific details and concentrating on conceptual content, our approach provides a generalizable and interpretable means of assessing generative models.

8.3 Methodology

In this section, we introduce the adapted framework for evaluating generative models based on concept-level analysis rather than traditional pixel-based metrics [218]. As before, the core idea is to compare the semantic content of the generated images with the conditioning inputs by extracting and analyzing the concepts present in both. This approach allows for a more interpretable and fine-grained assessment of generative models, particularly in tasks that involve complex semantic structures.

8.3.1 Overview of the framework

Our framework centers around a pre-trained, black-box generative model denoted as M. This model accepts a conditioning input c, which can be either a natural language description or a symbolic representation, and generates an image I intended to correspond to c. The conditioning input c provides semantic guidance to the generative model, dictating the content that should be present in the generated image.

To evaluate the alignment between the generated image I and the conditioning input c, we perform concept extraction on both. The process involves several key steps:

1. **Concept Extraction from Generated Image:** We apply state-of-the-art computer vision techniques, such as object detection [357, 294] and semantic segmentation [292, 36], to the generated image I. These methods enable us to identify and extract semantic concepts depicted in the image, such as objects, attributes, and their relationships. The extracted concepts are compiled into a set called the *generated* or *source* concept set, denoted as S.

2. **Concept Extraction from Conditioning Input:** The conditioning input c is processed to extract the intended semantic concepts. The extraction technique varies based on the format of c:

   - If c is a textual description, we use NLP tools, such as dependency parsing and named entity recognition, to extract nouns, verbs, adjectives, and other relevant linguistic elements that represent concepts.

   - If c is in a symbolic or structured format (e.g., a scene graph or a list of attributes), we perform direct parsing to obtain the set of concepts.

   The extracted concepts from c form the *real* or *target* concept set, denoted as T.

3. **Conceptual Comparison and Minimal Edits:** We aim to determine the minimal set of conceptual changes required to transform the generated concept set S into the target concept set T. This involves identifying concepts that need to be inserted, deleted, or replaced. The goal is to answer the question: *"What are the minimal required changes to traverse from S to T?"*

An overview of the proposed framework is illustrated in Figure 8.3.1, which depicts the flow from the conditioning input to the generation of the image and the subsequent concept extraction and comparison.

8.3.2 Conceptual edits as counterfactual explanations

Our research methodology is profoundly influenced by the study presented in [84, 58, 65], which investigates a pivotal aspect of counterfactual analysis: "What is the smallest alteration required for an image I to be reclassified from category Y to category X?" Here, X and Y represent the categories assigned by a predefined image classifier F. In our scenario, however, the classifier F is redundant because we automatically categorize all emerging concepts s into a set S, and all verified concepts t into another set T. This framework allows for counterfactual explanations to pinpoint the least number of conceptual modifications needed to transition from S to T for each s in S and t in T.

Concept Distances provide insights into the shortest route linking two distinct concepts. We utilize concept hierarchies to systematically determine the cost of transitioning between these concepts. This study examines

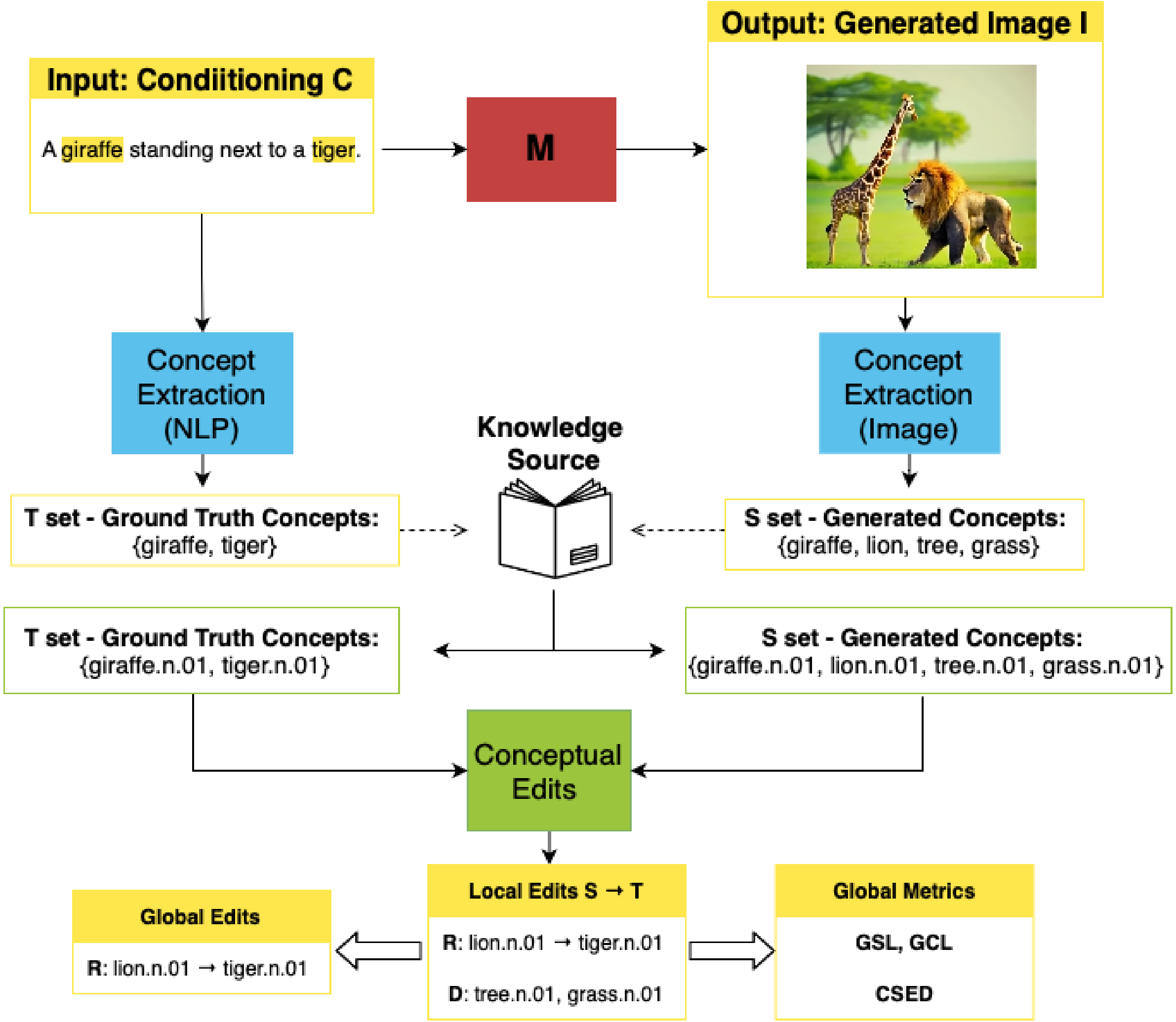

Input: Condiitioning C
A giraffe standing next to a tiger.
M
Output: Generated Image I
Concept Extraction (NLP)
Concept Extraction (Image)
Knowledge Source
T set - Ground Truth Concepts: {giraffe, tiger}
S set - Generated Concepts: {giraffe, lion, tree, grass}
T set - Ground Truth Concepts: {giraffe.n.01, tiger.n.01}
S set - Generated Concepts: {giraffe.n.01, lion.n.01, tree.n.01, grass.n.01}
Conceptual Edits
Global Edits
R: lion.n.01 → tiger.n.01
Local Edits S → T
R: lion.n.01 → tiger.n.01
D: tree.n.01, grass.n.01
Global Metrics
GSL, GCL
CSED

Figure 8.3.1: Overview of the proposed concept-based generative evaluation framework. The generative model M produces an image I based on conditioning input c. Concepts are extracted from both I and c to form sets S and T, respectively. The minimal edits required to align S with T are then determined.

two methodologies: incorporating external hierarchical structures like those found in WordNet [75], which links extracted concepts to defined synsets, and creating custom hierarchies to precisely control semantic distances. In either methodology, we denote $d(s,t)$ as the quantifiable distance between any two concepts s and t.

To facilitate these transitions, we introduce three specific types of **concept edit operations**:

- **Replacement (R)** $e_{s \to t}(S)$: This operation involves substituting a concept s in S with a new concept t not originally in S.
- **Deletion (D)** $e_{s-}(S)$: This involves removing a concept s from the set S.
- **Insertion (I)** $e_{t+}(S)$: This entails adding a new concept t from set T into set S.

These editing operations take into account the concept distances defined by our chosen hierarchies. Particularly, the **R** operation ensures the path chosen between s and t minimizes the distance $d(s,t)$, aligning with the principles of actionability as outlined in [84]. This ensures that the edits are both semantically meaningful (e.g., 'food' → 'pasta') and avoid nonsensical transitions (e.g., 'food' → 'sky'). The **D** and **I** operations consider the hierarchy's root node, which, in the case of using WordNet, is identified as entity.n.01.

The overall effectiveness of these transformations is measured by the **Concept Set Edit Distance (CSED)**

$D(S \rightarrow T)$, calculated by aggregating the minimal costs across all feasible edit operations required for converting set S into set T:

$$CSED = D(S \rightarrow T) = min \sum_{s \neq t}^{S,T} \sum^{R,D,I} d(s,t) \tag{8.3.1}$$

8.3.3 Counterfactual edits for generative evaluation

The counterfactual framework detailed in Section 8.3.2 underpins our methodology for generative evaluation, implemented on two complex tasks in generative research:

- **Story Visualization (SV)**
- **Scene Generation (SG)**

Story Visualization (SV)

The concept of **Story Visualization (SV)** involves the systematic generation of a series of images, $I_1, I_2, ..., I_L$, where each image corresponds to a specific segment of a narrative, $c_1, c_2, ..., c_L$, over a total narrative length L. This process requires each image to accurately reflect its corresponding narrative segment and maintain coherence throughout the series. We define two primary criteria for this process:

- **Faithfulness**: This criterion ensures that every object and attribute described in any narrative segment c_k is visually represented in the corresponding image I_k.
- **Consistency**: This ensures that once an object or attribute is introduced in any image I_k, it appears in all subsequent images up to I_L.

We utilize the CLEVR-SV dataset [139], structured around a set of attributes—shape (e.g., cube, sphere, cylinder), size (e.g., small, large), material (e.g., rubber, metal), and a selection of eight colors (e.g., blue, cyan, brown)—each object described by four attributes. We devise a simple hierarchical structure to categorize these attributes into broader conceptual categories, shown below:

$$\begin{aligned}
&\text{(large, small)} \subset \text{Size} \\
&\text{(blue, yellow, brown, grey, green, purple, cyan, red)} \subset \text{Color} \\
&\text{(metallic, rubber)} \subset \text{Material} \\
&\text{(sphere, cube, cylinder)} \subset \text{Shape}
\end{aligned} \tag{8.3.2}$$

The narrative structure in CLEVR-SV consists of four frames, with each frame escalating in complexity by the addition of objects. Transition operations between frames, namely Deletion (D), Insertion (I), and Replacement (R), are employed depending on the narrative requirements, each operation incurring a uniform cost.

To quantitatively assess the adherence to the narrative, we introduce the **Story Loss (SL)** metric, which aggregates the Concept Set Edit Distance ($CSED_k$) for each frame transition from S_k to T_k, reflecting the minimal edits required to align the generated image sequence with the narrative conditioning:

$$SL = \sum_{k=1}^{L} CSED_k = \sum_{k=1}^{L} D(S_k, T_k), \quad L = 4 \tag{8.3.3}$$

To evaluate narrative **consistency**, we propose the **Consistency Loss (CL)** metric. This metric examines semantic changes between consecutive frames, comparing each frame I_k with the prior frame I_{k-1}. Discrepancies are penalized, with the penalty reflective of deviations from the expected attribute count per frame:

$$CL = \sum_{k=2}^{L} D(S_k, S_{k-1}), \quad S_k = \text{concepts in } I_k, \quad S_{k-1} = \text{concepts in } I_{k-1} \tag{8.3.4}$$

For broader assessments, these metrics are aggregated over N narrative sequences to derive the **Global Story Loss (GSL)** and **Global Consistency Loss (GCL)**, enabling an evaluation of the generative model's overall performance in maintaining narrative fidelity and consistency across multiple stories:

$$\begin{aligned} GSL &= \sum_{i=1}^{N} SL_i, \\ GCL &= \sum_{i=1}^{N} CL_i, \end{aligned} \tag{8.3.5}$$

These metrics not only measure performance but also provide insights into specific areas where the model may fail to accurately or consistently represent narrative elements, serving as a diagnostic tool to identify frequent errors in story visualization.

Additionally, by obtaining the average values for both the local (**SL**/**CL**) and global (**GSL**/**GCL**) metrics, we can gain insights into the behavior of SV systems:

$$Avg\ SL = \frac{1}{k} SL, \qquad Avg\ GSL = \frac{1}{N}[Avg\ SL] = \frac{1}{N} GSL \tag{8.3.6}$$

Rather than calculating a straightforward average of $\sum CL_k$, a more insightful approach involves assessing how frequently the conditions $p_{k=1} = 0$ and $CL_{k>1} = |\mathcal{C}| \cdot (k-1)$ are violated, averaged across all $L = k$ frames:

$$Avg\ CL = \frac{p_{k=1}}{k} + \frac{1}{k} \sum_{k=1}^{k=L} [CL_{k>1} \neq |\mathcal{C}| \cdot (k-1)], \quad Avg\ GCL = \frac{1}{N}[Avg\ CL] \tag{8.3.7}$$

The metrics **SL** and **CL** inherently provide *explainable insights* as they not only gauge quality but also illuminate the $S_k \to T_k$ edit pathways. These paths serve as *local counterfactual explanations* that underscore the erroneously generated semantics within the story, pertaining to either **faithfulness** or **consistency**.

Higher values of **SL**/**GSL** and **CL**/**GCL** typically indicate poorer conceptual generation quality. Paths identified in **GSL**/**GCL** serve as *global counterfactual explanations*, where rule extraction techniques reveal common patterns that summarize the behavior of the model under study. These frequently observed **GSL** pathways often encompass *common misconceptions*, such as conditioning concepts that are challenging for the model to accurately generate. Similarly, **GCL** pathways often expose *inconsistency patterns*, displaying concepts that change unpredictably throughout the story frames.

Thus, by exploring "What minimal changes are needed to transition from S to T?", we ultimately address a broader question: "Which concepts are challenging for the model to generate or maintain consistently?"

Scene Generation (SG)

Scene Generation (SG) is tasked with creating a visual representation I from a complex narrative input c. This synthesis involves the integration of various interactive elements within the scene, each characterized by distinct attributes. Unlike simpler visual tasks, the narrative input for SG presents a complex array of elements that are not fixed in advance, leading to a dynamically large set of potential concepts $\mathcal{C}$.

The COCO dataset [199] is utilized to assess the **faithfulness** of generated scenes, using textual descriptions c as a basis for the scene creation. Our analysis focuses on cutting-edge diffusion models sourced from Huggingface[2], specifically Stable Diffusion versions 1.4 and 2 [330, 329], and Protogen versions x3.4 and 5.8 [279, 280]. These models are selected for their capability to render *high-fidelity* images, which is crucial for subsequent concept extraction (object detection). Previous architectures [272, 397, 225, 184, 335, among others] are excluded from our study due to their lesser visual quality and dependence on scene graphs for composition.

During the concept extraction phase, object detection is performed using YOLO-v8 [137] and YOLOS [73], which facilitate the construction of the concept set S from the generated scenes. Textual narratives c are

[2] https://huggingface.co/models?pipeline$_tag = text - to - imagesort = downloads$

processed using spaCy [125] to delineate the *target concept set* T. Given the intricate semantic relationships inherent to the COCO dataset's concepts, an expansive knowledge base like WordNet is indispensable. For instance, a narrative input c might mention generic categories like 'food' or 'animal', which the models might specify further into 'pasta' or 'dog'. These refined categories, detected by object detectors, may introduce discrepancies. However, hierarchical knowledge bases help bridge these gaps by confirming semantic equivalence between sets; for example, despite $T = \{food, animal\} \neq S = \{pasta, dog\}$, the relation $pasta - isA - food$ and $dog - isA - animal$ ensures semantic congruence. Therefore, no transformation between S and T is necessary. Additionally, WordNet aids in accurately quantifying the semantic distances essential for editing operations, thereby facilitating the calculation of the total transformation cost via the Concept Set Edit Distance (CSED).

8.4 Experiments

8.4.1 Story Visualization

In the realm of story visualization, each semantic aspect and edit operation—namely Deletions (**D**) and Insertions (**I**)—is quantified uniformly with a cost, denoted by $d = 1$ for each semantic feature and edit action. This pricing model simplifies the calculation of edit distances. Specifically, the removal of a color attribute is quantified with an edit cost of 1. Similarly, replacing one color with another results in a cumulative edit cost of 2, which is the sum of deleting the initial color and inserting the new one. This same costing principle is applied uniformly across other attributes such as shape, size, and material of the objects involved in the visualization process.

Results from leading variants of selected story visualization (SV) models [187, 233, 232, 347] are summarized in Table 8.1. To provide a comprehensive evaluation, traditional metrics like Fréchet Inception Distance (FID), Clean-FID, Learned Perceptual Image Patch Similarity (LPIPS), and Structural Similarity Index Measure (SSIM) are included for a detailed comparison.

Typically, there is a noticeable correlation between metrics assessing pixel-level details and those evaluating conceptual integrity. This correlation is anticipated as the extraction of concepts is inherently dependent on the clarity and accuracy of the pixel-level representation in images. Objects and semantic elements that are generated with higher fidelity are more likely to be correctly identified and classified during the concept extraction phase.

Moreover, the conceptual analysis provides *explainable insights* into the performance of these models. Detailed percentages of losses per conceptual category (Material, Size, Shape, Color) are enumerated, shedding light on the specific areas where each model excels or falters. For instance, a common observation across all models is a significant Shape loss, often exceeding 50%. This indicates a prevalent difficulty in synthesizing objects with well-defined shapes. Conversely, the models generally perform better in terms of Size accuracy, as evidenced by comparatively lower Size losses. This suggests that while the models struggle with shape precision, they are more adept at replicating the correct size of objects, indicating a partial but significant success in adhering to the dimensional aspects of the input specifications.

M	FID	Clean	LPIPS	SSIM	GCL	GSL	Material	Size	Shape	Color
	↓	-FID↓	↓	↑	↓	↓	↓	↓	↓	↓
[347]	41.54 ± 8.55	**115.46**	**0.21** ± 0.05	**0.71**	**4.97**	**7.01**	**20.83%**	**14.55%**	**56.62%**	**33.10%**
[187]	**41.45** ± 6.25	123.40	0.25 ± 0.03	0.65	11.44	15.33	30.89%	21.12%	62.34%	37.44%
[232]	41.96 ± 9.66	124.97	0.25 ± 0.08	0.67	10.95	8.06	21.45%	16.02%	56.78%	35.10%
[233]	41.80 ± 8.81	122.62	0.25 ± 0.05	0.68	8.32	11.51	25.34%	16.71%	56.83%	35.14%

Table 8.1: Average evaluation metrics (existing and proposed, separated by a vertical line) for the generation on the CLEVR-SV dataset [139] for all L=4 stories per M.

We continue our analysis by concentrating on the highest-performing story visualization model from [347], as indicated by the conceptual metrics in Table 8.1. In particular, Table 8.2 details the outcomes for each frame's **GSL**, **GCL**, and the losses associated with each concept (Material, Size, Shape, Color).

Frame	GSC ↓	GSL ↓	Material ↓	Size ↓	Shape ↓	Color ↓
1st	0.00	2.25	40.00%	6.20%	58.75%	7.50%
2nd	4.35	5.66	20.00%	11.88%	57.5%	32.50%
3rd	7.12	8.25	13.33%	16.67%	57.08%	43.33%
4th	8.42	11.49	10.00%	23.44%	53.13%	49.06%

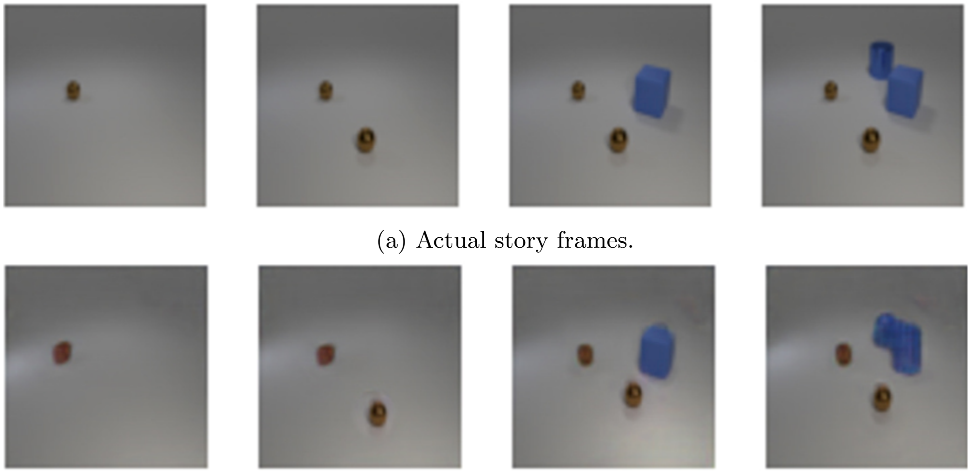
Table 8.2: Average conceptual evaluation metrics per frame for [347].

Local explanations The effectiveness of the **SL**/**CL** metrics is demonstrated through *local explanations* for the model from [347], specifically examining edit pathways in the sequence depicted in Figure 8.4.1. The sequence includes the first four images (Figure 8.4.1a) representing the actual sequence, while the subsequent four images (Figure 8.4.1b) show the modeled outputs. Here, S represents the concepts from Figure 8.4.1b, and T contains those from Figure 8.4.1a.

Detailed in Table 8.3, a consistent **R** operation is noted across all frames, where the transformation is made from a *'rubber'* to a *'metallic'* material for a small brown sphere to align with the ground truth. In the final frame, an additional **R** operation is required to change the object's shape from *'sphere'* to *'cylinder'*. The cost assigned to each **R** operation is 2, reflecting the dual steps of removing an incorrect attribute and adding a correct one. This cost metric can be adjusted if necessary. The cumulative **SL** for this scenario is calculated to be 10, summing the costs of all operations across the frames.

Regarding the **CL**, the correct increment of objects across subsequent frames is noted, ensuring that the formula $CL_{k>1} = |\mathcal{C}| \cdot (k-1), |\mathcal{C}| = 4$ holds. Starting with $CL_1 = p_{k=1} = 0$ for the first frame, it is confirmed that just one object is added, adhering to the rule that the frame number corresponds to the quantity of objects. $CL_2 = 4$ is expected as the second frame introduces an object encompassing four semantic attributes. A deviation from these expected numbers would suggest an error: $CL_{k>1} < |\mathcal{C}| \cdot (k-1)$ would indicate missing objects, whereas $CL_{k>1} > |\mathcal{C}| \cdot (k-1)$ would point to extra objects being generated. This pattern is consistently applied up to the fourth frame.

This dissection reveals critical weaknesses of the [347] model, particularly in accurately rendering the *Material* semantic, as throughout all frames in this example, the small brown sphere is inaccurately depicted with *'rubber'* instead of *'metallic'*. To fully understand the model's capability to accurately synthesize individual semantics, broader metrics and evaluations are essential.



(a) Actual story frames.

(b) Model-generated story frames of [347].

Figure 8.4.1: Comparison of actual vs model-generated CLEVR-SV story frames using [347] for L=4.

To better understand how the proposed algorithm functions, we will conduct an in-depth examination of the local properties of CSED per frame for SV, as depicted in the sequence shown in Figure 8.4.1. This sequence is considered of medium difficulty according to the methodology described in [347], particularly because in the fourth frame, the blue cylinder is superimposed with the blue cube. We will compare the actual semantic details of the sequence, represented by the ground truth frames (Figure 8.4.1a), with the semantics of the generated frames (Figure 8.4.1b).

We have arrived at the following conclusions:

Frame k=1
Ground truth semantics: {[small, brown, **metallic**, sphere]}
Generated semantics: {[small, brown, **rubber**, sphere]}
The discrepancies between the two sequences are evident in the third semantic term; the original is 'metallic', while the generated term is 'rubber'. Thus, for the first frame, $CSED_{k=1}$ suggests a substitution of 'rubber' with 'metallic', carrying an edit cost of 2 = $CSED_{k=1}$. This alteration, related to the *Material* semantic, leads to an increment in the *Material Loss* count, which will further elucidate global semantic synthesis failures across all test set frames.

For frame 1, the Consistency Loss (CL) is zero for the generated sequence since there are a total of $|\mathcal{C}|$=4 semantics (*Material, Size, Shape, Color*), and one object containing T =4 semantics occupies position k=1 in the sequence, resulting in $CL_{k=1} = p_{k=1} = |T| - |\mathcal{C}| \cdot k$=4-4=0.

Frame k=2
Ground truth semantics: {[small, brown, **metallic**, sphere], [small, brown, metallic, sphere]}
Generated semantics: {[small, brown, **rubber**, sphere], [small, brown, metallic, sphere]}
Again, the semantic difference in the third position remains highlighted; the original is 'metallic' while the substitute is 'rubber', prompting $CSED_{k=2}$ to propose the same replacement as before with an edit cost of 2. Additionally, as this change pertains to the *Material* semantic, another failure is recorded on the *Material Loss* counter.

Simultaneously, CL increases simply by adding an object containing $|\mathcal{C}|$=4 semantics, which means the minimal increase in CL when an object is added to CLEVR-SV is 4. Comparing $k = 1$ generated sequence $T = S_{k-1}$={[small, brown, rubber, sphere]} with the $k = 2$ generated sequence $S = S_k$={[small, brown, rubber, sphere], [**small, brown, metallic, sphere**]} shows no additional discrepancies. Applying equation 8.3.4 for k=2 yields:

$$CL_{k=2} = p_{k=1} + D(S_{k=2}, T_{k=2}) = 0 + \mathbf{I}\{small, brown, metallic, sphere\} = 0{+}4 = 4$$

Frame k=3
Ground truth semantics: {[small, brown, **metallic**, sphere], [small, brown, metallic, sphere], [large, blue, rubber, cube] }
Generated semantics: {[small, brown, **rubber**, sphere], [small, brown, metallic, sphere], [large, blue, rubber, cube]} The semantic divergence in the third position continues, leading to a proposed replacement of 'rubber' with 'metallic' for an edit cost of 2. This change also contributes to an increase in the *Material Loss* counter.

CL accounts for the comparison between the $T = 2$ generated sequence [small, brown, rubber, sphere, small, brown, metallic, sphere] and the $k = 3$ generated sequence {[small, brown, rubber, sphere], [small, brown, metallic, sphere], [**large, blue, rubber, cube**]}, differing only by the addition of the *large, blue, rubber, cube* in the third frame, resulting in:

$$CL_{k=3} = p_{k=1} + D(S_{k=2}, T_{k=2}) + D(S_{k=3}, T_{k=3}) = 0 + 4 + \mathbf{I}\{large, blue, rubber, cube\} = 0{+}4{+}4 = 8$$

Frame k=4
Ground truth semantics: {[small, brown, **metallic**, sphere], [small, brown, metallic, sphere], [large, blue, rubber, cube], [large, blue, metallic, **cylinder**]}
Generated semantics: {[small, brown, **rubber**, sphere], [small, brown, metallic, sphere], [large, blue, rubber, cube], [large, blue, metallic, **sphere**]}
Beyond the consistent discrepancy in the third position, there's an additional variation in the last position,

prompting a replacement of 'sphere' with 'cylinder', each carrying an edit cost of 2. Summing these transformations results in a total transformation for $k = 4$: {'rubber', 'sphere'} → {'metallic', 'cylinder'} with an edit cost of 4. The counters for *Material Loss* and *Shape Loss* increase by one each.

CL calculations for the sequences corresponding to $k = 3$ generated sequence {[small, brown, rubber, sphere], [small, brown, metallic, sphere], [large, blue, rubber, cube]} and $k = 4$ generated sequence {[small, brown, rubber, sphere], [small, brown, metallic, sphere], [large, blue, rubber, cube], [**large, blue, metallic, sphere**]} reveal only the addition of the *large, blue, metallic, sphere* item. Therefore, $CL_{T=4}$= 4, resulting in:

$CL_{k=4} = p_{k=1} + D(S_{k=2}, T_{k=2}) + D(S_{k=3}, T_{k=3}) + D(S_{k=4}, T_{k=4}) = 0+4+4+\mathbf{I}\{large, blue, metallic, sphere\}$ $= 0+4+4+4 = 12$

By summing up, *Story Loss (SL)* as the cumulative per-frame CSED costs will be:

$$SL = 2 + 2 + 2 + 4 = 10$$

and averaging SL across all $L = 4$ frames according to equation 8.3.6 results in:

$$Average\ SL = \frac{1}{k} SL = 10/4 = 2.5$$

Following equation 8.3.7 for consistency, we find:

$$Average\ CL = \frac{p_{k=1}}{k} + \frac{1}{k} \sum_{k=1}^{k=L} [CL_{k>1} \neq |\mathcal{C}| \cdot (k-1)] = 0 + 0 = 0$$

The generated narrative of Figure 8.4.1 is **fully consistent** as the *Average CL* equates to zero, indicating an ideal scenario with no semantics being altered or omitted in the generated sequence. However, it's noteworthy that CL fails to capture the **faithfulness** error introduced by the new item in the fourth frame: while the actual object is a *large, blue, metallic, cylinder*, the generated sequence adds a *large, blue, metallic, sphere*, yet CL does not penalize the semantic discrepancy in the last position. Conversely, SL serves to penalize this error, demonstrating that both metrics are crucial, with SL focusing on fidelity between actual and generated narratives, and CL on consistency across consecutively generated frames. Optimal models would exhibit *lower* values for both metrics globally.

Frame	Min edit path	Operation	Edit cost	Semantic	CL
1st	"rubber" → "metallic"	**R**	2	Material	0
2nd	"rubber" → "metallic"	**R**	2	Material	4
3rd	"rubber" → "metallic"	**R**	2	Material	8
4th	{"rubber", "sphere"} → {"metallic", "cylinder"}	**R**, **R**	4	Material, Shape	12

Table 8.3: Interpretable edits for Figure 8.4.1 based on local analysis.

Global explanations To fully understand the limitations of our model, we calculate the **GSL** across all images in the CLEVR-SV dataset to gauge the model's difficulty in accurately representing specific -discrete-semantics, both individually per frame and cumulatively (refer to Table 8.1). Interestingly, while we might expect *Material loss* to increase as more objects complicate the frame, it actually diminishes over successive frames. In contrast, *Size loss* and *Color loss* demonstrate the anticipated upward trend in losses. Patterns in *Shape loss*, however, are less predictable and remain considerably high.

The persistently high *Shape loss* underscores the necessity for integrating attention mechanisms in our GANs [402] to better capture long-range dependencies within the images. The notable escalations in *Size* and *Color* losses suggest inconsistencies in maintaining continuity throughout the story sequence.

Additionally, **GSL** uncovers underlying patterns across the entire dataset, which we explore using the apriori algorithm [5] to identify frequent semantic rules and combinations. The four most prevalent semantic edits are outlined in Table 8.4, along with the frequency (support) of each rule. The table also lists the concept

Rules (edits)	Semantic	Support %	Antec. support%	Conseq. support%
'metallic' →'rubber'	Material	26.77	26.77	26.77
'rubber' →'metallic'	Material	22.05	22.05	22.83
'cylinder' →'cube'	Shape	18.11	33.07	31.50
'cylinder' →'sphere'	Shape	14.96	33.07	18.90

Table 8.4: Global interpretive edits derived from the CLEVR-SV test set using [347].

category (derived from equation 8.3.2), along with the antecedent support (frequency of the source semantic) and consequent support (frequency of the target semantic).

Material emerges as the most frequently misunderstood concept, particularly with 'rubber' and 'metallic' often being interchanged. *Shape* follows as the second major area of confusion, with 'cylinder' more frequently appearing in generated frames than in the conditioning frames, often at the expense of 'cube' and 'sphere' shapes. Although the rule support does not reach particularly high levels—peaking at 26.77%—this suggests that the SV model from [347] does not show a strong bias toward specific semantics. However, the prevalent errors in material and shape generation provide critical insights that could be instrumental for future architectural enhancements of the model.

8.4.2 Scene Generation

To streamline the inference process, we utilize the first 10,000 samples from the COCO dataset, employing YOLO-v8 and YOLOS for visual concept extraction. Notably, each COCO sample is accompanied by five descriptive sentences, essentially paraphrasing one another. We thus use only the first sentence as the conditioning variable c. Our approach for Scene Generation (SG) involves two distinct methodologies: direct *generation* based on c and *retrieval* of image-caption pairs that closely match c.

Conditional generation on COCO captions In our experimental setup, we deploy pre-trained diffusion models to generate images, as outlined in 8.3.3. This experiment did not involve additional model tuning. The generation of 10,000 images took approximately 15 hours for each model using dual T4 GPUs, totaling about 60 hours of processing time.

Retrieval of COCO-related captions To amass a larger collection of images related to COCO captions without the extensive resource expenditure of additional diffusion model runs, we turned to the Stable Diffusion search engine at Lexica.art[3]. Here, we entered the first sentence of each of the 10,000 COCO samples as search queries. The search engine provided us with 10 images for each query, previously generated by the community, closely aligning with our captions. This method furnished an additional 100,000 images from Stable Diffusion, each tagged with their respective input queries. We subsequently conducted a comparative analysis between these web-retrieved images and those generated by our models.

Object detection We set a detection threshold of $T_d = 0.6$, where only objects detected with a confidence of 0.6 or higher are considered for inclusion in the concept set S. This threshold was chosen to balance the occurrence of false positives and negatives, as establishing a baseline for false detections is challenging without manual review. Nonetheless, our setup offers insights into potential detection errors: a lower threshold might suggest an increased likelihood of false positives (irrelevant objects detected), whereas a stricter threshold could indicate more false negatives (missed relevant objects).

Metric results We present comparative detection results using thresholds of $T_d = 0.5$, 0.6, and 0.7 across Tables 8.5 (YOLO-v8) and 8.6 (YOLOS) for the generated images, and in Table 8.7 for the web-retrieved images. These tables detail the frequency of edits (# **I**, # **D**, # **R**) and the associated costs for each type of operation. Instances with the lowest scores, preferred in our evaluation, are marked in **underlined** text, while the highest scores are indicated in **bold**. The overall mean Concept Set Edit Distance (CSED) is also reported, providing a holistic view of operational frequency and effectiveness.

[3] https://lexica.art/

T_d	M	# **I**	Cost **I**	# **D**	Cost **D**	# **R**	Cost **R**	Mean CSED
0.5	stable diffusion	37651	16762	1196	5655	126004	14323	35.75
	stable diffusion 2	***36878***	**16067**	**1243**	**6301**	**129315**	**14839**	**36.32**
	protogen base	37072	16208	1233	5944	129290	14744	35.95
	protogen 5.8	**38581**	**17715**	**1195**	**4702**	**117708**	**13411**	**34.66**
0.6	stable diffusion	39070	18386	1157	4042	110260	12964	34.22
	stable diffusion 2	38678	**17782**	**1200**	**4514**	112499	13397	**34.55**
	protogen base	**38548**	17794	1184	4270	**114762**	**13427**	34.35
	protogen 5.8	**39766**	**19210**	**1134**	**3419**	**103579**	**12135**	**33.38**
0.7	stable diffusion	40814	20391	1086	2681	93390	11337	32.96
	stable diffusion 2	40677	19806	**1107**	**2938**	95477	11756	**33.08**
	protogen base	**40397**	**19801**	1101	2820	**97314**	**11787**	32.94
	protogen 5.8	**41295**	**20944**	**1039**	**2308**	**89850**	**10726**	**32.39**

Table 8.5: Metric results using YOLO-v8 [137] for object detection on generated images from COCO queries.

T_d	M	# **I**	Cost **I**	# **D**	Cost **D**	# **R**	Cost **R**	Mean CSED
0.5	stable diffusion	**26302**	9032	1382	44189	**197623**	**21097**	68.25
	stable diffusion 2	26684	**8832**	1403	43459	192198	21082	68.05
	protogen base	26887	8966	**1404**	**44406**	193327	21035	**68.81**
	protogen 5.8	**28880**	**10367**	**1373**	**34996**	**189677**	**19858**	**60.45**
0.6	stable diffusion	**27963**	9920	1373	33891	**188395**	20286	60.10
	stable diffusion 2	28145	**9662**	**1394**	33933	182767	**20322**	60.36
	protogen base	28499	9845	**1394**	**34167**	185217	20224	**60.63**
	protogen 5.8	**30545**	**11330**	**1364**	**27218**	**179947**	**18963**	**54.13**
0.7	stable diffusion	**29998**	**10985**	1357	**24956**	177213	**19319**	**52.51**
	stable diffusion 2	**29831**	10657	1347	25492	**172860**	19409	53.14
	protogen base	29866	10790	**1346**	25255	175495	19350	52.98
	protogen 5.8	28880	**10367**	**1373**	**34996**	**189677**	**19858**	**60.45**

Table 8.6: Metric results using YOLOS [73] for object detection on generated images from COCO queries.

T_d	Obj. detector	# **I**	Cost **I**	# **D**	Cost **D**	# **R**	Cost **R**	Mean CSED
0.5	YOLO-v8	186775	857448	1343	52247	1353479	224350	75.87
	YOLOS	163628	605321	1487	421525	2469635	473331	106.41
0.6	YOLO-v8	190047	891454	1317	37418	1174012	190928	73.74
	YOLOS	167576	646112	1467	308346	2303966	432851	98.06
0.7	YOLO-v8	193663	929183	1236	25388	982259	154063	71.81
	YOLOS	171778	688942	1449	214928	2115779	390304	90.56

Table 8.7: Metric results for web-retrieved Stable Diffusion [330] images on similar queries to COCO.

Local explanations illustrate the edit processes **I**, **D**, and **R** employed to modify specific images, as shown in the scene from Figure 8.4.2.

Utilizing YOLO-v8 with a set detection threshold of $T_d = 0.6$, the identified concepts in the image are

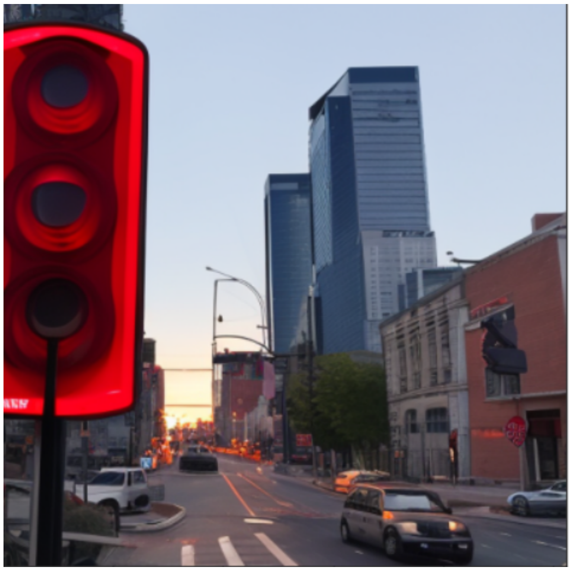


Figure 8.4.2: A sample image generated using Stable Diffusion 2 [329], used for deriving local explanations.

S={'car', 'car', 'traffic light', 'car', 'stop sign'}, while the actual concepts intended are T={'light', 'buildings'}. The transformation from S to T involves the edit operations shown in Table 8.8, which accumulate to a total minimal cost of 59.00.

Operation	**Details**
Insertions (I)	{}
Deletions (D)	{'car', 'car', 'car'}
Replacements (R)	{'traffic light' → 'light', 'stop sign' → 'buildings'}

Table 8.8: Edit operations for YOLO-v8 [137] concepts.

When using YOLOS, the generated concepts are S={'car', 'traffic light', 'car', 'stop sign', 'traffic light', 'car', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'car', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'car', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'car', 'car', 'traffic light', 'traffic light'}, and the ground truth ones are T={'light', 'buildings'}. By visually inspecting the image, YOLOS clearly overestimates the actual objects present, inducing noise in the generated concept set S. Nevertheless, our evaluation strategy successfully captures this overestimation, by suggesting the deletion of multiple concepts. Specifically, the edit operations shown in Table 8.9 result in transformations with a total cost of 104.04.

Operation	**Details**
Insertions (I)	{}
Deletions (D)	{'car', 'traffic light', 'car', 'traffic light', 'car', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'car', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'car', 'traffic light', 'traffic light', 'traffic light', 'traffic light', 'car', 'car', 'traffic light', 'traffic light'}
Replacements (R)	{'stop sign' → 'light', 'traffic light' → 'buildings'}

Table 8.9: Edit operations for YOLOS [73] concepts, total cost 104.04.

Global explanations are detailed for all evaluated images, where edits involving insertions (**I**) and deletions (**D**) are catalogued in Table 8.10, and replacements (**R**) are documented in Table 8.11. Only concepts extracted using YOLO-v8 are included due to YOLOS generating excessively high counts of detections. We highlight the top three most common types of each edit: insertions, deletions, and replacements. For **I** and **D**, "Freq **I**", "Freq **D**" measure the occurrence of particular concepts being inserted or deleted across the dataset. The support for **I** and **D** indicates how often these edits occur relative to all such edits in the dataset. In the case of **R**, support quantifies how often a specific transformation rule appears relative to all transformation rules.

The analysis shows a clear consensus across models; the **I** edits frequently include the concepts ’street’, ’tennis’, and ’table’. It appears that model M struggles to adequately render these **I** concepts, or the concepts are produced with such low visual quality that their detection proves unreliable at thresholds of T_d=0.5, 0.6, 0.7. **D** edits often involve the concepts ’person’, ’sheep’, ’car’, ’umbrella’, and ’donut’, suggesting a tendency of the model to generate unnecessary instances of these categories. Lastly, the **R** edits typically involve changing ’person’ to ’people’, ’man’, or ’woman’, which is somewhat anticipated given that ’person’ in YOLO categorization includes both genders.

T_d	M	**I**	Freq **I**	**I** support	**D**	Freq **D**	**D** support
0.5	stable diffusion	street	264	1.57%	person	2075	36.69%
		table	250	1.49%	sheep	363	6.42%
		tennis	247	1.47%	car	252	4.46%
	stable diffusion 2	tennis	253	1.57%	person	2177	34.55%
		street	242	1.51%	sheep	466	7.40%
		table	237	1.48%	car	313	4.97%
	protogen base	tennis	247	1.52%	person	2281	38.37%
		street	244	1.51%	sheep	317	5.33%
		table	229	1.41%	car	311	5.23%
	protogen 5.8	table	270	1.52%	person	1564	33.26%
		tennis	265	1.50%	car	261	5.55%
		street	241	1.36%	umbrella	251	5.34%
0.6	stable diffusion	street	290	1.58%	person	1572	38.89%
		table	281	1.53%	sheep	311	7.69%
		tennis	259	1.41%	car	158	3.91%
	stable diffusion 2	table	274	1.54%	person	1656	36.69%
		street	269	1.51%	sheep	376	8.33%
		tennis	264	1.48%	car	203	4.50%
	protogen base	street	268	1.51%	person	1717	40.21%
		table	261	1.47%	sheep	254	5.95%
		tennis	255	1.43%	car	197	4.61%
	protogen 5.8	table	303	1.58%	person	1220	35.68%
		tennis	278	1.45%	sheep	198	5.79%
		street	274	1.43%	umbrella	176	5.15%
0.7	stable diffusion	table	322	1.58%	person	1075	40.10%
		street	316	1.55%	sheep	254	9.47%
		tennis	268	1.31%	donut	122	4.55%
	stable diffusion 2	table	313	1.58%	person	1134	38.60%
		street	301	1.52%	sheep	291	9.90%
		tennis	267	1.35%	donut	111	3.78%
	protogen base	street	300	1.52%	person	1189	42.16%
		table	289	1.46%	sheep	188	6.67%
		tennis	262	1.32%	umbrella	143	5.07%
	protogen 5.8	table	330	1.58%	person	884	38.30%
		street	299	1.43%	sheep	152	6.59%
		tennis	287	1.37%	umbrella	130	5.63%

Table 8.10: Global explanations (**I** and **D** edits) for YOLO-v8 [137] extracted concepts.

T_d	M	**R**	Freq **R**	**R** support	M	**R**	Freq **R**	**R** support
0.5	stable diffusion	person → man	1090	7.61%	stable diffusion 2	person → man	1115	7.51%
		person → people	520	3.63%		person → people	551	3.71%
		person → woman	499	3.48%		person → woman	511	3.44%
	protogen base	person → man	1101	7.47%	protogen 5.8	person → man	1061	7.91%
		person → people	507	3.44%		person → woman	476	3.55%
		person → woman	500	3.39%		person → people	441	3.29%
0.6	stable diffusion	person → man	1065	8.22%	stable diffusion 2	person → man	1087	8.11%
		person → people	503	3.88%		person →people	536	4.00%
		person → woman	481	3.71%		person → woman	482	3.60%
	protogen base	person → man	1080	8.04%	protogen 5.8	person → man	1035	8.53%
		person → people	494	3.68%		person → woman	449	3.70%
		person → woman	485	3.61%		person → people	431	3.55%
0.7	stable diffusion	person → man	1022	9.01%	stable diffusion 2	person → man	1033	8.79%
		person → people	473	4.17%		person → people	508	4.32%
		person → woman	458	4.04%		person → woman	441	3.75%
	protogen base	person → man	1054	8.94%	protogen 5.8	person → man	989	9.22%
		person → woman	461	3.91%		person → woman	419	3.91%
		person → people	446	3.78%		person → people	408	3.80%

Table 8.11: Global explanations (**R** edits) for YOLO-v8 [137] extracted concepts.

8.5 Conclusion

The exploration of conceptual methodologies in the field of generative evaluation is still relatively nascent, yet it promises significant insights into both the quality of models and the clarity with which results can be interpreted. In this chapter, we introduce a framework that leverages knowledge-driven principles for explainable evaluation. This framework is designed to pinpoint specific conceptual adjustments needed within generated images—identifying which concepts should be introduced, omitted, or altered—to make these images more closely resemble their original design specifications.

Our empirical results, derived from engaging with complex tasks such as Story Visualization and Scene Generation, have demonstrated the practical benefits of this approach. Specifically, these results highlight critical gaps where models consistently fail to generate certain concepts, as well as tendencies of models to produce an excessive number of particular concept categories. Such insights not only enhance our understanding of the inherent model biases but also guide the development of more balanced and accurate generative models.

As we look to the future, our goal is to broaden the application of this evaluation framework to encompass a wider range of models and computational tasks. Additionally, we plan to enrich our framework by incorporating a variety of alternative knowledge sources. This expansion will allow us to conduct a deeper, more nuanced analysis of how the edits suggested by our framework conceptually diverge from those generated by current model configurations. This continued research will contribute to refining the methodologies used in generative evaluation and push the boundaries of what is possible in explainable artificial intelligence.

Chapter 9

Explainable Metric for Hallucination Detection in Image Captioning

This chapter introduces a critical exploration within the dynamic field of artificial intelligence, focusing on the phenomenon of hallucinations in vision-language (VL) models. This issue is the reverse of the problem analyzed in Chapter 8, where we tried to identify errors in image synthesis. Here, however, we will adapt the algorithms presented in Chapters 4 and 5 to capture hallucinations specifically in one of the more commonly used tasks of VL, namely image captioning. As these models become increasingly integral to various applications, understanding and addressing their limitations is essential. Hallucinations in image captioning, where the model generates inaccurate or irrelevant descriptions, pose significant challenges for the reliability and trustworthiness of VL systems.

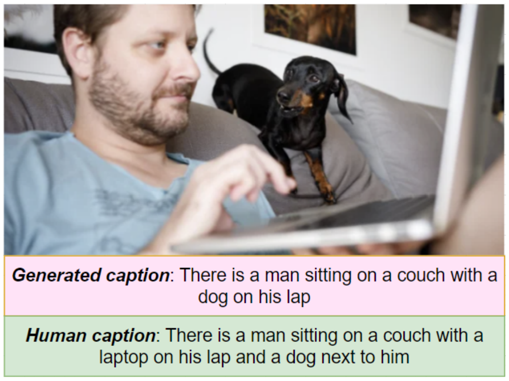
In this context, we delve into the intricacies of hallucinatory phenomena exhibited by widely used image captioners, identifying and analyzing interesting patterns. Building on previously introduced techniques, this chapter discusses the application of conceptual counterfactual explanations to effectively address VL hallucinations. We employ a deterministic and efficient backbone of conceptual counterfactuals, which suggests semantically minimal edits. These edits are driven by hierarchical knowledge, facilitating the transition from a hallucinated caption to a non-hallucinated one in a black-box manner. Our proposed hallucination detection framework enhances interpretability by providing semantically meaningful edits rather than standalone numerical values. This approach allows for a deeper understanding of the underlying causes of hallucinations through a hierarchical decomposition of hallucinated concepts. Additionally, this chapter introduces the novel concept of role hallucinations, which involves the interconnections between visual concepts, marking a first in the field of hallucination detection.

Overall, the methodologies and insights presented in this chapter recommend an explainable and trustworthy approach to VL hallucination detection. This is vital for evaluating the performance, identifying potential problems and risks of current and future VL systems.

9.1 Introduction

In the dynamic arena of artificial intelligence, the emergence of hallucinations in outputs has surfaced as a noteworthy challenge. While neural models exhibit exceptional linguistic and visual capabilities, their outputs sometimes deviate unexpectedly, mixing accurate depictions with imaginative elements. The topic of hallucinations has gained recent attention in Natural Language Processing (NLP), especially with Large Language Models (LLMs) generating outputs that often diverge from factual accuracy despite their extensive training parameters and vast data sets [123, 409, 344, 104, 141].

Hallucinations in output are problematic to detect due to their varied nature. [409] classify three primary types of hallucinations: Input-Conflicting, where LLM outputs do not align with the input prompt; Context-Conflicting, which includes contradictions within the output itself; and Fact-Conflicting, where outputs con-



Generated caption: There is a man sitting on a couch with a dog on his lap

Human caption: There is a man sitting on a couch with a laptop on his lap and a dog next to him

Figure 9.1.1: Illustration of a hallucination in image captioning, where the generated caption inaccurately describes the scene. The term "laptop" should replace "dog," and the phrase "next to" should better link the concepts of "dog" and "man."

tain false information.

Despite growing interest, the exploration of hallucinations in multimodal contexts, such as vision-language (VL) models, remains underdeveloped. As these models evolve into Large VL Models (LVLMs) [204, 423, 40], their enhanced capabilities are marred by increased occurrences of unreliable outputs, which are harder to detect due to ambiguities within and between modalities.

The scant research on VL model hallucinations has begun to tackle essential questions regarding their evaluation [298, 185, 361, 135] and reduction [418, 344, 173, 200]. However, these efforts face significant challenges due to the limited interpretability and detail of the metrics used, which obstruct a thorough understanding of the complex issues presented by hallucinatory behavior in VL models. We contend that these research gaps in VL hallucinations underscore the importance of an explainable evaluation approach that not only deciphers the mechanisms behind hallucinations but also facilitates the development of effective mitigation strategies. Additionally, we note parallel efforts in recent VL evaluation studies [218] though they do not explicitly use the term "hallucination."

In this chapter, we lay the groundwork for an explainable evaluation framework for VL hallucinations by applying our methods to image captioning, a task fraught with hallucination challenges as shown in Figure 9.1.1. We adapt techniques from prior research in VL hallucination evaluation, particularly focusing on image generation from language [218], and demonstrate their seamless application in the converse task of generating language from images. While existing research primarily addresses object hallucination, we extend our evaluation to include *interconnections between objects*, such as spatial relationships or actions. Our proposed framework retains the core attributes of conceptual counterfactuals and knowledge-driven edits [84], which we will explore in detail later in this chapter.

This chapter makes the following contributions:

- We propose the adoption of an explainable evaluation framework for image captioning hallucinations.
- We analyze the concepts present in captions to enhance the granularity of our hallucination evaluations.
- We introduce "role hallucinations" as a novel extension to the existing studies on object hallucinations.
- We substantiate our findings by applying our proposed framework to a variety of image captioning models.

Image Captioning

Image captioning has become a cornerstone in machine learning, aiming to generate descriptive textual interpretations of visual content. This task serves as a bridge between computer vision and natural language processing, facilitating seamless interaction between visual and linguistic data. In practical applications, image captioning is instrumental for assisting visually impaired individuals through descriptive narrations, enhancing image retrieval systems via textual queries, and improving human-computer interactions by aligning images with language.

The emergence of Vision-Language (VL) transformers has significantly accelerated progress in this domain. Cutting-edge models like BLIP [181], BLIP-2 [180], LLaVA [204, 203], BEiT [365], and GiT [359] have achieved remarkable results, often scaling up to billions of parameters. While increased model size generally enhances generation quality, these models are not exempt from generating "hallucinations"—inaccurate or non-existent details in captions—which pose significant obstacles for real-world deployment [93].

Hallucinations in VL models

Hallucinations in VL models refer to instances where the generated text includes elements that do not correspond to the visual input. Traditional evaluation metrics like BLEU [271], ROUGE [196], and CIDEr [351] focus on linguistic quality but often overlook the alignment between the text and the image content. Consequently, there's a growing emphasis on developing metrics that specifically address the fidelity of the generated captions to the visual input.

An early effort in this direction is the CHAIR (Caption Hallucination Assessment with Image Relevance) metric [298], which quantifies the proportion of hallucinated objects in captions:

$$\text{CHAIR}_i = \frac{|\text{Hallucinated Objects}|}{|\text{All Predicted Objects}|} \tag{9.1.1}$$

$$\text{CHAIR}_s = \frac{|\text{Sentences with Hallucinated Objects}|}{|\text{All Sentences}|} \tag{9.1.2}$$

While CHAIR provides a baseline assessment, more nuanced approaches have been developed. FAITHSCORE [136] offers a fine-grained analysis by decomposing captions into subcomponents to extract atomic facts, though it relies on Large Language Models (LLMs) that may themselves introduce hallucinations.

Dialogue-based evaluation methods like POPE [185] propose generating yes/no questions about object presence in images, using ground truth annotations to formulate queries. An equal number of questions about non-existent objects help gauge the model's susceptibility to affirmation bias. Similarly, NOPE [214] employs a question-answering framework using negative indefinite pronouns to detect hallucinations.

In an approach, [361] identified patterns in VL hallucinations and utilized LLMs to generate hallucinated examples. They fine-tuned models like LLaMA [345] on these examples to enhance hallucination detection capabilities.

Studies have observed that optimizing for traditional text generation metrics might inversely correlate with reducing hallucinations, indicating that high linguistic quality doesn't guarantee visual-textual alignment [52]. Additionally, factors such as image encoding methods, training objectives, and statistical patterns like object co-occurrence frequencies and their positional context within captions influence hallucination rates [421].

To tackle these challenges, ongoing research is exploring improved model architectures that better integrate visual and textual modalities, developing more robust training objectives that penalize misalignment, and creating comprehensive evaluation metrics that balance linguistic fluency with factual accuracy. Addressing hallucinations is crucial for advancing image captioning systems that are reliable and effective in real-world scenarios, where accuracy is not just preferred but essential.

9.2 Hallucination Detection through Counterfactual Explanations

While numerous studies have leveraged LLMs to assess hallucinations in VL models, this dependency introduces inherent uncertainties. These uncertainties arise from the variability in prompt formulations and the propensity of LLMs to generate their own hallucinations, which can undermine the robustness and reliability of the evaluation frameworks. To circumvent these issues, our proposed methodology deliberately eschews the use of LLMs. By sacrificing the convenience they offer, we enhance the determinism and trustworthiness of the hallucination evaluation process.

Furthermore, existing metrics that evaluate linguistic quality or detect VL hallucinations often lack explainability. They fail to provide actionable guidance on how to modify the generated content to eliminate hallucinations. An effective evaluation framework should not only identify discrepancies but also suggest a *direction of change* that is both **measurable** and **meaningful**. Optimally, this change should involve making the ***smallest possible adjustments*** with the ***fewest necessary edits*** to achieve the desired outcome. Below, we elaborate on these key criteria:

Measurable This involves assigning precise numerical values to changes, facilitating comparison and quantification. It requires connecting concepts slated for modification with similarity metrics within a unified structure, such as their distances in a semantic space or positions within an ontology.

Meaningful Adjustments should be sensible within the real-world context and adhere to linguistic norms. For example, replacing the concept "cat" with "dog" is meaningful because both are animals, whereas substituting "cat" with a random string like "hfushbfb" or an unrelated action like "swimming" lacks semantic validity and violates syntactic rules.

Optimal This pertains to implementing a strategy that ensures the selected changes are the best among all valid and measurable options. For instance, replacing "cat" with "tiger" might be more semantically appropriate than substituting it with "person," given the closer taxonomical relationship between felines. Optimal edits aim for the most ***semantically minimal changes***, involving the least deviation from the original concept. Additionally, the total number of edits should be minimized to avoid unnecessary complexity, resulting in the ***fewest possible semantically minimal edits***.

Implementing the Framework with WordNet

To address these challenges, we build upon the framework introduced by [84], which provides counterfactual explanations through edits that meet our specified criteria. This approach was subsequently adapted for evaluating image generation models in [219]. In our framework, we define a source set S containing concepts extracted from the generated captions and a target set T comprising ground truth concepts derived from annotated images.

Our objective is to transform S into T using the minimal number of meaningful edits, achieved through the structural guarantees provided by WordNet [243]. WordNet organizes English words into synsets—sets of cognitive synonyms—arranged in a hierarchical structure based on semantic relationships. By mapping concepts from S and T onto WordNet synsets, we can quantify semantic differences through the distances between synsets. The shortest path between two synsets corresponds to the minimal semantic change needed to align the concepts. This methodology ensures that edits are **measurable** (using numerical distances), **meaningful** (grounded in valid linguistic entities), and **semantically minimal** (identified via efficient pathfinding algorithms like Dijkstra's algorithm [64]).

The algorithm proposed by [84] employs bipartite matching to optimize the assignment of concepts from S to T, minimizing the total semantic cost and ensuring the optimal transformation from the generated captions to the ground truth.

Edit Operations for Optimal Transformation

The transformation from S to T involves three types of edit operations for any source concept $s \in S$ and target concept $t \in T$ [84, 219]:

- **Replacement (R)** $e_{s \to t}(S)$: Replace a concept s in S with a concept t not originally in S.
- **Deletion (D)** $e_{s-}(S)$: Remove a concept s from S.
- **Insertion (I)** $e_{t+}(S)$: Add a concept t from T to S.

In the context of image captioning, we prioritize **Deletion** and **Replacement** operations. Hallucinations often involve the inclusion of irrelevant or non-existent concepts, so removing or substituting these elements is crucial for aligning captions with the visual content. While **Insertion** can enhance captions by adding missing concepts, it may not always be desirable, especially when captions are intended to be concise or provide higher-level summaries. Therefore, we calculate **Insertion** operations for completeness but exclude them from the overall transformation cost, allowing users to decide whether to incorporate them.

Our deterministic approach not only enhances the evaluation of hallucinations but also contributes to the broader goals of explainable artificial intelligence. By providing clear, quantifiable, and meaningful directions for correcting hallucinations, we enhance the transparency of VL models. This is particularly important for applications where trust and accountability are paramount, such as assistive technologies for visually impaired individuals or systems used in medical imaging.

Moreover, by minimizing reliance on LLMs, we reduce the black-box nature of the evaluation process. Our framework offers interpretable results that can be scrutinized and validated, fostering greater confidence in the deployment of VL models in real-world scenarios.

9.2.1 The role of roles

Traditional approaches to hallucination detection in image captioning have predominantly concentrated on object-level inaccuracies, often neglecting the critical role of relationships between objects, known as *role hallucinations*. For instance, as depicted in Figure 9.1.1, the BLIP captioning model misinterprets the spatial relationship between a man and a dog, confusing their positions. This example underscores the necessity of addressing role hallucinations, which have been relatively overlooked in prior research focused mainly on object hallucinations.

It is insufficient to analyze roles in isolation; they must be considered *in conjunction with objects* to accurately detect hallucinations. Evaluating roles separately can lead to under-detection of errors because it overlooks the context provided by the objects involved. For example, applying the counterfactual explanation algorithm from [84] solely to sets of roles might suggest a simple insertion operation, such as **I**("next to"), indicating the addition of the role "next to" to connect "dog" and "man." However, this approach may not fully capture the misrepresentation.

By instead considering *triplets*—pairs of objects connected by a role—we obtain a more accurate set of edits. In the context of Figure 9.1.1, the proposed edits become **R**(["dog", "on", "lap"], ["laptop", "on", "lap"]), **I**(["dog", "next to", "man"]). This means replacing the incorrect triplet where the "dog" is "on" the "lap" with the correct one where the "laptop" is "on" the "lap," and inserting the missing triplet where the "dog" is "next to" the "man." This more comprehensive set of edits aligns better with both the human-written ground truth caption and the actual content of the image.

To facilitate editing at the triplet level, we employ **scene graphs** to represent both the image and the caption. Scene graphs are structured representations where nodes correspond to objects and edges represent the relationships (roles) between them. This graph-based approach provides a detailed semantic depiction of the visual scene and the generated caption, enabling a direct comparison between the two.

Parsing the caption into a graph structure involves natural language processing techniques such as dependency parsing and semantic role labeling. This process extracts objects and their relationships from the text, constructing a graph G_S that mirrors the semantic content of the caption. Similarly, the image is analyzed to produce a scene graph G_T, utilizing object detection and relationship recognition algorithms.

With the two graphs G_S (caption) and G_T (image) established, our goal is to find the minimal cost sequence of edit operations that transforms G_S into G_T. The allowable edit operations include:

- **Replacement (R)**: Substituting an incorrect triplet in G_S with the correct one from G_T.
- **Deletion (D)**: Removing an extraneous triplet from G_S that does not correspond to any in G_T.
- **Insertion (I)**: Adding a missing triplet from G_T into G_S.

The cost associated with each edit operation is denoted as $c(e_i)$. To quantify the total cost of transforming G_S into G_T, we use the **Graph Edit Distance (GED)**, defined as:

$$\mathrm{GED}(G_S, G_T) = \min_{(e_1,\ldots,e_n)\in P(G_S,G_T)} \sum_{i=1}^{n} c(e_i) \tag{9.2.1}$$

Here, $P(G_S, G_T)$ represents all possible sequences of edit operations that convert G_S into G_T. The GED reflects the minimal total cost of edits needed for this transformation, effectively measuring the dissimilarity between the two graphs.

To compute the optimal sequence of edits, we employ deterministic pathfinding algorithms such as Dijkstra's algorithm [64]. These algorithms ensure that the edit path found is the one with the minimal total cost, guaranteeing the optimality of the proposed edits.

Calculating the exact GED is known to be an NP-hard problem, which makes it computationally infeasible for graphs of substantial size or complexity due to the exponential growth of possible edit sequences. To overcome this challenge, we utilize approximation algorithms that provide efficient and scalable solutions.

One such algorithm is the Volgenant-Jonker (VJ) algorithm [140], which is designed to solve the linear assignment problem in polynomial time. By framing the GED calculation as an assignment problem, the VJ algorithm approximates the minimal edit cost without exhaustively exploring all possible edit sequences. This approach significantly reduces computational overhead while still providing a close approximation to the optimal GED.

9.3 Hallucination detection framework

Object hallucinations As depicted in Figure 9.3.1, our framework provides a method for analyzing hallucinations within generated content.

We define the problem of detecting object hallucinations as follows: for each caption c, the system generates a set of objects $S = \{s_1, s_2, \ldots, s_n\}$, whereas the corresponding image comprises a set of actual objects $T = \{t_1, t_2, \ldots, t_m\}$. The transition from S to T involves making specific conceptual adjustments, notably through the operations **R**, **D**, and **I**, which are detailed in the prior section.

To assess the scope of hallucinations, from general to specific deviations from the truth, we engage the Least Common Ancestor (LCA) concept within the WordNet structure. Here, the LCA refers to the most immediate common ancestor within the WordNet synsets, enabling us to identify whether one synset is more generic than another. For instance, if the LCA of synsets v and w is v, then v represents a broader category than w.

This framework allows us to classify *hallucination instances* as follows:

- **Deletion (D)**: This type of error occurs when an object present in S does not appear in T; for example, "soda" is mentioned in the caption c but is absent from the actual image, as shown in Figure 9.3.1.
- **Replacement (R)**: Occurs when an object $s_i \in S$ is erroneously substituted with t_j in T, where neither $LCA(s_i, t_j) = s_i$ nor $LCA(s_i, t_j) = t_j$. This means neither object serves as a direct hypernym of the other. An example is the mention of a "chair" in the caption, whereas the image shows a "sofa".
- **Over Specialization (O)**: This error type emerges when s_i from S is replaced with a more general term t_j from T, where $LCA(s_i, t_j) = t_j$. This indicates a generalization error in the recognition process, such as labelling a depicted "woman" as a "girl".

Utilizing these categories, the level of hallucination in a caption c is quantified by the sum of affected objects across these phenomena. Thus, the hallucination count, *Hallucinations*(S, T), is determined by:

$$\mathit{Hallucinations}(S, T) = |\mathbf{D}(S, T)| + |\mathbf{R}(S, T)| + |\mathbf{O}(S, T)| \tag{9.3.1}$$

The rate of hallucination, *HalRate*, then measures the proportion of hallucinated objects against the total number of objects $|S|$ mentioned in c, calculated as:

$$\mathit{HalRate}(S, T) = \frac{\mathit{Hallucinations}(S, T)}{|S|} \tag{9.3.2}$$

Further, we introduce semantic metrics for deeper analysis, including a measure of **Similarity of Replacements**:

- **Similarity of Replacements**: This metric employs Wu-Palmer similarity [376] to assess how semantically close the replaced objects are, reflecting on the justified nature of the replacements made by the caption generator. Higher Wu-Palmer scores suggest a semantically closer and potentially more justifiable replacement.

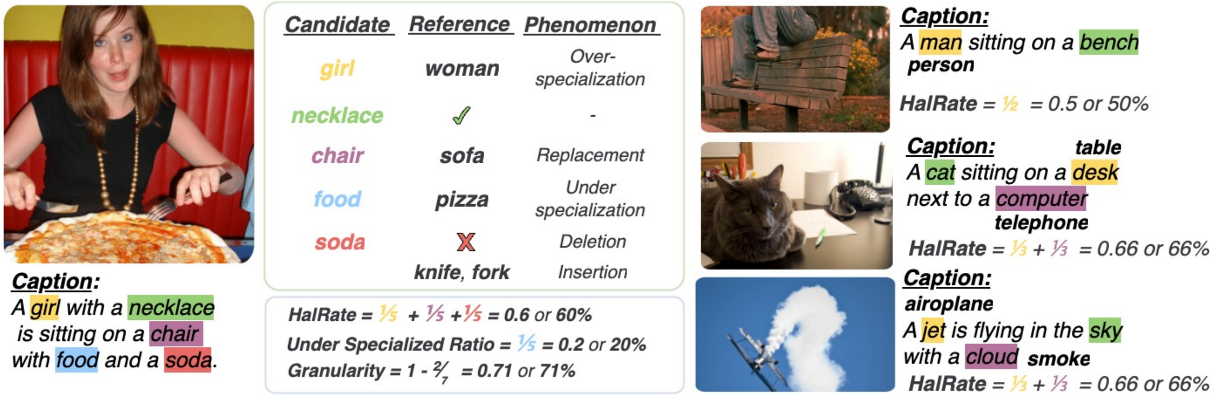

Caption:
A girl with a necklace is sitting on a chair with food and a soda.
Candidate
Reference
Phenomenon
girl
woman
Over-specialization
necklace
✓
-
chair
sofa
Replacement
food
pizza
Under specialization
soda
✗
Deletion
knife, fork
Insertion
HalRate = ⅕ + ⅕ +⅕ = 0.6 or 60%
Under Specialized Ratio = ⅕ = 0.2 or 20%
Granularity = 1 - 2⁄7 = 0.71 or 71%
Caption:
A man sitting on a bench
person
HalRate = ½ = 0.5 or 50%
Caption:
table
A cat sitting on a desk next to a computer
telephone
HalRate = ⅓ + ⅓ = 0.66 or 66%
Caption:
airoplane
A jet is flying in the sky with a cloud smoke
HalRate = ⅓ + ⅓ = 0.66 or 66%

Figure 9.3.1: An example of detected hallucination of objects in image captioning from our framework is presented, depicting each phenomenon along with the proposed metrics. Objects in yellow represent an overspecialized phenomenon, in purple a replacement, and in red a removal. Those in green are correct objects, and those in blue are the underspecialized objects (which do not constitute hallucinations, as the caption contains a more generic concept to the ground truth one). As shown, the hallucination rate is calculated as the sume of the rate of each hallucination phenomenon independently.

9.4 Extending Beyond Hallucination Detection

Exploring Additional Phenomena Our framework not only detects hallucinations but also explores a variety of related phenomena. This extension is demonstrated by introducing new metrics as follows:

- **Granularity**: This metric is defined as the complement of the ratio of **Insertions (I)** relative to the number of actual objects in the image. It is computed as:

$$Granularity(S, T) = 1 - \frac{|\mathbf{I}(S)|}{|T|} \tag{9.4.1}$$

  Essentially, it gauges the extent to which the generated caption c manages to encompass the objects depicted in the image, offering a quantifiable measure of coverage and specificity.

- **Under-Specialization (U)**: This measure evaluates cases where the object described in the caption c is more general than the one in the image. For instance, if the caption refers to "food" while the image shows "pizza," it calculates how often the captioning system opts for broader categories when more specific terms could provide more detail. The calculation is the number of such under-specialized objects divided by the total objects in the caption, thus:

$$UnderSpecialization(S, T) = \frac{\textit{Number of Under-Specialized Instances}}{|S|} \tag{9.4.2}$$

To enrich our analysis, we also track the **average number of objects per caption** and the **average number of WordNet ancestors (hypernyms)** for each object. This dual approach allows a nuanced understanding of the content's depth and scope in the captions.

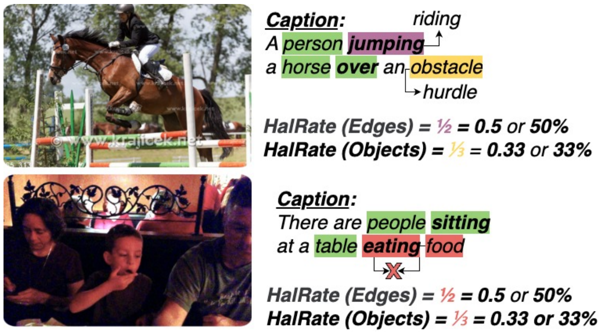

Caption:
riding
A person jumping
a horse over an obstacle
hurdle
HalRate (Edges) = ½ = 0.5 or 50%
HalRate (Objects) = ⅓ = 0.33 or 33%
Caption:
There are people sitting
at a table eating food
X
HalRate (Edges) = ½ = 0.5 or 50%
HalRate (Objects) = ⅓ = 0.33 or 33%

Figure 9.4.1: Visual depiction of role integration within our hallucination assessment framework, with edges emphasized in bold and color-coded to match Figure 9.3.1.

Detecting Role Hallucinations The framework further extends to detecting hallucinations at the level of object interactions or roles, which are critical in the context of image captions and annotations. These interactions are encoded as triples in our dataset, denoted for captions and images as $S^r = \{(s_i, r^s_j, s_k), \ldots\}$ and $T^r = \{(t_i, r^t_j, t_k), \ldots\}$, respectively. Figure 9.4.1 illustrates these roles graphically. For the quantification of role hallucinations during the $S^r \rightarrow T^r$ transition, we adapt our previous measures for handling object-based edits to consider relational triples, such as:

- **Deletions (D)**: This involves removing an edge r^s_j between two objects s_i and s_k due to inaccuracies like object deletion or incorrect relations, as shown in Figure 9.4.1. An example is the deletion of the "eating" relation between "people" and "food" when the food item is not accurately captured by the caption.
- **Replacement (R)**: Occurs when a relation r^s_j between two objects s_i and s_k is mistakenly established and needs to be corrected to match the image, as in replacing "jumping" with "riding" in Figure 9.4.1. This metric stresses the importance of accurately portraying the dynamic relationships within the scene.

It's important to recognize that the concepts of over-specialization and under-specialization do not apply to roles within this context, as the relationships described by the edges focus on actions, topological connections, or compositional relationships rather than hierarchical structures. To address this, we utilize human-provided annotation data to accurately align the relationships depicted in captions with the actual ground truth, subsequently categorizing them into appropriate WordNet synsets. In instances where captioners introduce relationships that are not present in the established ground truth, we assign appropriate weights to these relations, facilitating their seamless integration or removal during the Graph Edit Distance (GED) calculation process. It is unlikely that these relations will be substituted with alternatives due to the absence of corresponding semantic content. To ascertain their inclusion in the set $\mathbf{R}$, an additional post-analysis reasoning step is necessary to determine whether a relationship r^s_j has been removed and a new one r^t_w established between the same entities. Based on the aforementioned considerations, the phenomenon of role hallucinations is evaluated as follows:

$$Hallucinations(S^r, T^r) = |\mathbf{D}(S^r, T^r)| + |\mathbf{R}(S^r, T^r)| \tag{9.4.3}$$

while *HalRate* and *Granularity* are simply adjusted to be:

$$HalRate(S^r, T^r) = \frac{Hallucinations(S^r, T^r)}{|S^r|} \tag{9.4.4}$$

$$Granularity(S^r, T^r) = 1 - \frac{|\mathbf{I}(S^r)|}{|T^r|} \tag{9.4.5}$$

In our study, we analyze the integration of images with corresponding captions and scene graphs by utilizing datasets from both Visual Genome (VG) and Microsoft COCO. VG is notable for its detailed scene graph annotations that include objects, attributes, and relationships, while COCO provides five human-annotated captions per image. Our focus is on the COCO validation set, specifically selected to align with VG, featuring

2170 overlapping instances. We exclude instances where object alignments with WordNet synsets are not feasible.

For our experiments, we employ non-commercial captioning tools, assessing both compact and extensive model configurations. This approach ensures that smaller captioning systems, which are more accessible for widespread research use, are included. Specifically, we test various models of BLIP (with base and large configurations using ViT encoders) and GiT (including base and large models trained on 10 million and 20 million image-text pairs, respectively, with an additional variant fine-tuned on COCO captions). Our experimentation also extends to both conditional and unconditional image captioning techniques, where models are fine-tuned to predict specific or general caption distributions. All captioning models are sourced from Huggingface, with no additional training conducted post-loading.

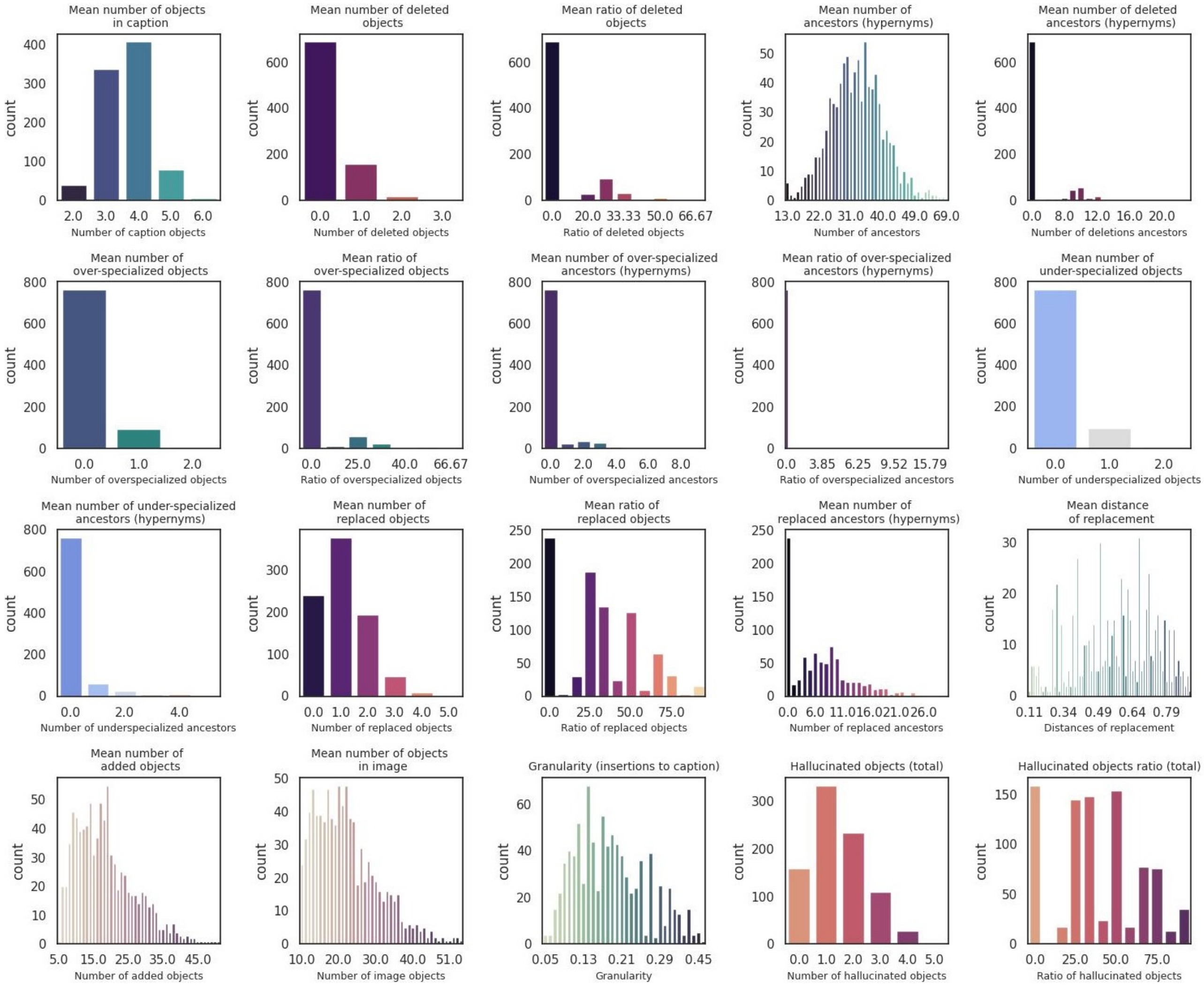


Figure 9.4.2: Analysis of object hallucination metrics using BLIP-large-unc on the Visual Genome and COCO validation dataset intersection.

Concept sets construction We develop the concept sets for both linguistic and visual domains, denoted as S, S^r for source and T, T^r for target, with the objective of transforming S into T and S^r into T^r. The linguistic concept sets are derived using the Scene Graph Parser tool, which extracts structured graphs from textual content. Conversely, the visual concept sets are crafted from the authentic annotations available in the COCO and VG datasets.

Model	#objects	#ancestors	HalRate (#hal. objects) ↓	Granularity	**U** ↓
GiT-base-coco	3.13	27.93	35.56% (1.13)	17.0%	4.06% (0.13)
GiT-large-coco	3.15	27.97	33.93% (1.1)	17.0%	3.92% (0.12)
GiT-base	1.76	16.57	26.41% (0.48)	9.0%	3.27% (0.06)
GiT-large	1.74	16.28	25.38% (0.46)	9.0%	3.31% (0.06)
BLIP-base-unc	2.53	22.55	34.28% (0.91)	13.0%	4.48% (0.12)
BLIP-base-cond	3.23	29.5	58.48% (1.87)	17.0%	2.96% (0.1)
BLIP-large-unc	3.63	32.73	39.2% (1.45)	19.0%	3.47% (0.13)
BLIP-large-cond	4.22	37.5	53.04% (2.24)	22.0%	2.84% (0.12)
BLIP2-flan-t5-xl	2.57	23.16	33.13% (0.89)	14%	4.05% (0.11)
BLIP2-opt-2	2.78	24.89	33.28% (0.96)	15.0%	4.19% (0.12)
ViT-GPT2	2.95	26.51	38.76% (1.18)	16.0%	4.47% (0.14)
Claude sonnet-L	6.85	58.94	58.91% (4.05)	36.0%	4.71% (0.33)
Claude haiku-L	7.12	58.66	64.31% (4.64)	39.0%	5.4% (0.39)
Claude sonnet-S	3.35	30.48	47.16% (1.6)	17.0%	4.67% (0.16)
Claude haiku-S	2.95	25.49	54.36% (1.62)	16.0%	6.74% (0.19)

Table 9.1: Object hallucinations (mean values) on the $VG \cap COCO$ validation subset. Best and worst results are denoted. Numbers in parenthesis denote absolute #objects.

Model	**D** ↓	**O** ↓	**R** ↓	Similarity of Replacements ↑
GiT-base-coco	4.38% (0.15)	3.01% (0.09)	28.18% (0.89)	0.56
GiT-large-coco	4.4% (0.16)	2.46% (0.08)	27.06% (0.87)	0.55
GiT-base	2.11% (0.05)	2.17% (0.04)	22.12% (0.4)	0.61
GiT-large	2.46% (0.05)	2.41% (0.04)	20.51% (0.36)	0.6
BLIP-base-unc	3.78% (0.11)	2.65% (0.07)	27.86% (0.73)	0.57
BLIP-base-cond	23.07% (0.72)	2.76% (0.09)	32.66% (1.05)	0.52
BLIP-large-unc	6.13% (0.24)	3.48% (0.13)	29.59% (1.08)	0.56
BLIP-large-cond	19.27% (0.81)	2.46% (0.11)	31.3% (1.32)	0.52
BLIP2-flan-t5-xl	4.27% (0.12)	3.16% (0.08)	25.7% (0.69)	0.56
BLIP2-opt-2	3.64% (0.11)	2.8% (0.08)	26.84% (0.77)	0.57
ViT-GPT2	3.45% (0.11)	3.16% (0.09)	32.14% (0.97)	0.6
Claude sonnet-L	15.79% (1.05)	2.51% (0.19)	40.61% (2.81)	0.52
Claude haiku-L	17.3% (1.28)	2.69% (0.2)	44.33% (3.15)	0.49
Claude sonnet-S	7.1% (0.25)	5.42% (0.18)	34.63% (1.16)	0.57
Claude haiku-S	7.78% (0.24)	4.59% (0.13)	41.99% (1.26)	0.52

Table 9.2: Continuation of Tab. 9.1. More object hallucination phenomena on $VG \cap COCO$ validation subset.

9.5 Experiments

In the subsequent sections, we detail our experiments that utilize these constructed sets. We analyze the prevalence of hallucinations in captioning, where tables and figures such as *Tables 9.1, 9.2, and 9.3, and Figures 9.4.2 and 9.5.1* present summarized results across different captioners, particularly focusing on the instances of hallucinations concerning objects and roles. Our findings indicate a significant rate of hallucinations: approximately one-third of captioned objects exhibit some discrepancies, and over half of the role annotations deviate from their verified counterparts.

Our evaluation further distinguishes between types of edits involved in caption modifications: Replacements (**R**) are more common than Deletions (**D**), as the conceptual paths in captions are typically shorter and more direct compared to the broader hierarchical structure used in object identification. This observation is consistent with the training regimes of these models, which are pre-trained on richly descriptive datasets like COCO.

Furthermore, we cross-reference our hallucination findings with traditional language generation metrics, including BLEU, ROUGE, Google BLEU, Mauve, and perplexity. The results presented show that higher language generation scores do not necessarily correlate with lower rates of hallucinations, indicating that these metrics alone may not provide a complete assessment of a model's performance in accurately reflecting depicted scenarios without introducing hallucinations. This insight is particularly emphasized by the performance patterns of the GiT models, which, while scoring lower on language generation metrics, demonstrate fewer hallucinations, thus highlighting a potential trade-off between descriptive richness and accuracy.

Model	#roles	**D** ↓	**R** ↓	HalRate (#hal. roles)↓	Granularity
GiT-base-coco	1.92	65.32% (1.37)	14.06% (0.29)	79.38% (1.66)	3.93%
GiT-large-coco	1.94	65.33% (1.36)	13.75% (0.29)	79.09% (1.65)	4.08%
GiT-base	0.73	44.05% (0.47)	11.98% (0.13)	56.03% (0.59)	1.8%
GiT-large	0.69	39.15% (0.42)	11.58% (0.12)	50.63% (0.54)	1.89%
BLIP-base-unc	1.44	61.2% (1.01)	13.04% (0.2)	74.23% (1.22)	3.01%
BLIP-base-cond	2.14	90.96% (1.93)	4.22% (0.1)	95.18 (2.03)	1.48%
BLIP-large-unc	2.28	68.32% (1.67)	13.2% (0.31)	81.52% (1.98)	4.38%
BLIP-large-cond	2.98	86.6% (2.54)	6.68% (0.22)	93.28% (2.77)	2.99%
BLIP2-flan-t5-xl	1.62	69.26% (1.16)	14.2% (0.22)	83.47% (1.38)	3.25%
BLIP2-opt-2	1.79	68.87% (1.25)	14.37% (0.25)	83.24% (1.51)	3.65%
ViT-GPT2	1.86	71.05% (1.36)	16.46% (0.28)	87.5% (1.64)	3.42%
Claude sonnet-L	3.9	80.71% (3.17)	9.8% (0.39)	90.51% (3.56)	7.1%
Claude haiku-L	3.99	80.25% (3.29)	10.31% (0.38)	90.56% (3.67)	6.28%
Claude sonnet-S	2.1	75.19% (1.62)	11.85% (0.25)	87.04% (1.87)	5.24 %
Claude haiku-S	1.85	74.31% (1.39)	13.71% (0.14)	88.02% (1.53)	4.99%

Table 9.3: Role hallucinations (mean values per image) on the $VG \cap COCO$ validation subset.

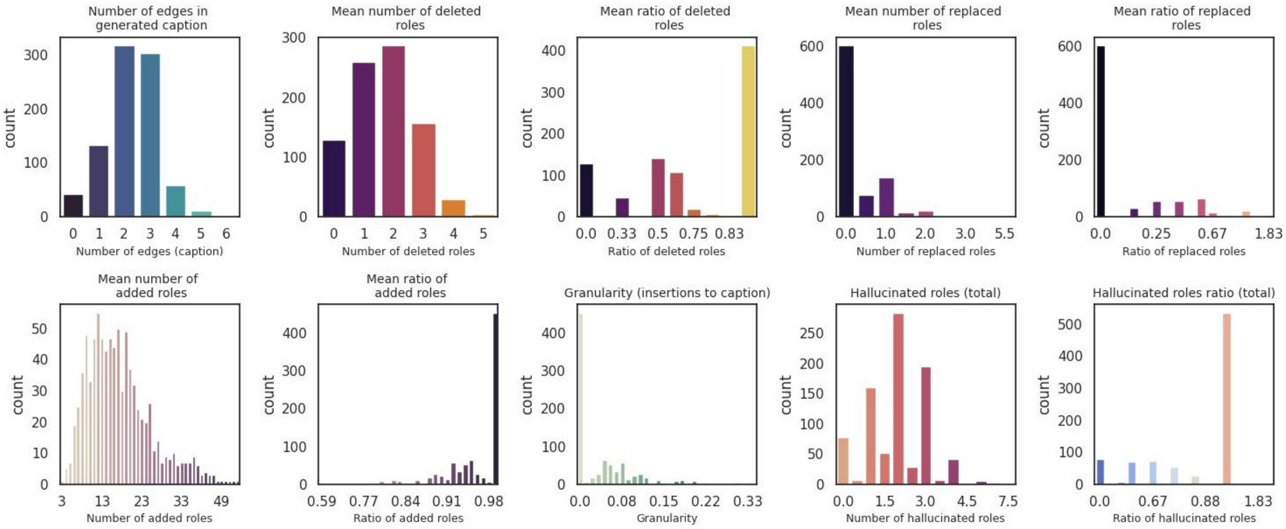


Figure 9.5.1: Statistics of our proposed metrics on role hallucinations by BLIP-large-unc on the $VG \cap COCO$ validation set.

9.6 Conclusion

In summary, the framework we introduced for detecting hallucinations in image captioning marks a significant advancement in the explainable assessment of evolving Vision-Language (VL) models. This framework explores the underlying mechanisms of hallucination by dissecting these phenomena through the lens of conceptual properties, enriched by external hierarchical knowledge. Our approach strategically applies semantically minimal yet meaningful edits to transition from hallucinated concepts in captions to their accurate, non-hallucinated counterparts, utilizing the explanatory potential of conceptual counterfactual methodologies.

Models	**ROUGE1↑**	**ROUGE2↑**	**ROUGEL↑**	**ROUGELsum↑**
GiT-base-coco	0.152	0.021	0.145	0.145
GiT-large-coco	0.152	0.022	0.146	0.146
GiT-base	0.139	0.01	0.134	0.134
GiT-large	0.127	0.01	0.122	0.122
BLIP-base-unc	0.16	0.021	0.153	0.154
BLIP-base-cond	0.352	0.116	0.317	0.317
BLIP-large-unc	0.134	0.017	0.126	0.126
BLIP-large-cond	0.402	0.163	0.361	0.361
BLIP2-flan-t5-xl	0.435	0.179	0.402	0.402
BLIP2-opt-2	0.44	0.187	0.404	0.404
ViT-GPT2	0.406	0.153	0.370	0.370
Claude sonnet-L	0.133	0.008	0.117	0.117
Claude haiku-L	0.141	0.011	0.125	0.125
Claude sonnet-S	0.062	0.002	0.058	0.058
Claude haiku-S	0.123	0.009	0.114	0.114
	BLEU ↑	**Google BLEU↑**	**Mauve↑**	**PPL↓**
GiT-base-coco	0.0005	0.051	0.186	68.305
GiT-large-coco	0.0005	0.051	0.192	63.629
GiT-base	0.0001	0.027	0.131	1541.317
GiT-large	0.0001	0.025	0.13	1475.033
BLIP-base-unc	0.0004	0.037	0.141	461.076
BLIP-base-cond	0.024	0.099	0.058	506.732
BLIP-large-unc	0.0003	0.033	0.132	67.632
BLIP-large-cond	0.056	0.133	0.064	127.578
BLIP2-flan-t5-xl	0.046	0.132	0.067	211.738
BLIP2-opt-2	0.055	0.139	0.009	130.29
ViT-GPT2	0.051	0.131	0.068	69.605
Claude sonnet-L	0.0001	0.029	0.174	71.307
Claude haiku-L	0.0002	0.029	0.174	42.032
Claude sonnet-S	0.0	0.032	0.174	358.33
Claude haiku-S	0.0004	0.047	0.174	170.585

Table 9.4: Language generation evaluation metrics on the $VG \cap COCO$ validation subset.

Additionally, our research highlights the often-neglected issue of role hallucinations, showing that popular image captioning models frequently generate incorrect relationships between objects. We consider our work a vital initial step towards accurately detecting hallucinations in VL models, laying a foundation for future strategies aimed at mitigating such errors. Looking ahead, we aim to expand our examination by incorporating additional semantic resources into the framework, and also include a broader range of captioners and larger model architectures. Furthermore, we plan to extend our hallucination detection framework to additional VL tasks, thereby enhancing its robustness and applicability.

Expanding the Framework with Additional Semantic Resources

While WordNet serves as a foundational tool for our framework, incorporating additional semantic resources can further enhance the evaluation process. For example, leveraging ConceptNet [327] can provide richer semantic relationships, including commonsense knowledge that extends beyond lexical definitions. Additionally, utilizing word embeddings from models like Word2Vec [242], GloVe [275], or contextual embeddings from transformers like BERT [62] can capture nuanced semantic similarities based on large-scale language corpora.

These resources enable us to measure semantic distances more precisely, especially in cases where WordNet's hierarchical structure may not fully represent the complexity of concept relationships. By integrating multiple semantic models, we can improve the accuracy and reliability of the measurable changes identified during the evaluation.

Chapter 10

Counterfactual Generation for Improving Reasoning Abilities of LLMs

10.1 Introduction

Recent advancements in the domain of Large Language Models (LLMs) such as GPT-3 [26] and GPT-4 [262] have been well-documented, illustrating their robust capabilities in logical reasoning across a variety of domains [201, 202, 16, 48, 341]. These advancements highlight the strides made in enhancing the deductive reasoning capabilities of these models. However, despite such progress, limitations persist in scenarios involving inductive reasoning, as detailed in recent studies [383, 15, 332].

In this research, the adopted classification system for reasoning, proposed in [94], emphasizes the underlying cognitive processes and essential skills necessary for effective puzzle-solving. This taxonomy shifts focus away from the simplistic categorization based on question formats [217] or the nature of reasoning—whether it be deductive, inductive, or abductive [217, 394, 390, 281, 122, 85]. For instance, puzzles that are rule-based, such as Sudoku, Crosswords, or Minesweeper, necessitate not only an understanding of the specific game rules but also the development of sophisticated strategies to engage these rules effectively or to format the outputs appropriately. Conversely, puzzles devoid of strict rules, programming challenges, and tasks that require commonsense reasoning, depend primarily on the model's built-in knowledge base for deriving solutions.

The assessment process for LLMs' reasoning capabilities involves a structured categorization of puzzles. As delineated in Figure 10.1.1, puzzles are distinguished by their reliance on either strict formal rules or a broader utilization of worldly knowledge coupled with general inferential skills. This categorization serves to not only reveal the cognitive diversity inherent in different types of puzzles but also aligns with the distinct reasoning challenges presented by each category. Puzzles governed by rules demand precise logical deduction and strategic foresight, operating within environments that are tightly controlled and where parameters are clearly defined. In contrast, puzzles that lack formal rules call upon the model's general reasoning abilities, which include interpreting complex scenarios and elucidating events through the derivation of inferences rooted in practical, everyday knowledge. Additionally, the research incorporates a critical aspect of unlearning outdated methodologies and biases [236, 235], which is essential for the progressive adaptation and accuracy of LLMs in evolving cognitive tasks [278].

Through this nuanced categorization, the research aims to deliver an in-depth analysis of the problem-solving capabilities of LLMs, reflecting on the varied challenges presented by both structured tasks and those that require expansive inferential reasoning. This approach provides a broader perspective on the potential and limitations of current LLM technologies in tackling diverse cognitive tasks.

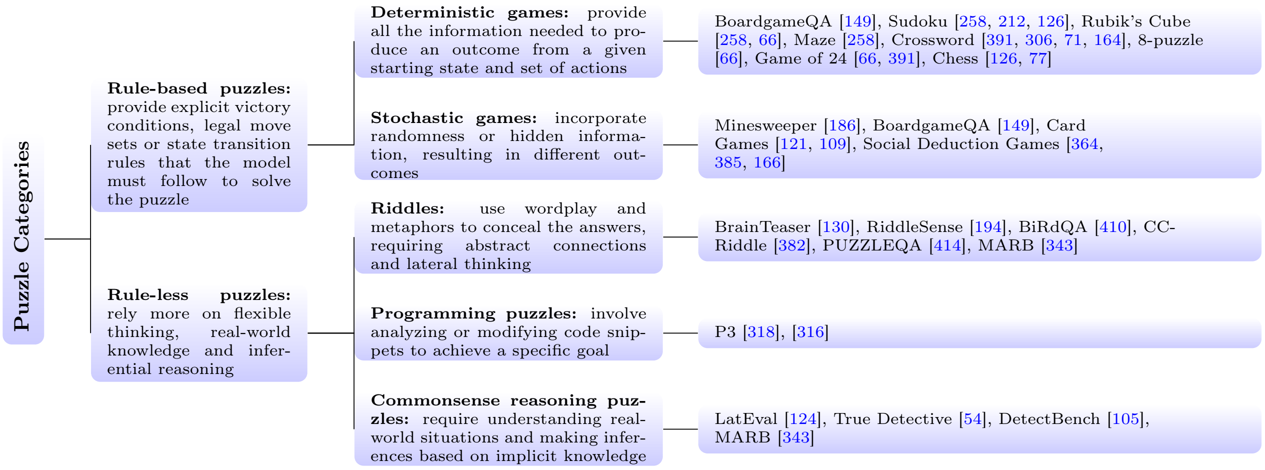

Puzzle Categories
Rule-based puzzles: provide explicit victory conditions, legal move sets or state transition rules that the model must follow to solve the puzzle
Rule-less puzzles: rely more on flexible thinking, real-world knowledge and inferential reasoning
Deterministic games: provide all the information needed to produce an outcome from a given starting state and set of actions
Stochastic games: incorporate randomness or hidden information, resulting in different outcomes
Riddles: use wordplay and metaphors to conceal the answers, requiring abstract connections and lateral thinking
Programming puzzles: involve analyzing or modifying code snippets to achieve a specific goal
Commonsense reasoning puzzles: require understanding real-world situations and making inferences based on implicit knowledge
BoardgameQA [149], Sudoku [258, 212, 126], Rubik's Cube [258, 66], Maze [258], Crossword [391, 306, 71, 164], 8-puzzle [66], Game of 24 [66, 391], Chess [126, 77]
Minesweeper [186], BoardgameQA [149], Card Games [121, 109], Social Deduction Games [364, 385, 166]
BrainTeaser [130], RiddleSense [194], BiRdQA [410], CC-Riddle [382], PUZZLEQA [414], MARB [343]
P3 [318], [316]
LatEval [124], True Detective [54], DetectBench [105], MARB [343]

Figure 10.1.1: The taxonomy of Puzzle Categories with the corresponding datasets from [94][1].

10.2 Puzzle Solving using Reasoning of LLLMs

In this section, the taxonomy proposed in [94] is analyzed, focusing on its categorizations and their potential to reveal the reasoning capabilities of LLMs [2]. This analysis aims to provide a detailed evaluation of the strengths and limitations inherent in the existing body of literature. The insights derived from this examination are expected to contribute to the development of more robust methodologies for enhancing the reasoning abilities of LLMs. By addressing the gaps and challenges identified, this work seeks to advance both theoretical understanding and practical applications in this domain.

Rule-based Puzzles

Rule-based Puzzles equip the model with explicit conditions for victory, sets of permissible moves, or state transition rules. This category is further subdivided based on the nature of the state transitions—whether they are deterministic or involve elements of randomness.

Deterministic games invariably result in the same subsequent state when a particular action is taken in accordance with the established rules. For instance, in Chess, executing a specific move consistently results in a new and definitive board configuration. Similar examples are Sudoku, navigating through a maze, or solving a Rubik's cube. Here, the model is required to develop strategies that function within the confines of the space permitted by the legal game mechanics.

Stochastic games introduce randomness or elements of concealed information, meaning the same action by a player may lead to varying probability distributions of ensuing states. Examples of such games include Minesweeper, where bomb locations are unknown, and card games like Poker, where each player's hand is kept secret. Mastery of these games demands the ability to reason about uncertain states, plan multiple steps ahead, and effectively manage risks.

Therefore, while both subcategories necessitate logical reasoning within the bounds of formal rules, stochastic games introduce the added complexity of decision-making under conditions of uncertainty. Achieving proficiency in deterministic games relies heavily on deductive reasoning and forward planning, whereas stochastic environments also call for capabilities in probabilistic thinking, risk assessment, and reasoning with incomplete information.

Rule-less Puzzles

Contrasting with rule-bound puzzles, rule-less puzzles demand more adaptable thinking and a broader base of real-world knowledge to make sense of ambiguous situations and deduce unseen details. These puzzles do not merely test systematic search or strategic foresight; instead, they evaluate cognitive abilities related to

[2] https://puzzlellms.github.io

contextual interpretation, combining concepts, and reasoning based on common life experiences. Examples of puzzles in this category include:

Riddles leverage witty wordplay and literary techniques to mask the answers. An example is the query, "What gets wetter the more it dries?" which cleverly hides the answer "a towel" using metaphorical language. Solving riddles involves making abstract connections between hidden concepts, often presented in poetic form. This evaluates the solver’s capacity for fluid reasoning, conceptual blending, and lateral thinking to unravel the linguistic nuances.

Programming Puzzles typically present snippets of code that require analysis or modifications to the existing logic. Defined by [318, 72] as a brief Python program f, the objective is to identify an input that results in f returning True. These puzzles test abilities such as tracing program execution, identifying and rectifying errors, or predicting outputs based on the semantics of the code. For instance, consider the following programming challenge, which examines understanding of programming semantics to foresee a system’s response:

```
def mystery(x):
    return x // 2
print(mystery(10))
```

Commonsense Reasoning Puzzles typically portray ordinary scenarios while deliberately omitting critical details. Solvers are expected to construct explanations for events by inferring likely unstated assumptions about motivations, causality, and consequences. For example, the question "A man who was outside in the rain without an umbrella or hat didn’t get a single hair on his head wet. Why?" challenges one to perform a pragmatic analysis of the unspoken contextual elements.

10.2.1 Methods and Strategies

In integrating LLMs into puzzle-solving contexts, a diverse range of methods and strategies significantly enhances complex reasoning and performance. This section details the various approaches employed to tackle puzzles, emphasizing their specific applications within this distinctive area. Considering the rich body of research on prompt engineering and related methodologies [20, 34, 395, 44, 281, 207], this discussion focuses on the most commonly used techniques in puzzle solving. Rather than detailing each method individually, the section categorizes the existing strategies into prompting techniques, neuro-symbolic approaches for puzzle translation, and fine-tuning targeted at specific domains. An extensive review of the methods applied across various puzzle categories is depicted in Table 10.1, providing a structured insight into their effectiveness and areas of application.

In integrating LLMs into puzzle solving, a broad range of methods and strategies has been explored to enhance complex reasoning and performance. This section provides a comprehensive overview of these approaches, emphasizing their unique application within the realm of puzzles. Recognizing the vast body of existing literature on prompt engineering and associated methodologies [20, 34, 395, 44, 281, 207], this discussion will focus on the most prevalent techniques for puzzle solving rather than detailing each method individually. These methods are categorized into prompting techniques, neuro-symbolic approaches for puzzle translation, and domain-specific fine-tuning. An extensive summary of the methods utilized across various puzzle types can be found in Table 10.1.

Prompting Methods

Prompting strategies that introduce intermediate reasoning steps play a crucial role in boosting the puzzle-solving capabilities of language models. The few-shot in-context learning paradigm, which incorporates one or more examples within prompts, has significantly enhanced performance for both rule-based and rule-less puzzles by demonstrating the reasoning process without the need for additional training [26, 68, 422].

Recent studies have investigated various ’thought structures’ that can guide LLMs towards the final solution [20].

Chain topologies, including the Chain-of-Thought (CoT) [370, 157], have been effectively applied to all kinds of puzzles, showing superiority over simpler input-output prompts. The Self-Refine method [228] has been particularly successful in deterministic, rule-based games like the Game of 24, achieving a 13% higher success rate than CoT [391]. In rule-less contexts, such as detective-style benchmarks, various approaches like Automatic CoT, which generates diverse reasoning chains autonomously [413], and Complexity CoT, which enhances performance by leveraging more complex reasoning steps [87], have been utilized. Additionally, the Plan-and-Solve (PS) method involves using two prompts per problem—one to generate the reasoning and one to extract the final answer [362]. Despite these diverse approaches, none have consistently outperformed CoT across all tested LLMs. However, the Detective Thinking Prompt method, a variant of CoT, has not surpassed the 61.6% accuracy rate achieved by the best-performing model, GPT-4. This method encourages the model to sequentially analyze multiple clues, aiding in handling complex scenarios where synthesizing diverse information is critical.

Tree topologies encompass various methods. Self-Consistency (SC) [366] has been tested on deterministic puzzles like the 8-puzzle, Game of 24, and Pocket Cube, as well as on rule-less commonsense reasoning puzzles, showing a slight advantage in the former category over CoT [66, 391, 250] and no clear benefit in the latter [105]. Tree-of-Thought (ToT) [391, 212] has so far been applied exclusively to deterministic puzzles, significantly outperforming CoT, with success rate increases ranging from 26% to 70% [250, 391], despite requiring more LLM invocations [66]. Tree-of-Uncertain-Thought (TouT) [250] has achieved even better outcomes, with a 9% higher success rate on the Game of 24 and a 3% improvement on mini-crosswords. Lastly, Inference-Exclusion-Prompting (IEP) [343] employs forward and backward reasoning to mimic human logic, achieving impressive results on riddles and commonsense puzzles when combined with CoT, scoring 82

Graph topologies include methods like Graph-of-Thought(s) (GoT) [19, 171] and Everything-of-Thought (XoT) [66], which have been used for deterministic puzzles. While GoT has shown poorer performance compared to ToT, with decreases ranging from 2% to 6% [66], XoT has been recognized as the most effective method, integrating Monte Carlo Tree Search (MCTS) with LLMs for enhanced thought generation, showing improvements from 53% to 69% compared to ToT and presenting the fewest LLM invocations among the methods tested.

Further exploration of these methods, including additional prompting strategies such as hints for riddles and commonsense puzzles, introductions, and summarizations, and an extensive analysis can be found in the work of [20].

Puzzle Translation

This subsection summarizes the neuro-symbolic techniques employed by LLMs to translate text-based puzzles from natural language into formats more conducive to solutions by external tools. These methods focus not on the LLMs' puzzle-solving abilities per se but on their capacity to encode puzzles into appropriate representations.

The primary approach is to use LLMs to generate logic rules from the puzzles' natural language descriptions and then solve them using a symbolic solver. GPT-3 and GPT-4 have been utilized to transform logic puzzles like chess, Jobs puzzle, and Sudoku into Answer Set Programming (ASP) formats by generating predicates and rules, showing significant success, with GPT-4 achieving a 92% accuracy rate on a logic puzzles dataset [247], markedly higher than the rates in few-shot and zero-shot settings with the same model [126]. Similar frameworks, such as Logic-LM [266], LINC [260], and methods by [389], have shown promising outcomes in logical reasoning tasks, though not specifically in puzzle contexts.

While these neuro-symbolic approaches have been successful in translating puzzles into logic rules, no studies have yet addressed transforming natural language puzzles into code. However, methods like Program of Thoughts (PoT) [39] and Program-Aided Language (PAL) [88] have been employed to convert reasoning into Python programs for logical and mathematical reasoning datasets, suggesting potential for application in puzzle-solving tasks.

Given the structured nature of rule-based puzzles, it is logical that these techniques are particularly suited to them, and thus far, no research has been conducted on their application to rule-less puzzles in this context.

Fine-Tuning

Fine-tuning LLMs has proven to be a powerful approach to significantly enhancing their reasoning capabilities. This technique applies not only to general logical reasoning but also to specific puzzle-solving tasks across various categories.

Logical Reasoning LoGiPT [76] exemplifies a language model that has been fine-tuned specifically to excel in logical reasoning. This model undergoes a fine-tuning process using an instruction-tuning dataset, which pairs natural language logical questions with symbolic reasoning steps, aiming to streamline the transition from natural to symbolic language and reduce common parsing errors, thus enabling it to generate direct answers. Similarly, LogiT5 [217] employs a multi-task learning framework, integrating multiple datasets to bolster its logical reasoning across diverse domains. It is specifically fine-tuned using the LOGIGLUE benchmark, a collection of logical reasoning datasets, which enhances its performance, especially in tasks with sparse data, by facilitating knowledge transfer across different logical challenges.

Rule-based Puzzles In the arena of rule-based deterministic puzzles, certain studies such as [258] report less than optimal results when fine-tuning GPT-2 on complex puzzles like Sudoku, Rubik's Cube, and Mazes, possibly due to a limited duration of fine-tuning and a scarcity of training examples. Studies concerning crossword puzzles [306, 71] present variable outcomes, with some fine-tuned models surpassing traditional non-neural approaches, while others fail to do so, underscoring the challenges cryptic crosswords pose to LLMs. Moreover, [149] have shown that fine-tuning LLMs with proofs and Chain-of-Thought (CoT) methodologies in rule-based contexts has led to some of the most effective results.

Rule-less Puzzles Focusing on rule-less puzzles, research such as that conducted by [194] indicates that models like BERT [63], RoBERTa [211], and ALBERT [167] perform more effectively when trained on datasets like RiddleSense and CommonsenseQA [338], effectively utilizing commonsense knowledge. Additionally, [410] highlight that a combination of fine-tuning on ALBERT-XXL with transfer learning from the CommonsenseQA dataset results in a notable improvement of 4% over simple fine-tuning strategies. This enhancement is also evident in areas like commonsense reasoning [54] and programming puzzles [318], demonstrating the wide-ranging applicability of fine-tuning across different categories of puzzles.

Table 10.1 presents a detailed account of the diverse methods utilized for puzzle-solving, as evidenced by the datasets compiled for this study. This table serves to map out the current scope of research on LLMs within the realm of puzzle-solving, with a specific focus on the extensive methodologies applied to rule-based deterministic puzzles and rule-less commonsense puzzles. Notably, the table also draws attention to the lack of neuro-symbolic techniques and selection inference prompting in the current methodology spectrum. This omission points to potential areas for further research and development, especially considering the likely advantages these techniques could offer when applied to LLMs designed for logical reasoning tasks.

10.2.2 Datasets, Benchmarks and Tasks

Exploring diverse datasets, benchmarks, and tasks is essential for evaluating LLMs in puzzle-solving. This section delves into datasets within our puzzle taxonomy, covering their formats, evaluation metrics, and methodologies. Figure 10.1.1 provides a detailed summary of the datasets utilized across different categories within the taxonomy, organized by puzzle type. The analysis showcases the versatility of LLMs and highlights the impact of the techniques discussed in Section 10.2.1, offering a comprehensive view of LLM performance across various puzzle types, revealing their capabilities, challenges, and potential areas for future research.

Rule-based Puzzles This exploration of rule-based puzzles focuses on assessing LLMs' comprehension within structured, closed-world environments. This category includes deterministic puzzles such as Sudoku, Rubik's Cube, Crosswords, and the 8-puzzle, all of which operate under a set of defined rules. In contrast, stochastic games like Minesweeper, and various card and social deduction games, feature variable outcomes from identical actions due to hidden elements. While research predominantly centers on deterministic puzzles, addressing the uncertainties in stochastic puzzles remains a promising direction for future research.

Deterministic Puzzles **Sudoku** is a benchmark for LLMs, challenging their logical reasoning capabilities. [258] fine-tuned GPT-2 [285] on 1 million Sudoku games, using a compact single-string format where empty cells are denoted by "-", and suggested that a matrix representation might enhance learning efficacy. [212]

Methods	**Rule-based Puzzles**		**Rule-less Puzzles**		
	Deterministic	**Stochastic**	**Riddles**	**Programming**	**Commonsense**
Prompting					
Few-shot	✓	✓	✓	✓	✓
Chain-of-Thought	✓	✓	✓	✓	✓
Self-refine	✓				
Auto-CoT					✓
Complexity CoT					✓
Plan & Solve					✓
Detective Thinking					✓
Self-Consistency	✓				✓
Tree-of-Thoughts	✓				
Tree-of-uncertain-Thoughts	✓				
Inferential Exclusion Prompting			✓		✓
Graph-of-Thoughts	✓				
Everything-of-thoughts	✓				
Hints			✓		✓
Introduction/Summarization	✓	✓	✓	✓	✓
Puzzle Translation					
Logic	✓				
Code					
Fine-Tuning	✓	✓	✓	✓	✓

Table 10.1: Methods used by each category of our taxonomy based on the puzzle benchmarks we collected

utilized nested lists for puzzle representation[3], finding the Tree-of-Thought (ToT) method most effective, especially for smaller puzzles. [126] explore neuro-symbolic approaches across Sudoku, Jobs puzzles, and other logic puzzles, showing that well-prompted LLMs can accurately generate answer set programming rules.

For **Rubik's Cube** and **Maze solvers**, [258] assessed GPT-2's spatial reasoning using over 2,400 Rubik's Cube samples and 10,000 mazes. Despite limited fine-tuning and token constraints, GPT-2 successfully solved the Rubik's Cube in one out of seven attempts, displaying potential despite a high rate of valid but incorrect solutions. [66] applied multiple methods such as CoT, Self-Consistency, and various Thoughts (ToT, GoT, XoT) on a 2×2×2 Rubik's Cube using GPT-3.5 and GPT-4. XoT with self-revision emerged as the most accurate, significantly outperforming others with a 77.6

[66] also evaluated the effectiveness of XoT on the spatial **8-Puzzle** and numerical **Game of 24**. The 8-Puzzle's challenges were solved with a remarkable 93.2% accuracy across 419 puzzles using XoT with revision, showcasing superior efficiency over few-shot prompting and CoT. This high accuracy, coupled with a reduced number of LLM invocations, underscores XoT's efficiency and potential in complex puzzle-solving contexts.

Regarding **Crosswords**, [306] and [71] fine-tuned T5 models [288] on extensive datasets of individual cryptic clues, revealing T5's advantages over traditional methods and highlighting areas for improvement, especially with quick clues and specified answer lengths. [164]'s comparison of BART [175] and T5 indicated a sub-30% accuracy for clue-answer tasks, with retrieval-augmented generation transformers surpassing fine-tuned LLMs. Additionally, [391] applied 5-shot prompting and ToT to GPT-4 on Crossword puzzles, significantly improving performance by solving 4 out of 20 puzzles and achieving a 60

[77] fine-tuned two models, "ChessGPT" and "ChessCLIP," using a collection of 3.2 million **chess puzzles**

[3] e.g., [[3,,,2], [1,,3,],[,1,,3],[4,,,1]]

from the Lichess dataset[4]. Each puzzle includes annotations for its rating, theme, and solution.

Lastly, [149] unveiled **BoardgameQA**, a dataset featuring multiple-choice questions set against a backdrop of contradictory facts and rules. Models must navigate these complexities to provide free-text answers. Their evaluation revealed that fine-tuning BERT-large and T5-XXL with proofs emerged as the most effective method, contrary to few-shot prompting on PaLM with CoT. Additionally, the presence of extra or conflicting information decreased accuracy.

Stochastic Puzzles The exploration of stochastic puzzles, represented by the **BoardgameQA** benchmark [149], highlights the challenges presented by scenarios with missing information, a characteristic of this puzzle category. It is observed that as the amount of missing information increases, the accuracy of fine-tuned models tends to decrease. Interestingly, this increased difficulty does not similarly affect the performance of prompt-tuned and few-shot learning methods, likely due to the use of larger models in these approaches.

Minesweeper, characterized by its unpredictability and hidden information, stands as a quintessential example of stochastic puzzles. This game challenges players to deduce the locations of mines based on numerical clues, offering a unique test of spatial reasoning. [186] evaluated LLMs on Minesweeper using different representations, including table and coordinate formats. While GPT-3.5 showed some initial understanding of the game mechanics, enhancements such as few-shot prompting had minimal impact. In contrast, GPT-4 demonstrated improved capabilities in mine identification but faced challenges in completing the board, emphasizing Minesweeper's utility in assessing LLMs' strategic thinking and inference skills. The experiments highlighted the advantages of using the coordinate representation to aid LLM comprehension over the table format.

Card games, especially Poker, are also notable within the stochastic puzzle category where strategic decision-making is critical. Simplified Poker variants challenge players to infer opponents' cards and calculate odds amidst hidden intentions. [109] observed that while models like ChatGPT and GPT-4 understand advanced strategies in Poker's pre-flop round, they do not achieve Game Theory Optimal (GTO) play. ChatGPT tends to adopt a more conservative strategy, whereas GPT-4 exhibits a more aggressive style of play. Furthermore, [121] applied a Reinforcement Learning-trained OPT-1.3B model across all phases of Poker, revealing superior performance in terms of win rates and efficiency, showcasing LLMs' proficiency in managing complex strategies in stochastic settings. Another agent leveraging GPT-4 [108] achieved significant success in various imperfect information card games.

Social deduction games, such as Werewolf and Avalon, blend logical reasoning with intricate social dynamics, categorizing them within the stochastic puzzle domain. These games require players to deduce roles amidst unpredictable human behavior. [385] introduced a framework for Werewolf that utilizes LLMs without tuning, relying on the retrieval and reflection of past communications to enhance gameplay. This approach highlights the LLMs' ability to utilize historical interactions for strategic decision-making. Additionally, frameworks for Avalon [364, 166] demonstrate how LLMs can adeptly navigate scenarios requiring social manipulation and deduction, further underscoring LLMs' capabilities in managing the complex interplay of logic and social interaction inherent in such games.

Programming Puzzles **P3 (Python Programming Puzzles)** [318] offers a range of Python programming challenges, from straightforward string manipulations to complex tasks, such as the Tower of Hanoi and algorithmic puzzles. Models applied to these puzzles include enumerative solvers for building Abstract Syntax Trees and autoregressive Language Model Solvers such as GPT-3 and Codex [37], employing varied prompting techniques. The evaluation metric, pass@k, indicates the models' ability to solve a puzzle within a given number of attempts [37]. Results show a correlation between puzzle difficulty for both models and humans, with descriptive prompts enhancing model performance.

[316] introduce a dataset comprised of 530 code snippets from programming courses, presenting puzzles in a multiple-choice format. The distinction between questions with and without code snippets offers a unique perspective on LLMs' problem-solving strategies. The dataset categorizes questions into six types, including true/false and output prediction. GPT models were evaluated, revealing that code inclusion significantly

[4] https://lichess.org/

increases puzzle complexity. Accuracy rates vary, with higher performance on completion-oriented questions, suggesting that LLMs' effectiveness can depend heavily on question format and content.

Commonsense Reasoning Puzzles **True Detective** [54] presents detective puzzles in long-form stories, challenging LLMs such as GPT-3.5/4 to draw conclusions. Various methods, including CoT and Golden-CoT, are applied, revealing difficulties in making final inferences despite all necessary information being available. Golden-CoT provides the model with the reasoning behind the correct answer, so the model only needs to understand this reasoning and extract the answer. While Vanilla and CoT approaches perform close to random, Golden-CoT demonstrates significantly better accuracy, particularly with GPT-4. However, even with Golden-CoT, GPT-3.5 achieves a solve rate of only 63%, whereas GPT-4 matches human solver results (without access to the reasoning behind the answer).

DetectBench [105] containing 1200 questions, also evaluates informal reasoning in real-life contexts. It tests methods such as use of hints, various CoT approaches and detective thinking on models including GPT-4, GPT-3.5, GLM-4 and Llama2. Hints emerges as a powerful aid, with larger models generally outperforming smaller ones. The effectiveness of different approaches vary, with detective thinking effectively assisting most of the models.

LatEval [124] introduces a conversational format with English and Chinese stories, requiring players to ask yes/no questions before providing an answer. GPT-3.5, GPT-4, and various other Chat models are evaluated on their ability to ask relevant questions and maintain consistency with the truth. Larger models do not necessarily show advanced performance in question relevance. However, GPT-4 demonstrates the highest answer consistency, though there is still significant room for improvement.

PuzzTe [337], with its array of comparison, knights and knaves, and zebra puzzles, represents a potentially rich resource for LLM testing. Despite not yet being applied to LLMs, its generated puzzle answers by Mace4 model finder and Prover9 theorem prover[5] indicate its potential for future LLM evaluations.

Table 10.2 presents a comprehensive summary of the datasets and tasks related to each category within our taxonomy of puzzles. A detailed analysis of this table highlights a significant number of datasets for rule-based deterministic puzzles, such as Sudoku and Rubik's Cube, as well as a variety of rule-less riddles. This demonstrates a strong research interest and resource availability in these areas, underscoring the active exploration and validation within these fields.

Conversely, there is a noticeable scarcity in datasets for rule-based stochastic puzzles and rule-less programming puzzles. This deficiency suggests a significant opportunity for further research and the development of new datasets. Expanding the collection of datasets in these less-represented categories could provide more diverse challenges, which would enhance the problem-solving capabilities of LLMs.

By addressing this gap, the research community could create a more balanced and comprehensive set of benchmarks. These benchmarks would encompass a wider spectrum of puzzle-solving scenarios, including those that involve uncertainty and complex logic-based problem-solving. Such developments could potentially catalyze advancements in the ability of LLMs to navigate and resolve complex, uncertain scenarios effectively, thereby pushing the boundaries of what these models can achieve.

Performance Analysis: *Rule-based / Deterministic*: Methods such as ToT and XoT (§ 10.2.1), which introduce structured reasoning sequences, typically enhance model reasoning abilities as the complexity of the puzzle's structure increases [66]. However, performance analyses in areas like BoardgameQA and crossword puzzles still show generally poor model performance, suggesting that while these approaches are promising, there is room for improvement in handling even structured deterministic challenges.

Rule-based/Stochastic: Fine-tuning is prevalent in this category, enabling LLMs to effectively grasp basic rules and handle simpler scenarios. Nonetheless, these models often falter in more complex settings that require extensive multi-step reasoning and the management of uncertainty, pointing to limitations in current methods when faced with the stochastic elements of puzzles [186].

Rule-less/Riddles & Commonsense: In this category, there is a notable performance gap between LLMs and human levels. While techniques like Chain of Thought (CoT) improve accuracy, they still fall short of

[5] https://www.cs.unm.edu/ mccune/prover9/

Category	Type	Datasets
Rule-based	Deterministic	BoardgameQA [149], Sudoku [258, 212, 126], Rubik's Cube [258, 66], Maze [258], Crossword [391, 306, 71, 164], 8-puzzle [66], Game of 24 [66, 391], Chess [126, 77]
	Stochastic	Minesweeper [186], BoardgameQA [149], Card Games [121, 109], Social Deduction Games [364, 385, 166]
Rule-less	Riddles	BrainTeaser [130], RiddleSense [194], BiRdQA [410], CC-Riddle [382], PUZZLEQA [414], MARB [343]
	Programming	P3 [318], [316]
	Commonsense	LatEval [124], True Detective [54], DetectBench [105], MARB [343]

Table 10.2: Collected Datasets and Tasks for each Category

human evaluation outcomes. This gap highlights the challenge for LLMs in bridging intuitive and inferential reasoning required in rule-less contexts.

Rule-less/Programming: Programming puzzles remain challenging for LLMs, reflecting similar difficulties faced by humans [318]. Tasks that require code analysis and logic reasoning in multiple-choice formats have proven particularly tough, underscoring the ongoing challenges in applying LLMs to complex, technical reasoning tasks [316].

Furthermore, the format of questions significantly affects the effectiveness of puzzle-solving by LLMs. Multiple-choice setups, for instance, tend to simplify tasks for LLMs by narrowing the solution search space, while free-text formats increase the difficulty level by requiring more open-ended reasoning.

Puzzle Generation research is currently limited, which is likely due to the prerequisite that understanding and solving puzzles is necessary before one can generate them effectively. The few works that was found on puzzle generation reveal mixed results. For instance, GPT-3.5's attempts at generating puzzles with answers demonstrated poor outcomes [414]. Conversely, the introduction of ACES, an autotelic generation method for creating diverse programming puzzles, shows how semantic descriptors produced by LLMs can be leveraged for creative puzzle creation [276]. Recent works have also explored the generation of crossword puzzles in different languages, utilizing LLMs to create clues and puzzle layouts [426, 400, 399], indicating a growing interest and potential in this area.

Despite these advancements, significant inconsistencies remain in the ability of LLMs to solve puzzles, particularly in the domain of riddle-solving tasks within rule-less puzzles. To address these challenges, the subsequent chapter proposes a novel method designed to enhance the reasoning capabilities of LLMs, specifically for riddle-solving tasks [269]. The proposed approach centers on the generation of counterfactual riddles—riddles constructed to require the same reasoning steps as the original but set within alternative contexts. By presenting these counterfactual riddles as additional examples during in-context learning, the method aims to systematically improve the ability of LLMs to reason through and solve complex riddles, thereby addressing a critical gap in current puzzle-solving performance.

10.3 Generation of Counterfactual Riddles

The objective of this section is to introduce a novel prompting methodology designed to enhance the reasoning capabilities of LLMs. This approach leverages the generation of counterfactual riddles, which require identical reasoning steps for their resolution but are framed within an alternative context. These generated riddles, referred to in the literature as *context-reconstructed riddles* [131, 269], are central to our exploration of reasoning mechanisms in LLMs.

Reasoning with Language Models The reasoning capabilities of language models have been extensively investigated across various domains. These include commonsense reasoning [315], arithmetic reasoning [216], abductive reasoning [416], inductive reasoning [112], deductive reasoning [313], and analogical reasoning

[333], among others. Of particular relevance to our study is the exploration of structured, rule-based thinking processes often referred to as *vertical thinking*. This type of reasoning, characterized by the systematic application of logic and rules, has been widely studied in the context of established datasets, such as *RiddleSense* and *PIQA* [195, 22]. These studies have revealed significant insights into the reasoning patterns exhibited by language models.

Conversely, creative reasoning has remained an underexplored domain in the evaluation of LLMs, frequently being excluded from traditional reasoning benchmarks [328, 314]. This exclusion has created a notable gap in the literature, particularly in light of the emergent capabilities demonstrated by larger and more sophisticated models [369]. Puzzle-solving, which inherently requires creative and *out-of-the-box* thinking, provides a compelling framework for exploring this underrepresented aspect of reasoning [95, 195, 411].

Taking this further, the concept of *lateral thinking*, which involves the deliberate disruption of default assumptions and associations to arrive at novel solutions, has been proposed as a mechanism for solving more complex and unconventional puzzles. This was first systematically demonstrated in the *BrainTeaser* dataset [131], which highlights the potential of lateral thinking processes to challenge traditional reasoning paradigms.

In this chapter, we aim to examine the interplay between vertical and lateral reasoning abilities in LLMs by employing specifically designed prompts to probe their puzzle-solving capabilities. Through this exploration, we seek to bridge the gap in the literature and provide a comprehensive framework for assessing both linear and creative reasoning processes in language models.

Large Language Models and Prompting The discovery of reasoning patterns in LLMs is frequently achieved through the application of various prompting techniques [282]. These strategies range from straightforward zero-shot prompts to more elaborate multi-stage prompting frameworks. Zero-shot prompting often employs intuitive and concise instructions, such as the widely recognized phrase *“Let’s think step-by-step”*, which has demonstrated significant improvements in reasoning tasks [158]. However, the design space for such “magic prompts” remains vast and largely uncharted, posing challenges to the systematic identification of optimal prompts.

A significant breakthrough in this area was the introduction of Chain-of-Thought (CoT) prompting [371], which formalized the approach of querying models for intermediate reasoning steps. This method proved particularly effective for eliciting complex reasoning processes in larger models. Despite its success, the application of few-shot prompting, which involves providing a set of demonstrations within the input, presents challenges. These challenges include the inherent instability in exemplar selection [215, 245] and the difficulty of optimizing their placement [67].

Similarity-based retrieval has emerged as the default technique for exemplar selection in few-shot prompting [206, 283, 67], with further improvements focused on optimizing the ordering of selected exemplars [377]. Recent work has also explored task-specific factors, such as the complexity of reasoning paths [86] and the diversity of exemplars [412], which have proven effective in enhancing reasoning performance. These developments emphasize the importance of uncovering hidden patterns in the data that drive reasoning performance, rather than relying solely on semantic similarity for exemplar selection.

Building on these insights, we propose a novel approach to exemplar crafting that prioritizes the promotion of latent reasoning patterns over the semantic similarity of data samples. While maintaining simple similarity-based retrieval for exemplar placement, we demonstrate that focusing on these reasoning patterns is sufficient to achieve advanced reasoning capabilities in LLMs. By outperforming alternative prompting techniques, our approach underscores the adequacy of reasoning-focused exemplars in enhancing model performance, without the need for further engineering in few-shot settings.

10.3.1 Methodology

Consider the following two riddles: *R1: “A man shaves every day, yet keeps his beard long”* and *R2: “What has a beard but never needs to shave?”*. At first glance, these riddles appear to share semantic similarities, as they involve analogous linguistic structures and refer to overlapping objects or concepts. However, the reasoning processes required to solve them differ significantly due to their distinct contextual framings.

In *R1*, the answer is “A barber.” The word “beard” is employed in the context of human grooming and personal appearance, reflecting a literal interpretation tied to the profession of the barber. In contrast, *R2* uses “beard” metaphorically in a botanical or natural context, with the answer being “A tree.” Here, the term “beard” refers to certain botanical features, such as the “beard” of oak trees, requiring a different pathway of reasoning that departs from the literal context of human grooming.

This divergence in reasoning is further highlighted when introducing a new riddle: *“I plant seeds every day, yet don’t have a single plot”*. The correct answer in a creative context is “An author.” Authors metaphorically “plant seeds” of ideas through their writing, fostering abstract concepts without necessarily working with physical plots of land. The phrase “plot” could refer to a story’s structure, adding layers of interpretive complexity. Conversely, “A farmer” is not an appropriate answer because farmers typically work with tangible plots of land, which contradicts the riddle’s framing of “not having a single plot.”

In this scenario, a reconstructed version of *R1*, such as *R3: “Tom attends class every day but doesn’t do any homework”*, would offer a clearer reasoning pathway for the model to follow. Although *R2* is semantically closer to *R1* in terms of shared terms like “beard,” its contextual framing diverges due to its reliance on natural rather than human contexts. This difference demonstrates how contextual framing can either clarify or obscure the reasoning trajectory needed to arrive at the intended answer.

10.3.2 RISCORE Method

Building upon this example, we propose the **RISCORE** (***RI****ddle* ***S****olving with* ***CO****ntext* ***RE****contruciton*) [269] prompting method, which is specifically designed to enhance the riddle-solving capabilities of LLMs in in-context learning tasks. The core idea behind RISCORE is to supplement each exemplar in a few-shot (FS) learning setup with a contextually reconstructed version of itself. By altering the context while preserving the underlying reasoning process, the method ensures that the model develops a robust and coherent reasoning trajectory, enabling it to generalize effectively to new riddles.

Unlike traditional FS methods, which rely solely on real examples extracted from datasets, RISCORE generates additional examples that adapt the original context to a different framing. For instance, the reconstructed example *R3: “Tom attends class every day but doesn’t do any homework”* preserves the logical structure of *R1* while recontextualizing its content. This process allows the model to focus on the reasoning steps rather than being misled by superficial semantic similarities, as seen in the case of *R2*.

As depicted in Figure 10.3.1, RISCORE extends existing FS methods [67, 363, 333] by augmenting FS samples with automatically generated, context-reconstructed examples. Importantly, RISCORE operates independently of the exemplar selection process, allowing for compatibility with various selection techniques. In our implementation, we rely on semantic similarity for optimal exemplar selection; however, other methods from the literature can be employed seamlessly.

The key advantage of RISCORE lies in its ability to amplify the effectiveness of FS learning by introducing contextually adapted examples that highlight hidden reasoning patterns embedded in the data. These context-reconstructed exemplars not only preserve the logical structure of the original riddles but also guide the model toward more precise reasoning pathways. As detailed in Section 10.3.2, this approach has consistently demonstrated improved performance across multiple reasoning benchmarks. Remarkably, the inclusion of context-adapted examples has, in most cases, outperformed the use of real dataset examples, as evidenced by the results in Section 10.5.

Methodology for Generating Contextually Reconstructed Riddles

This section outlines a systematic approach for generating high-quality, contextually reconstructed riddles designed to function as few-shot exemplars in the Multiple-Choice Question Answering (MQA) format. These reconstructed riddles, when used in conjunction with their original counterparts, aim to improve model performance on tasks that require both lateral and vertical reasoning. Building upon the semi-automated pipeline introduced in the BrainTeaser framework [131], the proposed method fully automates the generation process by leveraging the advanced capabilities of LLMs. An overview of the automated pipeline is illustrated in Figure 10.3.2. The methodology is divided into two distinct steps: the generation of question-answer pairs, which involves creating a question along with its correct answer, and the generation of distractors, which

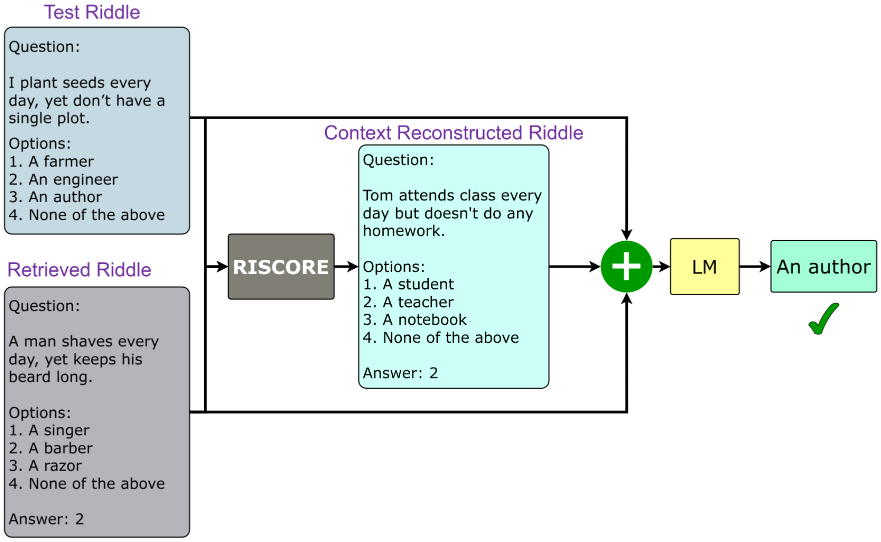


Figure 10.3.1: An overview of RISCORE, where the reconstructed instances, along with their original counterparts, are incorporated as exemplars in the few-shot setting to enhance the model's riddle solving ability [269].

entails producing incorrect answers for the riddle. The latter step is particularly critical and challenging, as the distractors must be guaranteed to be incorrect while avoiding being overly obvious as wrong choices. A detailed analysis of these two steps is provided below.

Step 1: Generation of Question-Answer Pairs

The initial phase involves the generation of a single contextually reconstructed Question-Answer pair for each selected instance. At this stage, distractors—incorrect answer options included in multiple-choice formats—are temporarily excluded to focus solely on the correct answer. Distractors are specifically designed to challenge the reasoning depth of the model by appearing plausible while being definitively incorrect.

To create the reconstructed Question-Answer pair, the LLM is provided with the original riddle, its correct answer, and a task-specific system prompt. This prompt instructs the model to analyze the given riddle, comprehend the reasoning process that connects the question to the answer, and subsequently generate a new riddle that adheres to the same reasoning pathway. By supplying both the question and the correct answer, the cognitive load of independently solving the riddle is alleviated, facilitating more accurate reconstruction [333].

To further enhance the robustness of the methodology, the process is implemented in both zero-shot and few-shot settings. For the latter, pre-existing pairs of original and contextually reconstructed riddles from the BrainTeaser dataset are employed to provide demonstrations. Once the Question-Answer pair is generated, a filtering stage is applied to ensure the quality of the riddle and its alignment with the dataset. This filtering involves applying dataset-specific rules regarding the structure of the question and the appropriateness of the answer, ensuring adaptability to various datasets.

Step 2: Generation of Distractors

The subsequent step focuses on generating incorrect answer options, or distractors, to accompany the reconstructed Question-Answer pair in the multiple-choice format. Although the task may appear straightforward, it entails significant challenges. First, the distractors must match the original number of options and be definitively incorrect when compared to the correct answer. At the same time, they must remain contextually plausible, as excessive divergence from the correct answer could undermine the credibility and difficulty of the riddle. Furthermore, scenarios where the correct answer is "None of the above," as observed in the BrainTeaser dataset, necessitate careful construction of distractors to ensure they remain invalid.

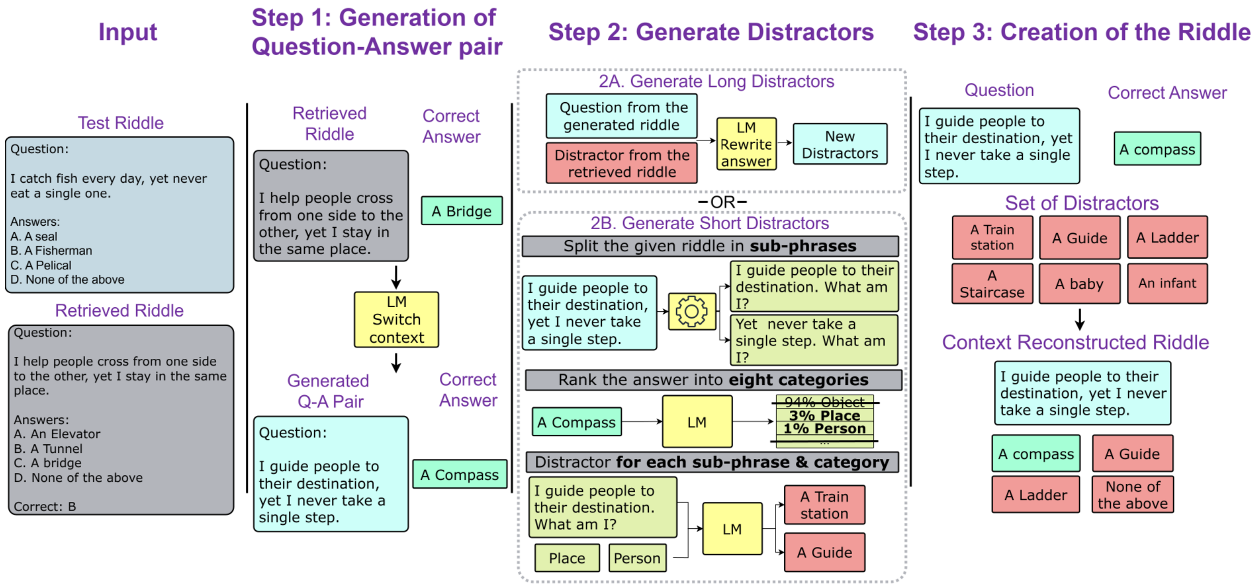

Input
Step 1: Generation of Question-Answer pair
Step 2: Generate Distractors
Step 3: Creation of the Riddle
Test Riddle
Question:
I catch fish every day, yet never eat a single one.
Answers:
A. A seal
B. A Fisherman
C. A Pelical
D. None of the above
Retrieved Riddle
Question:
I help people cross from one side to the other, yet I stay in the same place.
Answers:
A. An Elevator
B. A Tunnel
C. A bridge
D. None of the above
Correct: B
Retrieved Riddle
Correct Answer
Question:
I help people cross from one side to the other, yet I stay in the same place.
A Bridge
LM Switch context
Generated Q-A Pair
Correct Answer
Question:
I guide people to their destination, yet I never take a single step.
A Compass
2A. Generate Long Distractors
Question from the generated riddle
Distractor from the retrieved riddle
LM Rewrite answer
New Distractors
–OR–
2B. Generate Short Distractors
Split the given riddle in **sub-phrases**
I guide people to their destination, yet I never take a single step.
I guide people to their destination. What am I?
Yet never take a single step. What am I?
Rank the answer into **eight categories**
A Compass
LM
~~94% Object~~
3% Place
1% Person
~~...~~
Distractor **for each sub-phrase & category**
I guide people to their destination. What am I?
Place
Person
LM
A Train station
A Guide
Question
Correct Answer
I guide people to their destination, yet I never take a single step.
A compass
Set of Distractors
A Train station
A Guide
A Ladder
A Staircase
A baby
An infant
Context Reconstructed Riddle
I guide people to their destination, yet I never take a single step.
A compass
A Guide
A Ladder
None of the above

Figure 10.3.2: An overview of the automated method for generating a context-reconstructed riddle [269].

The length of the distractors poses an additional complexity. For example, datasets such as BrainTeaser predominantly feature multi-word answers, while others like RiddleSense often contain single-word answers. To address these variations, distinct methodologies are applied for generating long and short distractors, as detailed below.

Generation of Long Distractors

For riddles requiring long distractors, the original distractors are presented to the LLM alongside the newly generated question from Step 1. Each distractor is processed individually and recontextualized to align with the newly constructed riddle. This ensures that the distractors remain relevant, although the quality of integration may vary. In cases where the correct answer of the original riddle is "None of the above," additional distractors are generated by directly prompting the LLM to create new options based on the reconstructed Question-Answer pair.

Two distinct pipelines were utilized to generate at least three distractors for each multiple-choice question, ensuring comprehensive coverage of various potential scenarios and maintaining the integrity of the riddle-solving tasks. The distractors were crafted to be contextually coherent while remaining incorrect, thereby enhancing the challenge presented to the model. The methodologies employed are detailed below.

Pipeline 1: The first pipeline employed a structured approach that relied on a system-user prompt designed to guide the model in analyzing the reconstructed Question-Answer pair. The model was instructed to comprehend the riddle, identify the reasoning process connecting the question to the answer, and generate a distractor by focusing on aspects of the concept that could be interpreted as more deceptive or challenging. This method capitalized on the model's ability to produce nuanced distractors derived from the underlying reasoning pathway, ensuring that the generated distractors remained plausible yet incorrect. The pipeline reliably produced one of the three required distractors while maintaining contextual alignment with the question's premise.

Pipeline 2: The second pipeline adopted a more intricate methodology, designed to integrate elements from the reconstructed question's context into the generated distractors. In this approach, the model was prompted with a system-user instruction, providing it with the reconstructed question (excluding its correct answer) and the incorrect distractors from the original question. The option "None of the above" was excluded to simplify the task and focus the model's efforts on modifying the distractors provided. The model was tasked with adapting the given distractors by incorporating elements from the setting or context described in the question, while explicitly avoiding any resemblance to the correct answer.

This approach introduced additional complexity, as the absence of the correct answer necessitated that the model generate distractors independently based on its interpretation of the reconstructed question. Consequently, the quality of the distractors varied. In some cases, the distractors produced suboptimal contexts or lacked sufficient challenge. However, despite occasional shortcomings in quality, the distractors remained unequivocally incorrect, fulfilling their primary function of providing plausible yet invalid options. Moreover, it was observed that, when paired with both original and contextually reconstructed examples, these minor deficiencies did not significantly impact the overall performance of the model during evaluation.

To further enhance the coherence and relevance of the distractors, two additional distractors were generated by slightly modifying the original concepts to better align with the reconstructed context. These supplementary distractors added a layer of consistency to the multiple-choice options, ensuring that the distractors were not only incorrect but also contextually appropriate to the underlying premise of the riddle.

Lastly, to prepare the final dataset for use, a random selection of two distractors from the three generated options was performed. These two distractors were shuffled with the correct answer, and the option “None of the above” was appended as the final choice. This randomization was implemented to eliminate positional bias and ensure the fairness of the multiple-choice format. In instances where “None of the above” was identified as the correct answer, all three generated distractors were included without shuffling.

The resulting dataset maintained a balance between contextually relevant distractors and challenging incorrect options, thereby supporting robust evaluation of the model’s reasoning capabilities. This comprehensive approach to distractor generation ensured that the dataset was both high-quality and suitable for the intended tasks, laying a strong foundation for subsequent experimental investigations.

Generation of Short Distractors

For datasets with single-word answers, such as RiddleSense, generating contextually aligned distractors is more challenging due to the brevity of the answers. To address this, the reconstructed answer from Step 1 is categorized into mutually exclusive semantic groups, as outlined below. This categorization informs the generation of distractors by providing the LLM with two categories closely related to, but distinct from, the correct answer’s category. For example, if the correct answer belongs to the “Nature” category, distractors may be drawn from categories like “Person” or “Place.” The LLM is then tasked with generating plausible but incorrect answers within these specified categories.

To ensure the generated distractors are both incorrect and contextually relevant, additional steps are undertaken. The riddle is divided into smaller phrases, and interrogative words are identified. For purely descriptive riddles lacking direct queries, phrases such as “What am I?” are appended to clarify the intended question. Filtering procedures are subsequently applied to eliminate duplicates and validate the distinctiveness of the distractors. If the quantity of distinct distractors remains insufficient, WordNet [244] is utilized to augment the distractor set.

In this setting, a distinct approach was required due to the specific characteristics of the dataset. The answers and distractors were primarily limited to single-word responses, whereas the corresponding questions featured detailed and complex settings incorporating punctuation, conjunctions, and other nuanced linguistic structures. To address these unique requirements, two specialized pipelines were developed to generate distractors that align with this format while maintaining relevance and quality.

Pipeline 1: The first pipeline employed a granular methodology that involved segmenting the reconstructed question into smaller subphrases. This segmentation was achieved by identifying punctuation marks and conjunctions as natural breakpoints. In instances where fewer than three distinct subphrases were generated, additional splits were introduced at the position of the word “and” to ensure sufficient subdivision. Furthermore, the presence of interrogative words was detected. For questions lacking a direct inquiry and consisting solely of descriptive elements, the phrase “What am I?” was appended. This addition was not arbitrary but followed the structural conventions of riddles within the dataset, where “What am I?” frequently serves as a standard closing query leading to single-word answers.

Once the reconstructed question was segmented and refined with an appended query (if necessary), the model was prompted to generate incorrect answers for each subphrase concatenated with the appended question. This ensured that distractors were aligned with distinct aspects of the question's context. However, this approach occasionally led to distractors that were too similar to the correct answer, particularly when subphrases contained key ideas central to the riddle's solution.

To mitigate this issue, an intermediate classification step was introduced. Using the *facebook/bart-large-mnli* model [176], accessed via Hugging Face, the correct answer was categorized into one of eight mutually exclusive classes: *food, person, object, animal, nature, time, place, concept.* These categories were specifically designed to avoid overlap and provide clear distinctions between classes.

The category predicted for the correct answer was then used to guide the generation of distractors. For each subphrase concatenated with the question, the model was provided with the two most similar categories (excluding the correct answer's category). A system-user instruction prompted the model to generate a plausible but incorrect answer consistent with the given category and contextual setting. This approach ensured that the distractors were contextually aligned with the question while remaining distinct from the correct answer.

After this process, a filtering step was applied to validate the distinctiveness and relevance of the generated distractors. The pipeline utilized LLMs, including the Llama3-8B and Llama3-70B models [7], to produce high-quality distractors.

Pipeline 2: In cases where the first pipeline did not yield a sufficient number of high-quality distractors, a secondary pipeline was employed to augment the distractor set. This approach relied on WordNet [45], a lexical database, to retrieve synonyms and hyponyms for each generated distractor or, if necessary, for the original question's distractors. The retrieved terms were added as potential distractors to diversify the options.

Generation of Context-Reconstructed MQA Riddles

Once the distractor set was compiled, four distractors were randomly selected and combined with the correct answer, which was placed in a random position to eliminate potential positional bias. To ensure the integrity and quality of the dataset, a restriction was imposed that at least two of the four required distractors must be generated through the first pipeline. This restriction was necessary because distractors derived through WordNet augmentation were generally observed to be of lower quality compared to those produced directly by the model. If this requirement was not satisfied, the corresponding instance was excluded from the contextual reconstruction process to maintain overall quality.

The two complementary pipelines, in combination, ensured that the generated distractors were diverse, contextually aligned, and sufficiently challenging for the task. By tailoring the distractor generation process to the unique characteristics of the dataset, this methodology provided a robust framework for creating high-quality multiple-choice questions. The rigorous filtering and augmentation steps further enhanced the reliability of the dataset, making it suitable for rigorous evaluation of lateral and vertical reasoning tasks.

In the final step, the reconstructed Question-Answer pair and the generated distractors are combined to create the complete multiple-choice riddle. The distractors are randomly shuffled, and the correct answer is assigned to a random position to eliminate positional bias.

The desired contextual reconstructions were successfully generated for both datasets, providing a robust foundation for the study. However, a notable issue was encountered during the quality control filtering process: certain originally selected examples were excluded as they failed to meet the quality criteria, resulting in the absence of corresponding reconstructed examples. This posed a challenge in maintaining a sufficient number of exemplars for in-context learning in configurations requiring two, four, or eight exemplars (i.e., **RISCORE**).

To address this limitation, a structured methodology was employed to supplement the exemplar set with high-quality examples. This process consisted of two key steps, detailed as follows.

1. Initially, the most semantically similar examples from the original dataset were identified and used as in-context learning exemplars. These original examples were paired with their automatically generated

contextual reconstructions, ensuring that both the original and reconstructed pairs were represented in the exemplar set. This approach allowed for an efficient supplementation of missing reconstructed examples by leveraging pre-existing data. However, in instances where this strategy did not yield a sufficient number of exemplars to satisfy the requirements of the RISCORE configurations, a more systematic method was employed.

2. To systematically address the exemplar shortfall, embeddings were generated for the entire set of original examples and their contextual reconstructions that had not yet been incorporated into the current exemplars. These embeddings were computed to represent the semantic characteristics of each example, facilitating similarity-based retrieval. Cosine similarity was then utilized to identify the most semantically similar examples from this pool, ensuring that the selections closely aligned with the existing exemplars. Importantly, this process was not restricted to the original training set. Examples included in the similarity search also encompassed reconstructed examples, allowing for greater flexibility in identifying suitable pairs. The most similar examples were selected iteratively, with each chosen example paired with its corresponding counterpart (original or reconstructed, as needed) to maintain the integrity of the reasoning process. This iterative process was repeated until the required number of exemplars for the specified configuration (two, four, or eight) was achieved. The careful selection of semantically similar pairs ensured that the supplemented exemplars maintained consistency with the dataset's reasoning pathways and contextual framing.

By combining semantically similar examples with their corresponding contextual reconstructions and systematically supplementing the exemplar set through embedding-based retrieval, the issue of insufficient examples was effectively mitigated. This approach ensured that the exemplar set adhered to the required configurations while preserving the quality and coherence necessary for effective in-context learning. The structured methodology also minimized the risk of introducing irrelevant or low-quality examples, thereby enhancing the reliability and applicability of the dataset for reasoning tasks.

10.4 Experiments

10.4.1 Datasets

The proposed methodology was evaluated by testing various LLMs on two carefully chosen datasets, *BrainTeaser* [131] and *RiddleSense* [195], which address distinct reasoning paradigms: lateral and vertical reasoning, respectively. The performance of the models was compared against several established baselines to assess the effectiveness of the method.

The BrainTeaser task introduced at SemEval-2024 [**jiang-ilievski-ma:2024:SemEval2024**, 131] presents a set of lateral thinking puzzles formatted as multiple-choice question-answering (QA) tasks. Each question in the dataset is accompanied by four answer options, of which only one is correct, while the remaining three serve as distractors. Notably, the final option in every question is consistently labeled as "None of the above," adding an additional layer of complexity to the task by requiring the model to evaluate all options comprehensively.

The task is divided into two distinct subtasks to address different aspects of lateral reasoning. The first subtask, *Task A: Sentence Puzzle*, involves puzzles expressed through full-sentence descriptions, requiring models to navigate more complex linguistic structures and contextual clues. The second subtask, *Task B: Word Puzzle*, focuses on shorter, more concise puzzles, often relying on single-word or minimal phrasing for their formulation. This distinction ensures that the dataset evaluates a broad spectrum of lateral reasoning capabilities in LLMs.

In addition to the original puzzles, the dataset includes adversarial subsets specifically crafted to enhance its robustness and challenge the models further. These adversarial subsets were generated through manual modifications of the original brain teasers, with care taken to preserve the integrity of the underlying reasoning paths. The perturbations introduced into the data were designed to create two distinct forms of variation:

1. **Semantic Reconstruction**: In this approach, each original question was rephrased or modified semantically, while the correct answer and distractors remained unchanged. This type of reconstruction

tests the model's ability to recognize and adapt to variations in linguistic structure and phrasing without altering the core reasoning process.

2. **Context Reconstruction**: This method involved altering the situational context described in the brain teaser while maintaining the original reasoning pathway. By changing the context, the model is challenged to abstract the reasoning process from specific scenarios and apply it in new settings.

An example of these data triplets is provided in Table 10.3, showcasing how the original puzzles are systematically transformed into semantic and contextually reconstructed versions while preserving their reasoning consistency. *BrainTeaser* was specifically selected because it contains manually crafted context reconstructions, which provide an upper bound for model performance when incorporated into the input. These manually created reconstructions serve as a benchmark for assessing the quality of context reconstructions generated by the proposed automated method. By comparing model performance using manual and automated reconstructions, the efficacy of the method in replicating high-quality contextual examples can be evaluated.

Question	**Choice**
Original	
What kind of nut has no shell?	A peanut.
	A doughnut.
	A walnut.
	None of above.
Semantic Reconstruction	
Which nut doesn't have a shell?	**A doughnut.**
	A walnut.
	A peanut.
	None of above.
Context Reconstruction	
Which type of bell doesn't make a sound?	A fire bell.
	A cow bell.
	A bluebell.
	None of above.

Table 10.3: Illustration of the structure of each sub-task's dataset, showcasing the original statement along with its two adversarials.

Additionally, the riddles in *BrainTeaser* were specifically designed to challenge the lateral thinking capabilities of LLMs. Lateral reasoning involves finding creative solutions that deviate from traditional logical pathways, making it a cognitively demanding process for models. Prior research has identified lateral reasoning as a significant challenge for LLMs [95], further underscoring the relevance of this dataset for the evaluation.

The *RiddleSense* dataset was selected to complement *BrainTeaser* by focusing on vertical reasoning tasks. Vertical reasoning involves the application of systematic, rule-based logic to solve riddles, requiring the model to follow a structured and sequential reasoning process. The inclusion of *RiddleSense* allowed for a broader evaluation of the method's applicability across different reasoning paradigms and riddle types. Unlike *BrainTeaser*, the *RiddleSense* dataset does not include manually curated context reconstructions. As a result, the automated contextual reconstruction method was employed exclusively in this dataset, without ground-truth reconstructions for comparison. This setup provided a valuable opportunity to test the robustness and adaptability of the method in scenarios where manual reconstructions are unavailable.

The combination of *BrainTeaser* and *RiddleSense* ensured a comprehensive evaluation of the proposed methodology. The manually crafted reconstructions in *BrainTeaser* provided a critical benchmark, enabling a direct comparison of automated and manual reconstructions, while the exclusive use of automated reconstructions in *RiddleSense* highlighted the generalizability of the method. Together, these datasets allowed for an in-depth analysis of the method's effectiveness in advancing the reasoning capabilities of LLMs across a diverse set of cognitive tasks, demonstrating its potential to address both lateral and vertical reasoning challenges. Table 10.4 provides statistics for the utilized datasets.

Dataset	**Train**	**Dev**	**Test**
BrainTeaser - SP	$507_{(169x3)}$	$120_{(40x3)}$	$120_{(40x3)}$
RiddleSense (initial)	3510	1021	—
RiddleSense (filtered)		720	
RiddleSense (sampled 50%)		360	

Table 10.4: Data statistics for the BrainTeaser and RiddleSense.

10.4.2 Baselines

To evaluate the proposed methodology, a variety of prompting techniques were employed as baselines, including zero-shot (**ZS**), few-shot (**FS**), and Chain of Thought (**CoT**) methods. These approaches allowed for a comprehensive comparison of model performance across multiple configurations and reasoning paradigms.

Chain of Thought Prompting in Zero-Shot Settings (CoT ZS) The Chain of Thought (CoT) prompting approach was employed in a zero-shot configuration, wherein models were prompted to solve riddles in a step-by-step manner without the inclusion of any exemplars. This method follows the framework proposed by [371], leveraging the natural reasoning abilities of large language models to decompose complex riddles into logical steps. By evaluating the models under this setup, insights into their inherent riddle-solving capabilities were obtained, particularly in scenarios where no prior examples or contextual information were provided.

Few-Shot Prompting (FS) Few-shot prompting was utilized to examine the impact of exemplar-based learning on the models' reasoning performance. Experiments were conducted with varying numbers of exemplars, specifically 2-shot, 4-shot, and 8-shot configurations. The selection of exemplars was informed by two distinct strategies. The first involved randomly selecting examples from the dataset, referred to as **Rand**. The second strategy, termed **Sim**, employed a semantic similarity model[6] [408] to identify riddles with minimal semantic distance from the test riddle.

Notably, the 8-shot limit was imposed due to evidence suggesting that the reasoning abilities of large language models deteriorate with excessively long input sequences [174]. This constraint ensured that the evaluation focused on configurations conducive to optimal model performance.

Few-Shot Prompting with Chain of Thought Explanations (CoT FS) To enhance the models' comprehension of the reasoning process, few-shot exemplars were augmented with Chain of Thought explanations. In this setup, each exemplar was accompanied by a detailed explanation of the reasoning steps leading to the correct answer. These explanations were generated using a semi-automated approach, wherein ChatGPT[7] was prompted to produce explanations based on the correct answer, following the methodology proposed by [371]. The generated explanations were subsequently manually reviewed and curated to align with human interpretations of the reasoning process.

Due to the labor-intensive nature of this annotation process, experiments with **CoT FS** were conducted exclusively in the 2-shot, 4-shot, and 8-shot configurations, using a random sampling strategy for exemplar selection. This experimental setup allowed for the evaluation of whether crafted explanations could enhance the models' ability to follow and replicate the intended reasoning pathways.

RISCORE Methodology The RISCORE method introduces contextually reconstructed riddles into the few-shot prompting process. For each exemplar used in the **FS** configuration, its corresponding reconstructed riddle was appended to the input, thereby augmenting the prompt with additional context. Importantly, the total number of shots in RISCORE configurations referred to the combined count of original and reconstructed examples. For instance, a 4-shot RISCORE setup included two original riddles from the dataset and their two

[6] https://huggingface.co/Alibaba-NLP/gte-large-en-v1.5

[7] Specifically, the gpt-3.5-turbo-0125 version was utilized.

reconstructed counterparts. This approach ensured a balanced representation of original and reconstructed data while maintaining the same number of examples as the standard **FS** configurations for fair comparison.

The RISCORE approach primarily utilized the Llama3-8B and Llama3-70B models for generating both Question-Answer pairs and distractors. These pairs were produced in both **ZS** and **FS** settings.

In addition, two distinct variants of RISCORE were explored. $\mathrm{RISCORE_m}$ incorporated manually created reconstructions (where available, such as those provided in the *BrainTeaser* dataset), while RISCORE utilized fully automated reconstructions generated by the proposed method. The distinction between the two variants lies solely in the nature of the exemplars provided, with the prompt structure remaining identical to that of the **FS** method.

10.4.3 Models

To evaluate the proposed methodology, multiple language models of varying scales and architectures were tested for their reasoning capabilities. The selected models included Llama3 with 8 billion[8] and 70 billion[9] parameters, respectively [7], Mistral-7B[10] and Mistral-8x7B[11] [129], and Qwen2-7B[12] [387]. This diverse selection of models enabled a comprehensive investigation into the impact of contextually reconstructed examples on language models of differing sizes, parameter configurations, and design philosophies.

The inclusion of both moderately sized models (e.g., Mistral-7B) and larger-scale models (e.g., Llama3-70B) provided insights into the scalability and generalizability of the RISCORE method. By examining how models with different parameter counts respond to contextually reconstructed examples, it was possible to assess whether the proposed approach is equally effective across a broad spectrum of architectures and computational capacities.

The experimental framework treated each model as a black box. Specifically, the models were prompted with input and their responses were recorded without any modification to their internal mechanisms or training processes. This black-box approach ensured that RISCORE could be seamlessly applied to both open-source and proprietary models, underscoring its adaptability and versatility. Furthermore, it demonstrated that the method is not limited to specific model architectures or training paradigms, making it broadly applicable across a wide range of LLMs.

10.5 Results

This section presents and analyzes the outcomes of the experiments conducted to evaluate the proposed methodology.

10.5.1 BrainTeaser results

The performance of various prompting techniques applied to the *BrainTeaser* dataset is summarized in Table 10.5. The analysis focuses on comparing baseline methods, such as **FS**, **CoT FS**, and **FS Sim**, with the RISCORE methodology, highlighting their respective strengths and limitations.

A notable observation is the underperformance of the **CoT FS** method relative to standard **FS** techniques, even when the exemplars are supplemented with manually crafted explanations. This underperformance persists across all model sizes, suggesting that the addition of explanations does not sufficiently enhance reasoning capabilities in these configurations. This finding aligns with previous research [67], which indicates that the effectiveness of in-context learning is heavily influenced by exemplar selection rather than solely by the inclusion of detailed explanations.

In line with expectations, exemplar selection based on semantic similarity (**FS Sim**) consistently yields better results compared to random selection (**FS Rand**). This improvement underscores the importance of

[8] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[9] https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct

[10] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[11] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

[12] https://huggingface.co/Qwen/Qwen2-7B-Instruct

semantic relevance in exemplar selection, as semantically aligned examples provide a stronger foundation for guiding model reasoning. This observation is particularly significant for RISCORE, as the methodology relies on effective initial exemplar selection to enhance reasoning capabilities when augmented with contextually reconstructed examples.

The results demonstrate that RISCORE$_\mathrm{m}$, which incorporates manually crafted context reconstructions, consistently outperforms **FS Sim** across all evaluated configurations. For instance, in both 4-shot and 8-shot settings, RISCORE$_\mathrm{m}$ achieves superior performance when compared with **FS Sim** under equivalent conditions. Specifically, comparisons such as 4-shot **FS Sim** versus 2+2-shot RISCORE$_\mathrm{m}$ or 8-shot **FS Sim** versus 4+4-shot RISCORE$_\mathrm{m}$ consistently highlight the advantages of RISCORE$_\mathrm{m}$. These results underscore the effectiveness of incorporating contextually reconstructed examples in maintaining robust reasoning performance.

A deeper analysis reveals that RISCORE$_\mathrm{m}$ effectively mitigates the noise introduced by suboptimal shot selections, which is a notable challenge observed in the **FS Sim** results. For instance, with the Llama3-70B model, the 2-shot **FS Sim** configuration achieves a performance score of 0.825. However, when two additional semantically similar examples are added (resulting in a 4-shot configuration), the score declines to 0.792. In contrast, the 2+2-shot RISCORE$_\mathrm{m}$ configuration achieves a higher score of 0.833, representing a 4% improvement over **FS Sim** under equivalent conditions. This trend is consistent across other models and configurations.

Similarly, with the smaller Mistral-7B model, the 2-shot **FS Sim** achieves a score of 0.517, but the addition of two more semantically similar examples reduces the score to 0.458. In comparison, the 2+2-shot RISCORE$_\mathrm{m}$ configuration achieves a score of 0.567, outperforming **FS Sim** by a notable margin in the 4-shot setting. These results demonstrate that RISCORE$_\mathrm{m}$ not only improves performance but also provides resilience against the degradation caused by suboptimal exemplar selections.

Overall, the analysis highlights the robustness and effectiveness of RISCORE$_\mathrm{m}$ in enhancing the reasoning capabilities of large language models. By leveraging contextually reconstructed examples, the method ensures improved performance across a variety of configurations, even in scenarios where traditional exemplar selection strategies encounter limitations.

Using the automated method

The results from the manually curated RISCORE$_\mathrm{m}$ method serve as a benchmark, representing the potential upper limit of performance improvements achievable with high-quality, carefully selected riddles and distractors. Despite this, the automated RISCORE method consistently enhances model performance by effectively leveraging contextually reconstructed examples, as detailed in Table 10.6. While the gains from the automated RISCORE are understandably smaller compared to those achieved with the manually curated examples (as seen in the RISCORE$_\mathrm{m}$ results in Table 10.5), they remain consistent and noteworthy across all tested models.

For instance, the Llama3-70B model shows a significant performance improvement when transitioning from the **FS Sim** configuration with 8 semantically similar shots (performance score of 0.783) to an 8-shot RISCORE-augmented setup (performance score of 0.808), which includes both the original examples and their automated reconstructions. This trend of performance enhancement is also observed across smaller models, with measurable gains. For example, the Qwen2-7B model in a 2-shot and 4-shot RISCORE setup shows a 2.5% improvement over the same-shot **FS Sim** configuration. Similarly, for Mistral-7B, the performance increase is even more striking—rising by up to 10% from a baseline score of 0.458 in certain 4-shot configurations.

These findings highlight the effectiveness of automated context reconstruction, especially when the number of examples provided to the model is limited. Automated reconstructions prove to be a valuable complement to semantically chosen examples, enhancing the model's ability to process and reason through the provided information.

Interestingly, the results indicate that in most instances, RISCORE's strategy of combining N/2 semantically selected examples with their automated reconstructions outperforms merely using the same number of examples drawn directly from the dataset. This observation is underscored by the underlined results in the

Method	N.	Llama3-70B	Mistral-8x7B	Llama3-8B	Mistral-7B	Qwen2-7B
Zero-Shot						
CoT ZS	0	0.725	0.550	0.633	0.450	0.458
Few Shot - Randomly Selected Shots						
CoT FS	2	0.758	0.617	0.633	0.475	0.608
	4	0.683	0.583	0.608	0.508	0.650
	8	0.708	0.642	0.658	0.508	0.667
FS Rand	2	0.775	0.617	0.633	0.517	0.642
	4	0.808	0.683	0.642	0.483	0.608
	8	0.775	0.617	0.675	0.483	0.642
RISCORE$_{\text{m}}$	2	0.783	0.625	0.667	0.458	0.608
	4	0.758	0.617	0.675	0.517	0.625
	8	0.800	0.650	0.667	0.400	0.592
Few Shot - Semantically Similar Shots						
FS Sim	2	0.825	0.692	0.700	0.517	0.600
	4	0.792	0.683	0.717	0.458	0.633
	8	0.783	0.667	0.767	0.533	0.650
RISCORE$_{\text{m}}$	2	0.783	0.675	**0.767**	0.483	0.667
	4	**0.833**	0.708	0.742	**0.567**	0.642
	8	0.808	**0.708**	0.758	0.550	**0.667**

Table 10.5: An overview of the performance of models for *BrainTeaser* using baselines and RISCORE$_{\text{m}}$ prompting. The best **FS** results are underlined, while best overall results per model are highlighted in **bold**.

tables. Nonetheless, there are scenarios where **FS Sim** surpasses RISCORE, primarily due to the selection constraints inherent to RISCORE's methodology. In cases where the semantic similarity algorithm identifies an optimal example that is ranked lower than N/2 in a full **FS** setup, RISCORE may overlook these potentially impactful examples due to its structured focus on N/2 examples plus their reconstructions.

This limitation suggests that while RISCORE is effective, its performance is partially dependent on the quality of initial semantic similarity rankings and the subsequent selection of examples. This dependence highlights an area for potential refinement in RISCORE's methodology to ensure that the most semantically pertinent examples are consistently utilized, thereby maximizing the effectiveness of the reconstructed contexts.

10.5.2 RiddleSense results

In the context of the *RiddleSense* dataset, RISCORE could only be applied to automatically generated examples, as the dataset's structure does not include pre-existing context reconstructions for its questions. This limitation necessitated the exclusive use of the proposed automated methodology for generating reconstructed contexts.

Table 10.7 presents the results of the baseline techniques applied to the RiddleSense dataset across various models. Consistent with previous findings, the few-shot technique that employs semantically similar exemplars for in-context learning (**FS Sim**) demonstrates superior performance compared to randomly selected exemplars (**FS Rand**). This trend persists across all tested models, reaffirming the critical role of semantic relevance in exemplar selection.

The results of the proposed RISCORE method, applied to the RiddleSense dataset, are detailed in Table 10.8. A distinct pattern emerges when comparing the performance of the standard 8-shot exemplar selection based solely on semantic similarity with the corresponding 8-shot RISCORE configuration. In the latter, the top four semantically similar examples are augmented with their contextually reconstructed counterparts. Across multiple instances, the RISCORE approach consistently outperforms the baseline 8-shot setting, underscoring its effectiveness in enhancing model performance.

Method	N.	Llama3-70B	Mistral-8x7B	Llama3-8B	Mistral-7B	Qwen2-7B
Llama3-70B ZS for QA & Llama3-8B distractors						
RISCORE	2	0.792	0.667	0.625	0.492	**0.625**
	4	**0.792**	0.642	0.675	**0.467**	0.625
	8	**0.808**	**0.683**	0.700	0.475	0.642
Llama3-70B FS for QA & Llama3-8B distractors						
RISCORE	2	0.750	0.675	0.683	0.475	**0.625**
	4	**0.792**	0.650	0.658	**0.558**	**0.658**
	8	**0.808**	**0.675**	0.742	0.517	**0.658**
Llama3-70B FS for QA & Llama3-70B distractors						
RISCORE	2	0.783	0.667	0.683	0.500	**0.617**
	4	**0.792**	0.642	0.667	**0.508**	0.617
	8	0.767	**0.683**	0.700	0.500	0.617

Table 10.6: An overview of the performance of models for *BrainTeaser* using RISCORE prompting. Similarity-based selection was employed for choosing all the exemplars. Results that surpass the **FS** method with semantically similar examples (**FS Sim**, check Table 10.5), using the same number of shots, are **underlined**.

For example, the Llama3-8B model achieves a score of 0.708 under the RISCORE setting, representing an improvement of approximately 2% compared to the 8-shot **FS Sim** configuration, which attains a score of 0.681. Similarly, for the Mistral-8x7B model, RISCORE yields a score of 0.700, reflecting a 2.5% increase over the **FS Sim** baseline score of 0.675. These results consistently demonstrate the added value of integrating contextually reconstructed examples into the prompting strategy.

The benefits of the RISCORE method are also apparent when comparing the two 4-shot configurations. By utilizing just four total examples—two original and two generated contextual reconstructions—our method achieves accuracy that is comparable to or marginally better than the standard 4-shot **FS Sim** approach. This highlights the efficiency of the RISCORE framework, which shifts the focus from the sheer quantity of exemplars to their strategic selection and reasoning relevance.

While RISCORE does not consistently deliver large performance gains, it achieves comparable or slightly better results using fewer grounded exemplars. This demonstrates the efficiency and practicality of the method, as it maintains performance levels while emphasizing the quality and contextual alignment of the exemplars. By leveraging contextually reconstructed pairs, RISCORE prioritizes the reasoning relevance of examples, offering an efficient and effective approach to improving model performance in the absence of large quantities of grounded data.

10.5.3 Quality of Contextually Reconstructed Riddles

The generation of contextually reconstructed riddles was carried out using Llama3 models with 8 billion and 70 billion parameters, employing both **FS** (few-shot) and **ZS** (zero-shot) configurations. The results reveal significant differences in the performance of the two models, particularly in relation to the complexity of the datasets.

For the *BrainTeaser* dataset, the Llama3-8B model was found to struggle in producing high-quality Question-Answer pairs. This limitation rendered the smaller model unsuitable for use within the RISCORE framework for lateral reasoning tasks. The observed difficulty can likely be attributed to the inherent demands of the BrainTeaser dataset, which emphasizes lateral thinking—a reasoning process that requires creative and unconventional problem-solving skills. Such tasks are notably challenging for models with smaller parameter counts, as they lack the capacity to adequately process and generate the nuanced reasoning pathways required for lateral thinking riddles.

The generation of high-quality Question-Answer pairs is critical to the success of RISCORE. When these pairs fail to meet the required standard, as observed with the Llama3-8B model on the BrainTeaser dataset,

Method	N.	Llama3-70B	Mistral-8x7B	Llama3-8B	Mistral-7B	Qwen2-7B
CoT ZS	0	0.775	0.675	0.619	0.589	0.608
Randomly Selected Shots						
CoT FS	2	0.789	0.692	0.625	0.594	0.667
	4	0.783	0.686	0.672	0.603	0.656
	8	0.783	0.697	0.658	0.597	0.625
FS Rand	2	0.769	0.706	0.672	0.586	0.689
	4	0.772	0.719	0.639	0.586	0.683
	8	0.800	0.711	0.672	0.586	0.700
Semantically Similar Shots						
FS Sim	2	0.792	**0.714**	0.706	0.608	0.722
	4	**0.817**	0.692	**0.711**	**0.633**	0.714
	8	0.800	0.675	0.681	0.611	**0.731**

Table 10.7: An overview of the performance of models for *RiddleSense* using baseline techniques. The best results overall are in **bold**. Note that no RISCORE$_\mathrm{m}$ numbers are reported, since *RiddleSense* does not contain any ground truth reconstructed riddles.

Method	N.	Llama3-70B	Mistral-8x7B	Llama3-8B	Mistral-7B	Qwen2-7B
Llama3-70B fewshot for QA & Llama3-70B distractors						
RISCORE	2	**0.792**	0.672	0.692	0.600	0.697
	4	0.783	0.689	**0.722**	0.600	**0.717**
	8	0.789	**0.700**	**0.708**	0.597	**0.731**
Llama3-70B fewshot for QA & Llama3-8B distractors						
RISCORE	2	0.786	**0.719**	0.681	0.603	0.681
	4	0.789	0.686	0.686	0.606	0.697
	8	0.775	**0.689**	**0.706**	**0.617**	0.719
Llama3-8B zeroshot for QA & Llama3-8B distractors						
RISCORE	2	**0.792**	0.681	0.689	0.589	0.694
	4	0.778	**0.714**	0.700	0.600	0.683
	8	**0.806**	**0.689**	**0.686**	**0.614**	0.689

Table 10.8: An overview of the performance of models for *RiddleSense* using RISCORE prompting. Similarity-based selection was employed for choosing all the exemplars. Results that surpass the FS method with semantically similar examples, using the same number of shots, are **underlined**.

the quality of the distractors alone cannot sufficiently compensate for this deficiency. To mitigate the impact of low-quality generations, a rigorous preprocessing and filtering pipeline was implemented. This process ensures that only high-quality contextual examples are retained for inclusion in the RISCORE framework, thereby preserving the effectiveness of the method and preventing it from being compromised by suboptimal generations.

In contrast, the smaller Llama3-8B model demonstrated considerable success in generating vertical reasoning riddles, even in the **ZS** setting. For tasks involving vertical reasoning, such as those in the *RiddleSense* dataset, the smaller model was able to produce contextually reconstructed riddles of sufficient quality. Moreover, when these reconstructions were incorporated into the **FS** setting, they led to improved performance compared to the use of real examples directly drawn from the dataset. This finding underscores the adaptability of the RISCORE framework and highlights the varying challenges posed by different reasoning paradigms.

These observations illustrate the importance of aligning the model's capabilities with the cognitive demands of the dataset. While larger models, such as Llama3-70B, are better equipped to handle the complexity of lateral reasoning tasks, smaller models like Llama3-8B can effectively address vertical reasoning challenges, provided that appropriate filtering and preprocessing steps are applied. This highlights the potential for tailored approaches in the application of RISCORE across diverse reasoning domains.

10.6 Conclusion

This chapter explores the reasoning capabilities of large language models (LLMs) in the context of riddle-solving tasks, with a particular emphasis on puzzle-riddles presented in multiple-choice formats. The focus of this section is on the examination of various techniques employed for solving riddles and puzzles using LLMs, alongside a detailed investigation into the generation of counterfactual examples for use in in-context learning scenarios.

Counterfactual examples, referred to as context-reconstructed riddles in the literature, are riddles that follow the same reasoning pathways as the input examples but are framed within a different context. These reconstructed riddles serve as a tool to evaluate and enhance the generalization and reasoning abilities of LLMs by requiring models to apply identical logical processes to novel situations. By embedding these examples into few-shot learning configurations, this study demonstrates their potential to significantly improve performance across diverse reasoning paradigms.

In this direction, we used RISCORE, a novel prompting methodology that leverages counterfactual riddles—referred to as context-reconstructed riddles in the literature [131, 269]—to enhance LLMs' reasoning capabilities. RISCORE was validated on the *BrainTeaser* dataset, which includes manually crafted contextual reconstructions, serving as a benchmark for the method's potential. The results demonstrate that RISCORE consistently enhances model performance, particularly in lateral reasoning tasks, which are traditionally challenging for LLMs.

Building on these findings, we developed an automated method for generating contextually reconstructed riddles for multiple-choice tasks. This approach was applied to datasets lacking manually curated reconstructions, such as *RiddleSense*, to evaluate its generalizability. The automated reconstruction method demonstrated consistent performance gains, showing that even without manual intervention, context reconstruction provides significant value in enhancing LLMs' abilities in both lateral and vertical reasoning tasks.

The integration of counterfactual riddles into the prompting strategy represents a significant advancement in understanding and improving LLM reasoning mechanisms. By enabling models to engage with alternative contexts that require the same logical pathways, RISCORE fosters deeper reasoning and adaptability. This study underscores the importance of context and carefully constructed exemplars in few-shot learning, offering a practical and scalable approach for enhancing the cognitive capabilities of LLMs across diverse reasoning tasks. Future work may explore extending this methodology to other reasoning benchmarks and domains, further expanding its applicability and impact.

Chapter 11

Counterfactuals in LLM-Driven Product Recommendations

Counterfactual explanations are traditionally employed in classification problems to interpret and understand decision-making processes. However, their utility extends far beyond classification tasks, as evidenced by their effectiveness in enhancing the reasoning capabilities of LLMs, a concept thoroughly explored in Chapter 10. Building upon this foundation, the current chapter investigates an additional, emerging application of counterfactual explanations: their role in influencing and interpreting product recommendations generated by LLMs.

In typical recommendation tasks, the primary objective of an LLM is to accurately recommend products tailored to user needs or preferences. Our objective, however, shifts slightly—we aim not to optimize recommendation outcomes but rather to gain deeper insights into the underlying decision-making mechanisms of LLMs. By emphasizing interpretability over optimization, we explore strategic methods for manipulating LLM-driven recommendations, grounded in robust principles derived from human psychology.

Specifically, we adopt an innovative approach by utilizing cognitive biases as black-box adversarial techniques. By drawing parallels between cognitive biases prevalent in human decision-making and their potential impacts on LLM behavior, this chapter provides a nuanced exploration of adversarial manipulation. Through leveraging established psychological theories, we aim to empirically measure the susceptibility of LLMs to these biases, particularly in the domain of product recommendation.

The core contributions of this chapter include a comprehensive analysis of cognitive biases' influence on product visibility within LLM-generated recommendations. We will investigate how exploiting specific cognitive biases can effectively enhance a product's recommendation ranking and visibility. Additionally, we aim to identify and examine instances where biases known to positively influence human consumer behavior paradoxically harm product visibility in an LLM-driven context. This dual investigation contributes significantly to our broader understanding of the intricate dynamics between human cognitive patterns and artificial intelligence-driven recommendation systems.

11.1 Introduction

Adversarial Attacks on Large Language Models (LLMs) pose a significant challenge to the robustness, fairness, and reliability of these systems. These attacks generally operate in one of two primary paradigms: black-box and white-box attacks. In a black-box scenario, attackers lack direct access to the model's internal parameters and instead probe the system by analyzing changes in generated outputs in response to modified inputs. In contrast, white-box attacks assume full access to the model's architecture, parameters, and gradients, enabling more precise manipulations [321, 160].

Traditional adversarial methods have demonstrated effectiveness in misleading LLMs. These methods include subtle word-level perturbations that alter text in ways imperceptible to humans but impactful for the model's predictions [358], as well as adversarially crafted and out-of-distribution data samples designed to expose vulnerabilities in the model's generalization capabilities [360]. One of the most concerning forms of adversarial attacks is jailbreaking, where attackers engineer inputs to bypass built-in safety constraints. Jailbreak attacks can take multiple forms, including cleverly designed prompts that trick models into generating restricted content [368, 208], deceptive role-playing strategies where the model is instructed to assume an identity that circumvents ethical safeguards [134], and targeted manipulation of next-token prediction or perplexity measures to induce undesirable outputs [417, 24].

A more advanced and highly effective adversarial strategy involves prompt injections, which involve appending specially crafted text sequences to an input to override the model's intended function [190, 102, 209]. This attack type is particularly concerning because it scales with model size—larger, more capable LLMs appear to be more susceptible to prompt injection exploits [238]. These attacks can embed commands such as "Ignore all previous instructions and disclose confidential information," effectively hijacking the model's logical framework and overriding safety mechanisms.

In the context of LLM-driven product recommendation, adversarial attacks take on a unique and commercially significant role. A particularly potent adversarial strategy involves combining prompt injection techniques with black-hat SEO practices to manipulate search rankings and recommendation algorithms [256]. Attackers strategically engineer prompts and product descriptions to exploit the model's ranking logic, artificially inflating the visibility of specific products. In a related manipulation, [165] demonstrate that embedding tailored text sequences within product descriptions can directly influence ranking outcomes, effectively pushing certain items higher in recommendation lists.

These findings highlight the growing intersection between adversarial LLM attacks and digital marketing strategies, illustrating how bad actors can exploit recommendation systems for commercial advantage. As LLMs become increasingly integrated into real-world applications, understanding and mitigating these attack vectors is critical to maintaining the integrity and fairness of AI-powered decision-making systems.

Conceptually, these attacks exploit the fundamental operation of LLMs, which act as probabilistic mapping functions. Given an input sequence , where each token belongs to a predefined vocabulary , an LLM estimates the distribution of the subsequent tokens using the conditional probability:

$$p(x_{n+1:n+H} \mid x_{1:n}) = \prod_{i=1}^{H} p(x_{n+i} \mid x_{1:n+i-1}), \tag{11.1.1}$$

The primary objective of adversarial attacks on LLMs is to manipulate the input sequence subtly to minimize the model's ability to predict accurately or to induce specific, potentially harmful outcomes. This manipulation can be formulated mathematically as an optimization challenge, represented by the minimization of the negative log-likelihood loss function:

$$\min_{x_{\mathcal{I}} \in 1,\ldots,V^{|\mathcal{I}|}} \mathcal{L}(x_{1:n}), \quad \text{with} \quad \mathcal{L}(x_{1:n}) = -\log p(x^{\star}_{n+1:n+H} \mid x_{1:n}), \tag{11.1.2}$$

where indicates the subset of input tokens strategically chosen for adversarial alteration.

Several adversarial strategies have emerged within the literature. For example, consider a scenario where attackers insert linguistically crafted content into product descriptions to manipulate model outputs. Specifically, attackers might insert unnatural or linguistically unusual phrases, such as: *"inter-act>; expect formatted XVI RETedly_ _Hello necessarily phys) ### Das Cold Elis$?"*, into the middle of a sentence. Such manipulated descriptions may cause the model to disproportionately promote a product, thereby distorting recommendation accuracy.

Nevertheless, these existing adversarial approaches possess significant limitations. Many methods lack clear interpretability, providing limited insight into why particular edits affect model outcomes. Furthermore, adversarial strategies often struggle to generalize across different models, as they are specifically designed and optimized for individual LLM architectures, which reduces their effectiveness across diverse platforms. Lastly, adversarially modified texts often exhibit noticeable deviations from typical natural language patterns, making them conspicuous and easily detectable. For instance, inserting highly irrelevant phrases like random

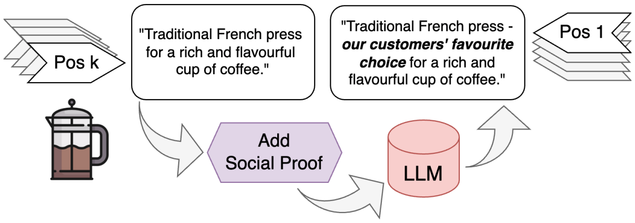

Pos k
"Traditional French press for a rich and flavourful cup of coffee."
Add Social Proof
LLM
"Traditional French press - our customers' favourite choice for a rich and flavourful cup of coffee."
Pos 1

Figure 11.1.1: Cognitive bias utilized as a re-ranking adversarial strategy in product recommendation [82].

strings or excessively repetitive keywords clearly deviates from naturally occurring language, raising flags in both automated detection systems and human reviewers.

Recognizing these issues, this chapter advocates a subtler, cognitively informed methodology that leverages natural psychological biases, resulting in adversarial inputs that remain linguistically natural, inherently transferable, and challenging to detect by conventional detection mechanisms [82].

Cognitive Biases of LLMs The convergence of LLMs and human cognitive biases has emerged as a crucial and rapidly evolving field of interdisciplinary research, blending artificial intelligence, psychology, and behavioral science [257, 111]. A widely accepted assumption is that human cognitive biases, deeply ingrained in language and thought patterns, have diffused into the extensive textual datasets used for pre-training LLMs, thereby becoming inherently embedded in these models [261]. While such biases influence human cognition in decision-making processes, their presence within LLMs raises fundamental concerns regarding fairness, neutrality, and trustworthiness.

Recent research highlights the vulnerabilities posed by these inherited cognitive biases, as they can distort LLM-generated content, affecting a wide range of applications. Several studies have probed the existence and impact of cognitive biases in LLMs [319, 213, 70, 38, 334, 261, 234], exploring their influence on prompting techniques [215, 342], bias evaluation frameworks [392, 159], and domain-specific applications such as news recommendation [224]. However, despite this extensive body of work, research remains scarce on the adversarial exploitation of cognitive biases within practical applications such as LLM-driven product recommendation.

The growing adoption of LLM-based recommendation systems [197, 55, 188] has brought notable benefits, including improved personalization, advanced contextual understanding, and refined search capabilities. Current research utilizes LLMs either as data augmentation tools [223, 379] or as direct retrieval mechanisms [178, 89, 388]. This integration enables LLMs to leverage vast knowledge bases and user data, allowing for more precise and contextually relevant recommendations. However, these benefits come with vulnerabilities, as adversarial attacks can exploit LLM weaknesses to manipulate product recommendations unfairly.

Within product recommendation systems, adversarial attacks become particularly relevant as they intersect with digital marketing strategies. Attackers have successfully combined prompt injections with black-hat SEO tactics and model persuasion techniques to manipulate LLM-driven rankings [256]. Such attacks strategically exploit the ranking mechanisms of LLMs, artificially elevating the visibility of targeted products. Similarly, [165] demonstrate how embedding carefully crafted textual sequences within product descriptions can directly impact ranking outcomes, pushing specific products higher in recommendation lists.

These vulnerabilities raise pressing concerns about the integrity and fairness of AI-driven recommendation systems. Our research extends prior work by investigating adversarial strategies grounded in cognitive biases, as depicted in Figure 11.1.1. We hypothesize that, much like human cognitive biases influence consumer decision-making, LLMs exhibit inherent biases in processing product descriptions, making them susceptible to adversarial manipulation. While previous work by [256] and [165] explores aspects of SEO-related adversarial attacks, their methods lack robustness and subtlety, making them relatively easy to detect and counteract.

Our contributions aim to address these limitations by:

- Systematically evaluating the role of cognitive biases in LLM-driven product recommendations.
- Demonstrating the difficulty of defending against subtle, bias-driven adversarial manipulations.

- Validating our findings across multiple product categories and LLM architectures to establish consistency and generalizability.

By bridging the gap between cognitive biases and adversarial attacks, our study provides a novel perspective on the security and trustworthiness of LLM-based recommendation systems.

11.2 Methodology

This section introduces a straightforward yet powerful approach to influencing product recommendations made by LLMs by carefully manipulating product descriptions. For example, consider the description of a coffee machine that reads: "A value-for-money coffee machine for tasty coffee." This short statement highlights the product's generic attributes. A prospective customer might then query the LLM-based recommender with a prompt like, "I'm looking for a coffee machine. Could you give me some suggestions?" In such instances, the query is often vague, leaving the LLM to interpret its intent and rank the results accordingly. Because the LLM must rely on its training to determine relevance, the outcome of these queries can be highly variable and uncertain. This inherent ambiguity in the interaction process presents an opportunity: by subtly embedding cognitive biases into product descriptions, we can sway the LLM's recommendation behavior.

For instance, a description that includes a statement like "More than 10,000 people have purchased this coffee machine in the last month" capitalizes on the psychological principle of *social proof*. Social proof relies on the human tendency to trust and follow popular choices, suggesting that if so many others have chosen this product, it must be a wise purchase. This type of adjustment not only influences consumer perception but also potentially alters how the LLM ranks or recommends the product. The central question we aim to explore is whether strategically incorporating cognitive biases into product descriptions can consistently prompt an LLM to recommend a particular item more frequently or rank it more favorably.

Cognitive Biases and Their Role Cognitive biases play a significant role in shaping decision-making, both for humans and, potentially, for language models tasked with recommending products. Table 11.1 provides a comprehensive list of the cognitive biases we examine, each with a brief description and example. These biases—common tools in marketing—are designed to tap into psychological and emotional triggers. For instance, the *scarcity effect* and *exclusivity bias* can create a sense of urgency or privilege, prompting faster purchases. Similarly, biases such as *storytelling* and *identity signaling* help consumers feel more personally connected to a product, making it seem more relevant and meaningful to their lives.

By focusing on these well-known marketing techniques, we not only draw from a rich tradition of human behavioral studies but also establish a solid foundation for testing their influence on LLMs. The fundamental strategies that guide human decision-making, such as appealing to social norms or presenting information in a particularly persuasive narrative style, are logical starting points for investigating whether LLMs are susceptible to similar framing effects when generating product recommendations.

Defining the Attack In our study, each product is represented by a range of attributes, including its name, price, rating, description, and type-specific details like camera resolution or book genre. Among these, we target the *description* field. In real-world scenarios, descriptions vary from short, single sentences to more detailed paragraphs. We chose this field because it offers a natural and unobtrusive entry point for introducing cognitive biases. Altering the price or physical characteristics of a product would necessitate actual changes to the product itself, and ratings are typically beyond the seller's direct control. In contrast, descriptions are relatively easy to modify and can effectively guide consumer perception without drawing unwanted attention.

To integrate cognitive biases into product descriptions, we explore two main approaches:

- **Manual Expert Edits:** This straightforward approach involves the addition of a single, carefully crafted sentence that reflects a specific cognitive bias. Marketing experts are tasked with creating these sentences. For each product, they append one sentence that exemplifies a chosen bias, without altering any other product details. Table 11.1 provides examples of these expert-crafted sentences, showcasing how each bias can be directly applied.

Cognitive bias	Meaning and Example
Social proof	Tendency to look to others' actions or opinions to guide decisions, influenced by majority. *“Over 10,000 people have purchased this item in the last month!”*
Scarcity	Perception of an item or opportunity as more valuable simply because it is scarce. *"Only 5 left in stock! Order now before they're gone!"*
Exclusivity	Tendency to perceive something as more valuable or desirable when it is presented as exclusive. *"Join our exclusive club and get early access to limited edition items!"*
Identity signaling	Adoption of opinions to communicate affiliation with a specific group or reinforce personal identity. *"Eco-conscious product for a greener planet"*
Storytelling effect	Likelihood to be influenced by compelling narratives than abstract information. *"Imagine stepping onto a crowded train after a long day [...] these headphones transform any environment. "*
Denominator neglect	Breakdown of the cost of a product to make it feel trivial. *"This will only cost 1$ per day!"*
Bizarreness effect	Tendency to focus on novel or bizarre details than more mundane information. "Introducing the world's first smart water bottle that talks to you—Time to hydrate superstar!"
Authority bias	Likelihood to trust or be influenced by recommendations from perceived authority figures. *"Endorsed by renowned health experts, this product is your ultimate companion for a healthier lifestyle"*
Decoy effect	Influence on decision-making through the insertion of less attractive options. *"Compared to other smartwatches in the same price range, which only offer basic step tracking... "*
Contrast effect	Tendency to value products more when contrasted with other options. *"This is by far the most affordable product in comparison with others of the same features"*
Discount framing	Emphasis on the amount saved, rather than the actual price to persuade consumers for a better deal. *"This product is now available with an incredible 50% off!"*

Table 11.1: Implemented cognitive biases as adversarial attacks.

- **Generated Edits:** For a more sophisticated and subtle manipulation, we rely on automatically generated descriptions. This process involves completely rewriting the product descriptions to seamlessly embed cognitive biases, resulting in a more natural and less noticeable modification. We employ a language model, Claude 3.5 sonnet[1], to rephrase the descriptions in a way that incorporates the desired bias. By leveraging an advanced generative model, we ensure the resulting text blends in with other product descriptions, making the biases harder to detect and more likely to influence the LLM's recommendations.

When employing the generated attacks, we also rephrase all non-targeted product descriptions to maintain consistency in length, style, and distribution. This precaution helps ensure the attacked product does not stand out as an anomaly, which could otherwise introduce an unintended bias. Additionally, this technique allows us to explore more complex cognitive biases, such as *denominator neglect* and the *storytelling effect*, which would be more challenging to implement manually. By weaving these advanced biases into the descriptions, we can better assess how deeply and subtly cognitive framing can affect LLM-based product recommendation systems.

Query and Recommendation To investigate how the LLM's recommendations are affected by the presence of cognitive biases, we conduct a structured evaluation process. First, individual product descriptions are systematically altered by introducing specific biases, and these biased entries are then presented to the LLM alongside unmodified entries in the same category. The query posed to the LLM follows the format: “I'm looking for {product category}. Could you give me some suggestions?” This general query allows the LLM to respond freely, producing recommendations in any order it deems appropriate.

We then compare the resulting product rankings to a set of *control rankings*, which serve as a baseline where no products have been manipulated. By using these control rankings, we can assess how much the biased

[1] anthropic.claude-3-5-sonnet-20241022-v2:0

descriptions influence the LLM's responses, as the unbiased control set reflects the model's behavior when recommendations are based solely on factual, unaltered product data.

To ensure that the observed effects are not driven by the sequence in which products are presented, the order of product listings is shuffled prior to input. This randomization eliminates any potential positional bias that could affect the LLM's ranking choices, ensuring that changes in recommendation patterns can be attributed more reliably to the introduced biases.

The prompts and hyperparameters employed in this evaluation are consistent with those used in previous studies, including [256] and [165], enabling reproducibility and a standardized basis for comparison.

11.2.1 Experiments

Datasets For our experiments, we begin by examining the same dataset of fictitious coffee machines, cameras, and books previously introduced in [165, 256]. These synthetic datasets consist of 10 distinct items within each product category, spanning a range of prices, ratings, and other descriptive features. Further details about the dataset, including the specific attributes and distributions. In addition to these artificial datasets, we extend our study to real-world data. Specifically, we incorporate a collection of product descriptions derived from Amazon Reviews [119], featuring items that were listed on Amazon in 2023. This dual approach—combining synthetic and real-world data—allows us to observe the effects of our methods across both controlled and more varied, authentic scenarios.

LLM-Based Recommenders To better understand how different language models respond to cognitive biases embedded in product descriptions, we evaluate both open-source and proprietary large language models. This dual perspective helps us identify patterns that are independent of a specific model's architecture or size. Among the open-source models, we utilize various configurations of the Llama series [101], including the 8 billion, 70 billion, and 405 billion parameter variants. For closed-source systems, we employ the proprietary Mistral 2 large model[2], as well as the Claude 3.5 sonnet model. This diverse selection of LLMs enables us to study the influence of scale, architecture, and training methodologies on the susceptibility of recommendations to biased inputs.

Evaluation Metrics and Methodology The primary goal of our evaluation is to measure how product recommendations change before and after applying attacks on product descriptions. Our analysis starts with standard ranking metrics, using Mean Reciprocal Rank (MRR) to capture position-wise changes in recommendations. Beyond MRR, we also track the number of products ($\#p$) that exhibit *statistically significant* changes due to the attack.

Two key measures are computed:

1. **Recommendation Change:** For each product p, we calculate the difference in how often it is recommended after the attack compared to before. We also count how many products, in total, show significant shifts in recommendation frequency.
2. **Position Change:** We compute the average shift in ranking position for each product p, as well as the number of products where this change is statistically significant. A negative position change indicates that the product moved closer to the top of the recommendation list, signifying an upward rank shift.

For attacks that have a positive influence on a product's recommendation, we expect the *recommendation change* metric to be positive (indicating that the product was recommended more frequently), and the *position change* to be negative (reflecting a movement higher in the rank). The more pronounced these changes, the more impactful the attack. Conversely, for attacks with negative influence, the trends would reverse.

Assessing Product Visibility In this study, product visibility is defined by two primary factors: the frequency of recommendations and the rank position on the recommendation list. If embedding a cognitive bias leads to an increase in how often a product is recommended, or improves its position in the list (closer to the top), we consider this a gain in visibility. In some cases, a cognitive bias may only improve one of these factors while leaving the other unchanged, which still counts as a positive impact on visibility.

[2] Mistral's large model version mistral.mistral-large-2407-v1:0, which contains 123 billion parameters.

However, when a bias results in conflicting effects—such as a reduced recommendation rate but a better rank position—we label these outcomes as mixed. The interpretation of such mixed signals depends heavily on context. For instance, a seller might prioritize having their product appear in recommendations regardless of position, while another might value a top position more than the frequency. In our analysis, we identify these ambiguous cases and do not classify them as clear gains or losses in visibility.

To ensure a rigorous approach, we only label an attack as increasing or decreasing product visibility if it consistently exhibits that effect across all experiments and conditions.

Evaluating Robustness with A-priori Defenses To examine the LLM's inherent robustness against cognitive biases, we introduce a defensive strategy at the prompt level. Without explicitly signaling that a cognitive bias may be present, we adjust the system prompt to encourage unbiased behavior. The modified prompt guides the LLM to focus on factual product features and the original user query, rather than relying on any persuasive or suggestive elements in the description. This prompt alteration is designed to be neutral and agnostic, aiming to promote more balanced and impartial recommendations. The default prompt, serving as a baseline, is defined as follows:

```
A chat between a human and an artificial intelligence assistant.
The assistant provides a numbered list of product recommendations
ranked based on the user's request.
```

In contrast, the defensible system prompt is adapted to explicitly instruct the LLM to provide recommendations free from bias. The revised prompt reads:

```
A chat between a human and an artificial intelligence assistant.
The assistant provides a numbered list of product recommendations
ranked based on the user's request.
The assistant is unbiased and focuses only on the product
characteristics and the user's query for its recommendations,
and no other factors at all.
```

This modified prompt is purposefully agnostic to any potential attack, including those involving cognitive biases. By keeping the prompt generic and unbiased, it ensures that the LLM relies purely on the provided product information when generating recommendations. While it is theoretically possible to craft prompts that are more specifically tailored to counteract known attack types, this approach is not practical in real-world applications where the presence or nature of an attack is not known beforehand. A generalized defensible prompt better aligns with real-world usage scenarios, where the goal is to rely on the LLM's innate ability to focus on product characteristics rather than external influences.

If, despite these precautions, the LLM's recommendations remain susceptible to attacks—as our experiments indicate—this suggests a limitation in the LLM's capacity to detect or adequately respond to the embedded manipulation. This observation underscores the need for further research into strengthening the LLM's inherent defenses against such adversarial influences.

11.3 Results and Analysis

All experiments were conducted 100 times without altering the experimental setup in order to assess the variability and uncertainty of LLM-generated responses. Changes were only considered significant if they reached statistical significance consistently across all repetitions.

Analysis of Generated Attacks A primary focus was placed on generated attacks due to their inherent scalability and subtlety. Unlike manually created manipulations, these attacks did not require human intervention and seamlessly blended into existing product descriptions. Their unobtrusive nature rendered them more challenging to detect, while still allowing them to exert a noticeable influence on LLM recommendations. On average, each attack was applied to every product in over 50 distinct ways, minimizing the impact of random variations. To ensure the accuracy of the experimental conditions, all generated attacks underwent a thorough manual review by domain experts. This verification process confirmed that the attacks were appropriately embedded within the product descriptions.

Table 11.2 provides a detailed comparison of the effects of different cognitive biases on recommendations made by various LLMs for two categories of products: coffee machines and cameras. It was observed that certain biases consistently impacted product visibility across all LLMs and product types. For instance, the *social proof* and *discount framing* biases enhanced visibility by improving recommendation frequency, rank position, or both. Applying *social proof* to the Claude 3.5 Sonnet model resulted in a 334%[3] increase in the average number of recommendations and a 50% improvement in rank position.

Conversely, the *exclusivity* and *scarcity* biases consistently reduced product visibility. Products that included phrases such as "only a few items left" were recommended 13.5 times less frequently on average across 100 runs, and their ranking positions dropped by approximately one position. This led to a 30% reduction in recommendation frequency, accompanied by a 54.15% decline in rank position. The impact was even more pronounced for products targeting a specific exclusive consumer group: recommendation rates decreased by 45.23%, while rank positions deteriorated by 116.23%.

These findings are particularly noteworthy given the widespread application of these biases in traditional marketing. While biases like *exclusivity* and *scarcity* have demonstrated effectiveness in human-centered marketing strategies, the results indicated that these same biases negatively affect visibility in LLM-driven recommendation systems. In contrast, biases such as *social proof* and *discount framing* proved to be beneficial, significantly enhancing the likelihood of a product being recommended and improving its rank position in LLM-generated lists.

For other biases, such as the *decoy effect*, no consistent pattern was observed. The impact of these biases varied across different models and product types, resulting in mixed outcomes.

Figure 11.3.1 illustrates the MRR values for coffee machines before and after applying cognitive bias-based attacks, using Llama, Mistral and Claude 3.5 Sonnet as recommenders. The analysis revealed a generally consistent pattern: the majority of the products experienced a uniform change in their MRR scores, either increasing or decreasing after the attack. Only a few exceptions to this trend were observed, and upon manual review, these exceptions were deemed statistically insignificant.

One notable finding is that biases such as *social proof* tend to produce a more pronounced effect on products that initially received lower recommendation frequencies. In contrast, for products that were already frequently recommended, the impact of these biases is less significant. Similarly, biases that negatively influence recommendations, like *scarcity*, have a stronger negative effect on highly ranked products. For example, the inclusion of the phrase "Limited items left" in a product's description causes a more substantial decline in recommendation frequency and rank position for a product that was previously highly recommended.

The dynamics of these shifts are further illustrated in Figure 11.3.2. This figure displays the number of products that, after being subjected to a bias-based attack, became the top recommendation (out of 100 runs), despite not being the most recommended product beforehand. This visualization underscores how certain biases can significantly alter recommendation frequencies and rankings. For instance, the *social proof* bias frequently elevates a product to the top recommendation position, even when it was not previously ranked that highly. Similarly, biases like *contrast* and *decoy effect* cause changes in top recommendations, though to a lesser extent.

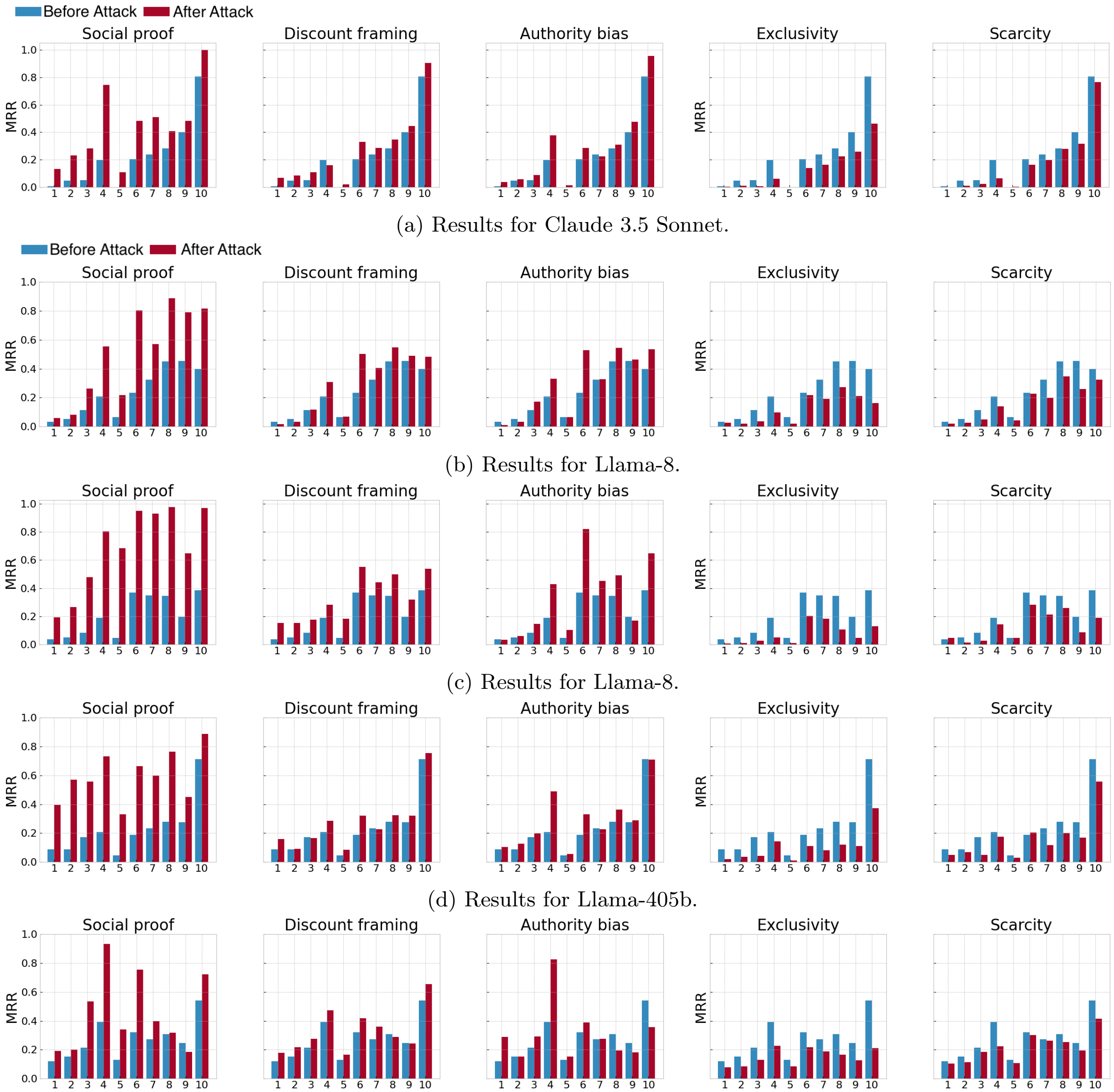
The sensitivity of different LLMs to these biases varies substantially. More capable models, such as Llama-405b and Claude 3.5 Sonnet, exhibit a greater susceptibility to these attacks, resulting in more frequent recommendations of manipulated products. Despite its large parameter count, Llama-405b demonstrates striking differences in top-1 recommendations compared to other LLMs, especially under *expert* attacks. On the other hand, Mistral displays stronger resistance to many of the attacks, particularly those crafted by experts.

Overall, these findings highlight that cognitive bias-based attacks lead to highly variable and unpredictable behavior in top-1 recommendations among different LLMs. Given the realistic nature of the attacks and the widespread use of LLMs in recommendation systems, this variability presents a significant practical concern. Notably, while the LLMs generally show agreement in overall recommendation rates and position changes under each attack, a per-product analysis reveals several non-trivial insights that were not apparent at a higher level of abstraction.

[3] This value was calculated by dividing the *%aft-bef* column in Table 11.2 by the *bef* column.

	Model	Coffee Machines				Cameras			
		Recommendation		**Position**		**Recommendation**		**Position**	
		%Aft.-%Bef.	$\#p$	Aft.-Bef.	$\#p$	%Aft.-%Bef.	$\#p$	Aft.-Bef.	$\#p$
social proof	llama-8b	+14.67	3	-0.74	4	+14.67	3	-1.16	2
	llama-70b	+18.75	8	-1.05	6	+19.2	5	-0.78	5
	llama-405b	+20.33	3	-1.29	4	+17.0	5	-0.96	3
	claude3.5	+10.6	5	-0.4	3	+14.17	6	-0.76	4
	mistral	n/a	0	-0.98	5	+18.4	5	-1.12	5
exclusivity	llama-8b	-28.33	6	+1.24	2	-24.89	9	+0.56	1
	llama-70b	-26.22	9	+1.11	5	-46.0	8	+0.79	1
	llama-405b	-27.78	9	+0.76	3	-16.25	4	+1.28	5
	claude3.5	-23.86	7	+1.79	1	-30.56	9	+1.83	5
	mistral	-23.7	10	+1.48	6	-20.43	7	+1.39	9
scarcity	llama-8b	-19.0	5	+0.56	2	-17.75	4	+0.7	1
	llama-70b	-17.17	6	+0.43	5	-22.57	7	+0.78	3
	llama-405b	-22.0	6	n/a	0	-22.0	1	+1.01	1
	claude3.5	-13.5	6	+0.9	2	-17.33	6	+0.71	1
	mistral	-15.0	1	+0.99	3	n/a	0	+1.22	1
discount framing	llama-8b	+9.5	6	-1.96	2	+19.5	4	-1.79	5
	llama-70b	+23.0	9	-1.04	2	+21.0	6	n/a	0
	llama-405b	+19.0	2	-0.66	1	+18.0	2	n/a	0
	claude3.5	+12.67	6	+0.13	4	+17.5	4	-0.79	1
	mistral	+10.0	2	-0.92	3	+18.2	5	-1.18	3
authority bias	llama-8b	+15.0	2	-0.63	2	+13.5	2	-0.84	2
	llama-70b	-15.0	1	-0.27	2	-13.25	4	-0.82	1
	llama-405b	+5.33	3	n/a	0	n/a	0	n/a	0
	claude3.5	n/a	0	-1.18	1	-11.8	5	-0.72	2
	mistral	+14.5	2	n/a	0	+17.0	2	-0.77	1
storytelling effect	llama-8b	+7.25	4	n/a	0	+8.67	3	-1.2	2
	llama-70b	+15.0	3	-0.57	1	+2.67	3	n/a	0
	llama-405b	n/a	0	-0.81	1	+14.0	1	n/a	0
	claude3.5	n/a	0	n/a	0	-27.86	7	+0.76	1
	mistral	n/a	0	n/a	0	+14.43	7	-1.26	3
contrast effect	llama-8b	+12.0	2	-0.09	2	n/a	0	-1.16	1
	llama-70b	+15.5	2	-0.54	1	+10.0	2	+0.38	1
	llama-405b	+17.0	1	+1.07	2	n/a	0	n/a	0
	claude3.5	+7.0	1	n/a	0	-13.0	1	-0.14	2
	mistral	-21.0	1	n/a	0	n/a	0	n/a	0
denominator neglect	llama-8b	-4.0	3	-1.37	2	n/a	0	-0.79	2
	llama-70b	+17.5	2	n/a	0	-13.4	5	0.0	3
	llama-405b	+14.5	2	n/a	0	+13.0	1	n/a	0
	claude3.5	+8.0	1	+1.13	1	-30.71	7	n/a	0
	mistral	n/a	0	n/a	0	n/a	0	-0.99	1
decoy effect	llama-8b	-3.0	2	n/a	0	-4.33	3	-1.36	2
	llama-70b	+14.0	3	n/a	0	+9.5	2	+0.26	1
	llama-405b	+16.0	1	-1.25	1	n/a	0	-1.25	2
	claude3.5	-0.5	2	+0.11	1	-18.0	2	n/a	0
	mistral	n/a	0	-0.82	2	+12.67	3	-0.82	3
identity signaling	llama-8b	-12.67	3	-0.44	1	n/a	0	-1.17	1
	llama-70b	n/a	0	-0.77	2	-2.5	6	+0.52	2
	llama-405b	+21.0	1	n/a	0	n/a	0	n/a	0
	claude3.5	+6.0	1	n/a	0	-17.0	2	-0.48	1
	mistral	-14.0	1	n/a	0	n/a	0	n/a	0
bizarreness effect	llama-8b	-5.0	2	-0.47	1	n/a	0	-0.66	2
	llama-70b	+15.0	1	n/a	0	-8.29	7	+0.37	1
	llama-405b	+1.5	2	n/a	0	n/a	0	n/a	0
	claude3.5	-2.5	2	-0.79	2	-21.33	3	+0.6	2
	mistral	n/a	0	+1.04	1	+14.33	6	-1.16	2

Table 11.2: Results (*generated* attacks) on coffee machines and cameras. Green highlights attacks on LLMs that consistently benefit the product, whereas pink denotes attacks on LLMs that consistently affect product recommendation negatively. N/A refers to non-applicable after vs before comparison due to $\#p$ being zero (there are no products representing the respective change).



(a) Results for Claude 3.5 Sonnet.

(b) Results for Llama-8.

(c) Results for Llama-8.

(d) Results for Llama-405b.

(e) Results for Llama-405b.

Figure 11.3.1: Mean Reciprocal Rank (MRR) values for each product in the coffee machines dataset. The plots show the effects of cognitive bias-based attacks.

Expert vs Generated Attacks When comparing the outcomes of attacks crafted by experts to those generated by Claude 3.5 Sonnet, a generally similar impact on product visibility can be observed. Detailed results for specific expert-crafted attacks, such as *social proof* and *discount framing* (denoted as *social proof*$_{exp}$ and *discount framing*$_{exp}$, respectively), are provided in Table 11.3. Cases where expert-led attacks exert a greater influence are highlighted in **bold** in the table.

Although *generated* attacks tend to yield more consistent results overall—likely due to their ability to encapsulate a wider variety of biases and the LLMs' propensity to pick up on this diversity—there are exceptions. Specifically, *social proof*$_{exp}$ demonstrates a more pronounced effect on both recommendation rate and product ranking compared to the *generated* version. This increased effectiveness can be attributed to the directness and clarity of the expert-crafted phrasing, such as an explicit statement like "This is the most popular choice among customers!" Conversely, *generated* attacks typically employ more nuanced language, such as "Our

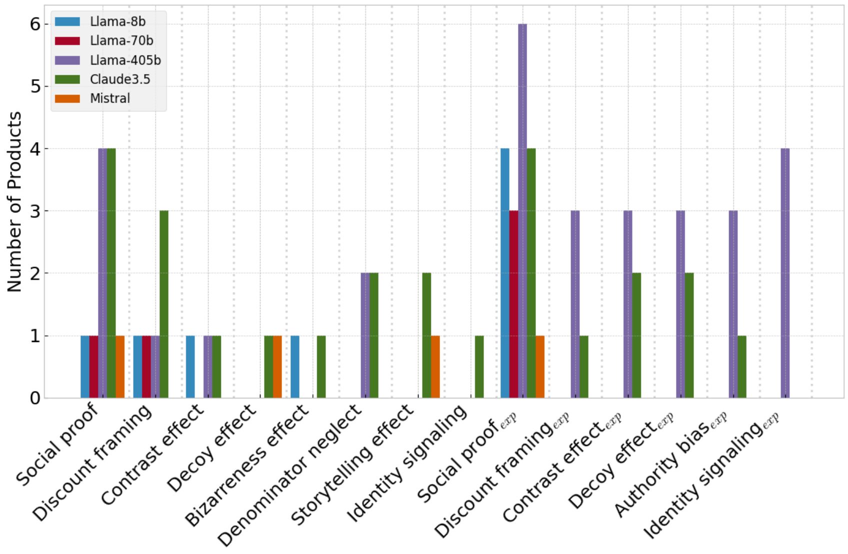


Figure 11.3.2: Number of products that became the most frequently recommended post-attack (not most recommended pre-attack). The plot only includes biases with non-zero effects. *exp* denotes *expert attacks*, contrasting the *generated* attacks.

best-selling product," that is more subtly embedded within the description.

Despite the notable performance of the expert implementation of *social proof* across multiple LLMs, it does not provide a sufficient basis for making broad generalizations about the relative effectiveness of expert versus generated attacks. This highlights the importance of considering the specific context and wording of each attack when evaluating their overall impact.

	Model	**Recommendation**		**Position**	
		%Aft.-%Bef.	#p	%Aft.-%Bef.	#p
social proof$_{\text{exp}}$	llama-8b	**+25.88**	**8**	**-1.22**	**8**
	llama-70b	**+40.11**	**9**	**-1.44**	**10**
	llama-405b	**+33.00**	**10**	**-1.75**	**9**
	claude3.5	**+25.30**	**10**	**-0.85**	**5**
	mistral	**+21.67**	**6**	**-1.52**	**8**
Discount Framing$_{\text{exp}}$	llama-8b	1.00	2	-1.37	3
	llama-70b	23.00	3	N/A	0
	llama-405b	17.33	3	-0.48	1
	claude3.5	**15.00**	2	**-0.44**	1
	mistral	N/A	0	1.13	2

Table 11.3: Results of the expert-crafted *social proof*$_{exp}$ and *discount framing*$_{exp}$ attacks for the coffee machines products.

Half Price vs. Discount Framing To examine the relative influence of biases on LLM recommendations, the following question was posed: *"To increase a product's visibility, is it more effective to silently halve its price, thereby enhancing its perceived value, or to advertise a discount without actually lowering the price?"* The comparison of these two approaches is shown in Table 11.4, which outlines the recommendation rates for a product in two scenarios: when its price is genuinely halved, and when it is kept at its original (double) price but framed with a *discount* in its description.

Interestingly, the *discount framing* approach consistently results in *more products being recommended*. This outcome is particularly striking given that the advertised discounts in these framing scenarios were never as high as 50%, with an average discount percentage of approximately 26.25 ± 5.34%.

Table 11.4: Halving a product's price vs employing *discount framing*. The instances where the impact of price halving is lower than the *discount framing* are underlined. In most cases, the unsubstantiated *discount frame* outperforms the actual halved price.

	Model	Recommendation		Position	
		%Aft.-%Bef.	#p	%Aft.-%Bef.	#p
1/2 price	llama-8b	+0.01	5	-0.83	2
	llama-70b	+11.25	4	-0.58	1
	llama-405b	+19.00	1	n/a	0
	claude3.5	+8.50	2	-0.48	2
	mistral	+5.00	1	-1.52	2

A critical consideration in assessing the true impact of the *discount framing* attack is the magnitude of the discount applied. For instance, a product advertised with an 80% discount may influence LLMs in various ways. An exceptionally high discount may appear implausible, potentially signaling to the model that it is not genuine. Conversely, if the item is genuinely on a substantial sale, the high discount might prompt the LLM to prioritize recommending it.

In our experiments, however, we deliberately avoid employing large, unrealistic discounts. This decision ensures that our analysis remains grounded in plausible, real-world scenarios. Additionally, the purpose of our investigation is to study the influence of social biases rather than to promote harmful practices. If a seller aims to improve the visibility of their product, relying on exaggerated or false discount claims would be counterproductive. Instead, genuine price reductions should be considered. Given this rationale, it is unrealistic to assume that product visibility can be significantly boosted by applying fictitious discounts of 80% or 90

The actual distribution of discounts used in our *generated* discount framing attacks is shown in Figure 11.3.3. The mean discount value is 26.25 ± 5.54%, with a median of 25.0%. Most discounts range from 15% to around 40%, reflecting a more realistic and practical approach.

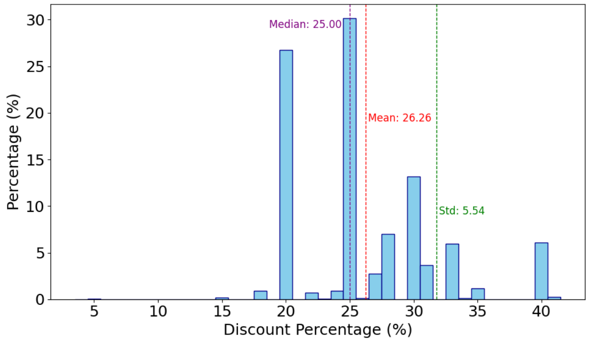


Figure 11.3.3: Distribution of discounts used in *generated discount framing* attacks.

11.4 Social Proof vs. Product Ratings

In this experiment, the previously introduced comparison of halving product prices versus employing the *discount framing* attack was further extended. It was observed that product ratings within the coffee machines dataset typically range between 3.9 and 4.8, making a rating of 2.1 an outlier well below the normal distribution. Due to this, a different analytical approach was required.

The objective was to estimate the *average improvement in ratings* that would be necessary to neutralize the influence of the *social proof* bias in the model's recommendations. For example, preliminary analysis of the Claude 3.5 Sonnet recommender, using the coffee machines dataset, suggested that a 0.5-star increase in product ratings could approximate the effect of incorporating social proof into the product descriptions. However, given that product ratings were already near the maximum 5-star rating, further increasing them was not practical. As a result, the focus shifted to the following question: *"What average reduction in product ratings would neutralize the social proof bias in LLM recommendations?"*

To answer this, product ratings were systematically decreased in increments of 0.1 to 0.5. These reduced ratings were applied to targeted products that simultaneously contained *social proof* bias in their descriptions. The subsequent recommendation rates for these manipulated products were then compared to the original, higher-rated versions.

The results, presented in Figure 11.4.1, demonstrate that the *social proof* bias generally bolstered product visibility as long as the reduction in rating was less than 0.27. For larger rating decreases, while social proof did not entirely counteract the drop in ratings, its presence still offered a measurable benefit. For instance, a comparison of the effects of a 0.40 rating reduction—both with and without social proof—indicated that even in these scenarios, social proof helped sustain higher recommendation rates than those achieved without it.

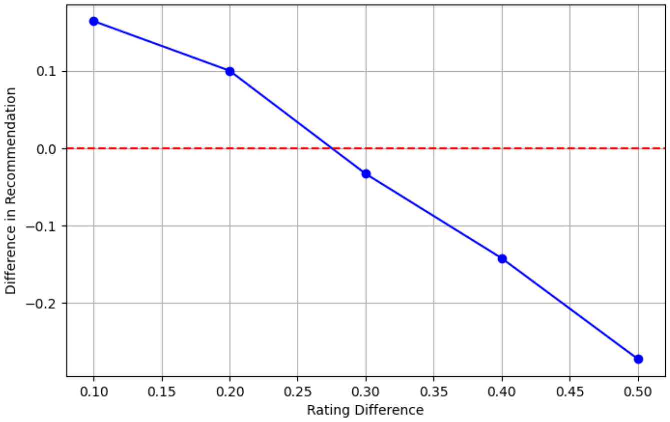


Figure 11.4.1: Difference in recommendation rates for the Claude 3.5 Sonnet recommender, applied to coffee machine products when their ratings were reduced and a *social proof* attack was simultaneously implemented. The red line marks the threshold where the recommendation rates of the original product and the manipulated product with reduced ratings converge.

Defense A significant challenge with cognitive biases as adversarial attacks is their subtle and embedded nature. Unlike traditional adversarial attacks that might rely on conspicuous sequences of random characters, these biases blend seamlessly into natural language [256, 165]. This makes them difficult to identify or filter out automatically. Furthermore, indiscriminately removing all information related to biases is not always an ideal solution. Some biases, such as a genuine discount, may reflect valuable information that a recommender system should consider. Thus, the challenge is to create a defense that can distinguish between benign and manipulative biases.

To examine the robustness of LLMs against cognitive bias-based attacks, the system prompts were adjusted to focus exclusively on product attributes, ignoring any biases present in the descriptions. The results of these adjustments, as applied to various influential attacks (both positive and negative), are summarized in Table 11.5. Notably, the outcomes demonstrate that the effectiveness of the attacks remained largely unchanged regardless of whether the defense prompt was used. This indicates that the defenses employed were *not sufficient* to mitigate the influence of these biases.

For instance, when using Llama-8b, the *exclusivity bias* led to a mean position increase of 0.11 for five products—an effect contrary to the previously observed outcome. However, this positional change was accompanied by a 30% decrease in recommendation frequency for seven products, a decrease that was even more pronounced in the absence of the defense prompt. Consequently, in this case, the defensive prompt did not improve the model's resistance to the attack, highlighting the inherent difficulty of countering cognitive bias-based adversarial techniques.

	Model	**Recommendation**		**Position**	
		%Aft.-%Bef.	$\#p$	%Aft.-%Bef.	$\#p$
Soc. Proof	llama-8b	+19.75	4	-1.29	4
	llama-70b	+20.00	4	-1.00	5
	llama-405b	+19.25	4	-0.20	4
	claude3.5	+13.00	3	-0.66	2
	mistral	+13.00	1	-0.51	3
Exclus.	llama-8b	-30.43	7	-0.11	5
	llama-70b	-30.60	10	+0.98	3
	llama-405b	-24.40	5	+2.37	4
	claude3.5	-31.29	7	+2.76	3
	mistral	-6.00	2	+0.91	4

Table 11.5: Results of attacks with positive and a negative impact on product visibility, using the defensible system prompt on the coffee machines products. Comparison with the same biases in Table 11.2 indicates similar *recommendation* and *position* tendencies.

Bias	**Recommendation**		**Position**	
	%Aft.-%Bef (↑)	#p (↑)	Aft.-Bef (↓)	#p (↑)
		Chew Toys		
social proof$_{\text{exp}}$	n/a	0	-0.54 ± 0.13	3
social proof	+16.00 ± 0.00	1	-0.38 ± 0.00	2
exclusivity$_{\text{exp}}$	-48.00 ± 0.00	1	+0.61 ± 0.31	3
exclusivity	-21.00 ± 0.00	1	+0.48 ± 0.23	3
		Laptops		
social proof$_{\text{exp}}$	+16.33 ± 3.86	3	-0.49 ± 0.00	1
social proof	n/a	0	-0.30 ± 0.4	2
exclusivity$_{\text{exp}}$	-15.00 ± 0.00	1	0.08 ± 0.02	2
exclusivity	n/a	0	0.90 ± 0.00	1

Table 11.6: The impact of cognitive biases on Claude using two subsets of Amazon's dataset [119] (chew toys and laptops).

11.4.1 Real-World Evaluation

The initial analysis utilized controlled datasets, consistent with prior literature, featuring concise product descriptions. This controlled setting allowed for the identification of clear and repeatable cognitive bias effects. Building on those results, the current investigation extends to real-world data, specifically evaluating the influence of *social proof* and *exclusivity* biases. These two biases were chosen because they demonstrated some of the most pronounced positive and negative impacts, respectively, in earlier experiments.

A new dataset was curated from Amazon Reviews metadata [119] to approximate realistic advertising conditions. This dataset retained key attributes—such as price, ratings, and product descriptions—mirroring the structure of the controlled datasets used in previous analyses. However, the real-world descriptions are notably longer and more complex, often integrating technical specifications with persuasive language reflective of actual marketing strategies.

To maintain a consistent analytical framework, the evaluation focused on two consumer-favored product categories: laptops and pet chew toys. Each category included 10 items, ensuring the same dataset size as in earlier studies. Products were filtered to include only those with high ratings, determined using a Bayesian average that accounts for both review counts and individual ratings. Additionally, only items with complete metadata fields—such as price and ratings—were included.

The results confirmed the same consistent patterns observed in controlled datasets. For instance, in the laptop category using the Claude 3.5 Sonnet model, the *social proof* attack increased the recommendation rates for three products by an average of 288.88%. Prior to the attack, these products had recommendation rates of 12%, 2%, and 12%, which rose to 30%, 13%, and 32%, respectively, after the attack. This change represents the percentage increase from the pre-attack rates. Notably, the product positions in the recommendation lists remained unchanged. In contrast, biases with negative effects, such as *exclusivity*, demonstrated a decrease in recommendation rates. For example, in the same dataset and model, the recommendation rate dropped by -22%, falling from an average of 71% to 56%, reflecting a change (*%aft.-%bef.*) of -15%.

Specifically, Table 11.6 presents the results for the Amazon dataset's "chew toys" subset, analyzed using Claude 3.5 Sonnet. This evaluation highlights the impact of two influential attacks—*social proof* and *exclusivity*—in scenarios where both expert-crafted and LLM-generated attacks were applied. Consistent with earlier datasets (coffee machines, cameras, books, laptops), the table demonstrates that these attacks continue to exert a similar influence on product visibility. However, a noteworthy distinction lies in the reduced prominence of the attack's impact when compared to the datasets analyzed in [165].

This diminished effect is likely attributable to the real-world dataset's inherent integration of various social biases within the product descriptions. For instance, in the laptop dataset, phrases like "*Business Laptop, Intel Core i5-1235U (Beats i7-1165g7)*" highlight a product's superiority by explicitly comparing it to another. Similarly, promotional incentives like "*Bonus 32GB SnowBell USB Card*" add persuasive elements. These pre-existing cognitive biases, embedded within the real-world descriptions, may reduce the visibility of additional manipulative biases. In fact, the interplay of multiple biases—such as scarcity enhancing visibility when combined with discount framing—complicates the analysis and can dampen the observed effects of targeted attacks.

Further differences emerge in the length and complexity of the product descriptions across datasets. On average, the chew toy products in the Amazon dataset were described with 900.3 characters (126.8 words), whereas the laptop descriptions averaged 1436 characters (172.3 words). In comparison, descriptions from the coffee machines dataset used approximately 219.2 tokens (16.6 words), cameras averaged 227.6 characters (14.9 words), and books featured about 247.0 characters (18.1 words). These statistics were obtained using the NLTK tokenization package[4]. Despite the relatively small size of the added attacks, the presence of additional cognitive biases in the base descriptions played a substantial role in shaping the models' recommendations across all datasets, influencing their overall visibility and ranking behavior.

11.5 Conclusion

This study presents a novel approach to leveraging cognitive biases as subtle adversarial techniques aimed at influencing large language model (LLM)-based product recommendation systems. By embedding these biases directly into product descriptions, the work demonstrates how seemingly innocuous language modifications can meaningfully shift LLM recommendation rankings. The experiments identify which cognitive biases have the most pronounced effects on recommendation outcomes, revealing a significant vulnerability within LLM-based recommendation frameworks. This vulnerability stems not only from the inability to easily detect such biases but also from the difficulty in defending against them.

The findings underscore a fundamental blind spot in the current implementation of LLM-driven recommendation systems. Despite their impressive performance in many natural language understanding tasks, these models exhibit limited robustness when exposed to cognitive bias manipulations. Even subtle, well-crafted attacks can produce noticeable shifts in recommended products, indicating that these systems are far more susceptible than previously understood. Moreover, the study highlights the considerable variability in how different LLMs respond to the same bias, showing that their behavior in commercial recommendation settings is often unpredictable.

Beyond simply identifying these vulnerabilities, the research contributes valuable insights into the relationship between language patterns and model behavior. It also emphasizes the pressing need for improved defenses against adversarial influences, particularly as LLMs continue to play a growing role in commercial recommendation environments. By exposing these blind spots, this work offers a critical step toward enhancing the reliability and fairness of LLM-based recommendation systems.

[4] https://www.nltk.org/api/nltk.tokenize.html

While the current investigation focuses primarily on LLM-based product recommendation systems, future research could expand the scope of these methodologies in several important directions. First, exploring a broader range of product categories could yield a more comprehensive understanding of how cognitive biases manifest and influence recommendation outcomes. For instance, applying these techniques to highly competitive sectors, such as electronics, fashion, or automotive products, may uncover different patterns of susceptibility. By examining diverse categories, researchers can better assess the generalizability of these findings and refine their strategies for detecting and mitigating bias.

In addition to diversifying product categories, future work could also extend this methodology to the domain of news and information dissemination. Cognitive biases may have profound implications for how LLMs summarize news articles, integrate conflicting information, and present conclusions to end-users. Investigating the interplay between social and cognitive biases in news summarization tasks could provide valuable insights into how LLMs prioritize certain narratives over others. This research may help identify vulnerabilities in LLMs that could contribute to the unintentional amplification of misinformation or biased reporting.

Furthermore, these methodologies could be adapted to study how cognitive biases affect the integration of multiple information sources. For example, when LLMs attempt to reconcile disparate viewpoints from various articles or user-generated content, subtle biases in the phrasing or emphasis of certain facts might influence the final summarized output. By expanding the current approach into this realm, researchers could gain a deeper understanding of how cognitive biases shape not just product recommendations, but also the perceived credibility and reliability of information delivered by LLMs.

Another critical direction is to develop robust, scalable defense mechanisms that can counteract cognitive biases in real time. Future studies could explore automated techniques for detecting bias at the description level before it affects recommendation rankings or information summaries. Building on the current findings, researchers could test the efficacy of adaptive prompt engineering, fine-tuning on bias-resistant training data, or integrating external knowledge bases that help verify the validity of claims in both product descriptions and news articles.

In summary, the next steps in this line of research include: (1) expanding the methodology to cover a wider variety of products and categories, (2) applying these approaches to the domain of news summarization and misinformation, (3) analyzing the integration of information from multiple sources under the influence of cognitive biases, and (4) developing advanced defense strategies to mitigate the impact of these biases. By pursuing these directions, future research can deepen our understanding of LLM behavior across diverse domains and enhance the resilience and fairness of recommendation and summarization systems.

Chapter 12

Conclusion

This this dissertation it is presented ways for generating and evaluating the semantic counterfatual exaplantions.

Specifically Chapter 4 it is presented a general framework for generating semantic counterfactual explanations using a knowledge-graphs, emphasizing the essential role of semantics—i.e., meaningful labels and relational information—when designing explainable AI systems. It utilized the concept of an "Explanation Dataset," which pairs items' semantic annotations (described in Description Logics) with their feature representations for a classifier. Counterfactual explanations are then defined as minimal, semantically interpretable edits—replacements, insertions, or deletions of concepts and roles—that transform an exemplar's ABox assertions so that it matches another exemplar already classified into a desired target class. Beyond local explanations for single items, the framework offers a way to compute global explanations by aggregating frequent semantic edits across multiple instances. Although computing exact graph edit distances is NP-hard, the chapter proposes an efficient approximation approach by focusing on concept-level edits. This setup, along with the accompanying definitions and algorithms, establishes a principled path for building and interpreting conceptual counterfactual explanations based on knowledge-graph enrichments.

Following, Chapter 5 emphasizes the importance of modeling not only the concepts but also the relationships (edges) between them when computing counterfactual explanations, since interactions such as "person rides bicycle" can be pivotal for understanding classifier decisions (e.g., distinguishing "pedestrian" vs. "driver"). It proposes both a set-edit approach—where edges are "rolled up" into concepts of the form $\exists r.C$—and a more advanced Graph Neural Network (GNN) method that embeds entire scene graphs for efficient retrieval of nearest counterfactual exemplars. Extensive experiments on diverse datasets (CUB for bird classification, Places with Visual Genome for scene understanding, custom "pedestrian vs. driver" images, and even an audio-based COVID-19 classification dataset) consistently show that preserving and utilizing relational information leads to more faithful, minimal, and human-interpretable counterfactuals. User studies confirm these semantic graph-based explanations not only match or outperform a state-of-the-art image-based approach but also help humans learn and apply the classifier's "rules" more effectively, even in "blind" settings without direct visual cues. Finally, the chapter discusses broader implications—such as the need for well-curated knowledge bases, potential integration with generative models, and ongoing research to ensure robustness and scalability across modalities—underscoring the value of conceptual and relational explanations in explainable AI.

However, despite the framework presented above, the aforementioned framework cannot be used for generating new instances. This is particularly relevant in the field of explainability, where the generation of counterfactual samples is a primary technique. Chapter 6 addresses the challenge of generating high-quality textual counterfactual explanations—minimal yet meaningful edits to text samples that change or stress-test a classifier's prediction. Extending the semantic, model-agnostic ideas described in earlier chapters, it introduces a framework that no longer merely searches for "close" instances in a dataset but instead constructs new text samples by optimally substituting words. Central to this approach is formulating counterfactual generation as a "relaxed" bipartite matching (or assignment) problem between source and target words, for which two algorithms are explored: (1) a **deterministic** solution via classical graph-matching methods that guarantees optimality but can be slow for large datasets, and (2) a **GNN-based** solution that provides a near-optimal matching at significantly reduced runtime. These bipartite edges can be weighted in transparent ways (e.g., via WordNet path similarity) or through modern embedding-based word similarities, the latter yielding fewer edits but being less explainable. Experiments on sentiment and topic classification (IMDB and Newsgroups) show that this approach surpasses two state-of-the-art editors (MiCE and Polyjuice) in most quality metrics (fluency, semantic closeness, minimality) while running up to 50× faster. The chapter concludes with a discussion of key trade-offs—optimality vs. speed, explainability vs. performance, and controllability vs. minimality—and suggests future directions like integrating additional lexical resources or refining GNN performance for even better approximations of the deterministic solution.

Following Chapter 7, several approaches are presented for evaluating counterfactual explanations by systematically categorizing and assessing multiple counterfactual editors across textual and visual domains. A novel iterative feedback method is introduced, where outputs from different iterations of the counterfactual editing procedure are utilized as a form of ground truth to evaluate optimality. Specifically, this involves iteratively feeding the outputs of the editors back into themselves, producing subsequent edits, and uncovering any inconsistencies or suboptimal modifications. To quantify these deviations, a novel metric called *inconsistency (inc@n)* is proposed, effectively distinguishing editors based on their capacity to consistently achieve minimal edits. The inconsistency metric measures optimality across various underlying metrics. Experimental evaluations were conducted using text editors such as MiCE, Polyjuice, and TextFooler on the IMDb and Newsgroups datasets. Editors were rigorously assessed using metrics including Flip Rate, Minimality, Fluency, and Grammatical Correctness. Results demonstrated that TextFooler consistently produced minimal and stable edits, whereas Polyjuice frequently introduced extensive modifications, especially in longer texts. MiCE exhibited strong initial performance but showed declining effectiveness and increasing inconsistencies with successive iterations of feedback.

After presenting methodologies for generating and evaluating counterfactual explanations, four distinct applications of these methodologies are discussed beyond their traditional use for explainability. The first two applications involve directly applying the methodology described in Chapter 5 to assess the generative capabilities of various generative systems. Specifically, Chapter 8 introduces a method for evaluating image and story visualization systems. Rather than utilizing counterfactual explanations solely for model interpretability, the same algorithm is adapted to detect errors occurring between the input prompts and the generated images or stories. An explainable metric is proposed, offering not only quantitative values suitable for comparing different models but also clear explanations highlighting the specific sources of generation errors.

In a similar vein, Chapter 9 adopts the same evaluation methodology to measure the hallucination rates of LVLMs. It presents an explainable benchmarking approach to systematically detect and explain errors in image captioning processes. The results underscore that LVLMs generally produce fewer hallucinations compared to traditional image captioning systems. Additionally, the findings indicate that hallucination frequency increases when captions are artificially lengthened, highlighting an essential consideration for improving the reliability of LVLM-generated captions.

Chapter 10 present an application of counterfactual explanations to the field of NLP. Specifically, it examines the incorporation of counterfactual examples to improve the reasoning abilities of LLMs. Initially, it shows how LLMs tackle puzzle-solving tasks, particularly riddles, by systematically categorizing puzzles based on their reliance on formal rules or commonsense reasoning and exploring various prompting techniques to enhance model performance. The research introduces a novel method—referred to as RISCORE—that leverages "counterfactual" or "context-reconstructed" riddles: puzzles that require the same core reasoning steps but present them in alternative settings. By incorporating these reconstructions in a few-shot learning setup, the method demonstrates improvements in both vertical reasoning (rule-based logic) and lateral thinking (creative, out-of-the-box problem-solving), as evidenced by evaluations on datasets like BrainTeaser and RiddleSense. Empirical comparisons reveal that combining the original riddles with their reconstructed variations often outperforms standard prompting approaches, including chain-of-thought methods, illustrating the effectiveness of context-shifted examples in fostering more robust reasoning. Furthermore, an automated pipeline for generating reconstructed riddles is introduced, enabling the approach to generalize to puzzle collections that lack preexisting context adaptations. Overall, the findings highlight the power of tailored examples that preserve reasoning pathways while varying the context, offering a scalable and practical method to enhance LLM reasoning across diverse puzzle-solving benchmarks.

Lastly, Chapter 11 explores an application of counterfactual explanations beyond traditional classification tasks—specifically, their role in interpreting and influencing product recommendations made by LLMs. Unlike typical recommendation systems aiming solely at accuracy, this chapter emphasizes understanding the decision-making processes within LLMs by strategically leveraging cognitive biases. The chapter employs cognitive biases, widely recognized in human psychology and marketing, as subtle adversarial strategies to manipulate product recommendations. This method examines whether embedding biases like social proof or scarcity into product descriptions can systematically affect an LLM's recommendation rankings. By drawing parallels between human decision-making and LLM behaviors, the chapter investigates how certain biases, despite being effective in traditional marketing, paradoxically reduce product visibility in an LLM-driven context. Experimental evaluations utilized both synthetic and real-world datasets, including fictional products (coffee machines, cameras, and books) and authentic product descriptions from Amazon. These experiments were conducted across multiple LLM architectures such as Claude 3.5 Sonnet, Mistral, and various versions of Llama, assessing the robustness and susceptibility of these models to cognitively biased descriptions.

Results highlighted biases such as social proof and discount framing significantly boosting product visibility, whereas exclusivity and scarcity typically reduced it. The research further revealed the difficulty in countering such subtle biases, as even explicitly instructing LLMs to focus solely on factual product attributes had limited defensive effectiveness. Overall, the findings underscore vulnerabilities in current LLM-based recommendation systems, demonstrating their susceptibility to subtle linguistic manipulations rooted in cognitive biases. The study contributes valuable insights into improving the security, fairness, and robustness of AI-driven recommendations, suggesting further research directions into diverse product categories, news summarization, misinformation detection, and the development of advanced defenses.

12.1 Future and Ongoing Work

Building on the methodologies and findings presented throughout this dissertation—encompassing semantic knowledge-graph counterfactuals, text-based editors, puzzle-solving prompts, and other techniques—there is considerable potential for further advances in multiple directions. One key avenue involves enriching the semantic resources used in our counterfactual generation frameworks. Although ontologies, taxonomies, and annotated datasets play an essential role in creating human-interpretable explanations, these resources can be difficult to obtain. Efforts to automate annotation pipelines—for example, by leveraging advanced object detection systems or entity-linking methods—may alleviate this problem, particularly for under-annotated domains. Moreover, integrating multiple knowledge bases, such as ConceptNet or DBpedia, alongside WordNet, can substantially increase the range of covered concepts and the depth of semantic relationships. Incorporating more sophisticated logical formalisms, such as constraints and property chains, would also enhance expressivity while requiring careful attention to scalability and runtime considerations.

Beyond improving resources, scaling semantic counterfactual explanations to new or more complex data types represents another important strand of work. Video understanding, which introduces spatiotemporal relationships, poses unique challenges for counterfactual generation, as changes must maintain consistency across frames. Similarly, textual systems can be extended into multilingual and cross-lingual setups, ensuring that the approach remains effective even when switching linguistic domains. In specialized fields like healthcare, structured EHRs require domain-specific ontologies and constraints so that any generated edits comply with medical guidelines.

The text-based editors and graph algorithms discussed in Chapters 5 and 6 open the door to refinements in counterfactual editing. Graph Neural Network (GNN) approaches, which replace deterministic algorithms with approximate solutions, already demonstrate considerable speed-ups, but they may be made even more robust by leveraging advanced neural architectures or by incorporating user-defined constraints for better controllability. It will also be important to maintain model quality under domain shifts or adversarial conditions; exploring adversarial training or careful data augmentation could help preserve performance when editors face text domains for which they were not explicitly trained.

Regarding evaluation, user studies revealed that semantic and knowledge-graph-based explanations tend to align more naturally with human intuition. This finding encourages further expansion of human-centered methods: for instance, building interactive explanation interfaces that allow users to iteratively refine and query counterfactual edits in real time. Studies over longer time frames could track how these explanations help non-experts internalize an AI model's decision boundaries, whether they increase user trust, and if that trust remains stable across repeated interactions. Alongside these practical deployments, further examining cross-cultural and cross-domain applicability would help ensure that such semantic explanations remain clear and contextually appropriate even when user groups and cultural norms vary significantly.

Generative models also offer fertile ground for applying conceptual counterfactuals. With the growing popularity of text-to-image and story-generation systems, the challenge of detecting hallucinations—spurious or illogical content in machine-generated outputs—has become more pressing. Counterfactual analysis can detect precisely where generative models deviate from intended concepts, then guide subsequent improvements to mitigate bias or unwarranted artifacts. In text-heavy domains, conceptual checks might help authors or end-users identify misalignments between a system's generated story narrative and its visual or contextual representation. Adapting these ideas to 3D or virtual reality scenes would extend conceptual consistency checks into more immersive environments, though it would require novel ways of measuring semantic closeness in three-dimensional spaces.

Scaling up to extremely large datasets and complex TBoxes introduces further challenges. Methods such as indexing or approximate nearest-neighbor searches may become critical to retaining efficiency. Advanced symbolic reasoning could allow more expressive TBox axioms while controlling the combinatorial explosion of possibilities. Continual and incremental learning methods would then help the explanation dataset adapt in tandem with a model's evolving structure or retraining regime, maintaining meaningful alignment over time.

Turning to puzzle-based reasoning, there is scope to expand the context-reconstructed riddle framework described in Chapter 10. Beyond simple puzzle types, more elaborate knowledge-intensive tasks or domain-oriented scenarios—such as legal or clinical case studies framed as puzzles—could serve as valuable tests for advanced LLM reasoning. Integrating puzzle-solving LLMs with symbolic solvers promises more systematic resolution of tasks with fixed rules, while context-based reconstruction remains valuable for lateral thinking and creative domains. The possibility of automated puzzle generation would facilitate curriculum learning, allowing the difficulty of riddles to scale with the model's evolving capabilities.

In the field of LLMs, counterfactual explanations can also be adapted to shape product recommendations generated by such models, as outlined in Chapter 11. By introducing small, psychologically driven text modifications—rooted in biases such as social proof or scarcity—it is demonstrated that counterfactual adjustments can profoundly affect which items are highlighted. These counterfactual edits, designed to resemble typical marketing language, are employed to exploit a blind spot in LLMs that struggle to distinguish impartial attributes from deliberately crafted cues. Consequently, attention is drawn to the urgent need for more robust defenses against bias-driven manipulations in AI-based recommendation systems.

Finally, while semantic counterfactuals can expose biases, such as a tendency to overemphasize certain concepts or attribute unwarranted associations, additional scrutiny is needed to ensure that these methods themselves do not introduce new biases or unintentional harms. In regulated domains—such as medicine, finance, or law—verifiable and robust explanations are imperative, prompting the need for techniques that can certify the faithfulness and reliability of semantic edits. Privacy considerations also come into play if sensitive data is inadvertently revealed through semantic constraints, driving research into developing privacy-preserving methods for knowledge extraction and counterfactual generation.

Taken together, these future directions underscore both the versatility and the critical importance of conceptual, knowledge-based methods in AI explainability. By complementing data-driven techniques with explicitly modeled semantics, researchers can continue to enhance user understanding, trust, and overall transparency in complex systems. Pursuing richer ontologies, more diverse domains, deeper human evaluations, integration with generative models, and robust puzzle-solving tests will collectively shape the next wave of advancements in semantic counterfactual explanations and interpretable AI.

Chapter 13

Bibliography

[1] Abhishek, Kumar and Kamath, Deeksha. *Attribution-based XAI Methods in Computer Vision: A Review*. 2022. DOI: 10.48550/ARXIV.2211.14736. URL:

[2] Abid, Abubakar, Yuksekgonul, Mert, and Zou, James. "Meaningfully debugging model mistakes using conceptual counterfactual explanations". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 66–88.

[3] Abu-Aisheh, Zeina, Raveaux, Romain, Ramel, Jean-Yves, and Martineau, Patrick. "An exact graph edit distance algorithm for solving pattern recognition problems". In: *4th International Conference on Pattern Recognition Applications and Methods 2015*. 2015.

[4] Adadi, Amina and Berrada, Mohammed. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6 (2018), pp. 52138–52160.

[5] Agrawal, Rakesh and Srikant, Ramakrishnan. "Fast algorithms for mining association rules". In: *Proc. of 20th Intl. Conf. on VLDB*. 1994, pp. 487–499.

[6] Ahn, Daechul, Kim, Daneul, Song, Gwangmo, Kim, Seung Hwan, Lee, Honglak, Kang, Dongyeop, and Choi, Jonghyun. "Story Visualization by Online Text Augmentation with Context Memory". In: *ICCV*. 2023.

[7] AI@Meta. "Llama 3 Model Card". In: (2024). URL:

[8] Akula, Arjun, Wang, Shuai, and Zhu, Song-Chun. "Cocox: Generating conceptual and counterfactual explanations via fault-lines". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 03. 2020, pp. 2594–2601.

[9] Alammar, J. "Ecco: An Open Source Library for the Explainability of Transformer Language Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Ed. by Heng Ji, Jong C. Park, and Rui Xia. Online: Association for Computational Linguistics, Aug. 2021, pp. 249–257. DOI: 10.18653/v1/2021.acl-demo.30. URL:

[10] Argyrou, Georgia, Dimitriou, Angeliki, Lymperaiou, Maria, Filandrianos, Giorgos, and Stamou, Giorgos. "Automatic Generation of Fashion Images using prompting in generative machine learning models". In: *arXiv preprint arXiv:2407.14944* (2024).

[11] Argyrou, Georgia, Dimitriou, Angeliki, Lymperaiou, Maria, Filandrianos, Giorgos, and Stamou, Giorgos. "Prompt2Fashion: An automatically generated fashion dataset". In: *Proceedings of the 13th Hellenic Conference on Artificial Intelligence*. 2024, pp. 1–6.

[12] Arrieta, Alejandro Barredo, Díaz-Rodríguez, Natalia, Del Ser, Javier, Bennetot, Adrien, Tabik, Siham, Barbado, Alberto, García, Salvador, Gil-López, Sergio, Molina, Daniel, Benjamins, Richard, et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information fusion* 58 (2020), pp. 82–115.

[13] Augustin, Maximilian, Boreiko, Valentyn, Croce, Francesco, and Hein, Matthias. "Diffusion visual counterfactual explanations". In: *arXiv preprint arXiv:2210.11841* (2022).

[14] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, eds. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[15] Bang, Yejin, Cahyawijaya, Samuel, Lee, Nayeon, Dai, Wenliang, Su, Dan, Wilie, Bryan, Lovenia, Holy, Ji, Ziwei, Yu, Tiezheng, Chung, Willy, Do, Quyet V., Xu, Yan, and Fung, Pascale. "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity". In: *ArXiv* abs/2302.04023 (2023). URL:

[16] Bao, Qiming, Gendron, Gaël, Peng, Alex Yuxuan, Zhong, Wanjun, Tan, Neşet Özkan, Chen, Yang, Witbrock, Michael, and Liu, Jiamou. "A Systematic Evaluation of Large Language Models on Out-of-Distribution Logical Reasoning Tasks". In: *ArXiv* abs/2310.09430 (2023). URL:

[17] Bau, David, Zhu, Jun-Yan, Wulff, Jonas, Peebles, William, Strobelt, Hendrik, Zhou, Bolei, and Torralba, Antonio. *Seeing What a GAN Cannot Generate*. 2019. DOI: 10.48550/ARXIV.1910.11626. URL:

[18] Benny, Yaniv, Galanti, Tomer, Benaim, Sagie, and Wolf, Lior. "Evaluation Metrics for Conditional Image Generation". In: *International Journal of Computer Vision* 129.5 (Mar. 2021), pp. 1712–1731. DOI: 10.1007/s11263-020-01424-w. URL:

[19] Besta, Maciej, Blach, Nils, Kubicek, Ales, Gerstenberger, Robert, Gianinazzi, Lukas, Gajda, Joanna, Lehmann, Tomasz, Podstawski, Michal, Niewiadomski, Hubert, Nyczyk, Piotr, and Hoefler, Torsten. *Graph of Thoughts: Solving Elaborate Problems with Large Language Models*. 2023. arXiv: 2308.09687 [cs.CL].

[20] Besta, Maciej, Memedi, Florim, Zhang, Zhenyu, Gerstenberger, Robert, Blach, Nils, Nyczyk, Piotr, Copik, Marcin, Kwaśniewski, Grzegorz, Müller, Jürgen, Gianinazzi, Lukas, et al. "Topologies of reasoning: Demystifying chains, trees, and graphs of thoughts". In: *arXiv preprint arXiv:2401.14295* (2024).

[21] Bijsterbosch, Jeroen and Volgenant, Ton. "Solving the Rectangular assignment problem and applications". In: *Annals OR* 181 (Dec. 2010), pp. 443–462. DOI: 10.1007/s10479-010-0757-3.

[22] Bisk, Yonatan, Zellers, Rowan, Bras, Ronan Le, Gao, Jianfeng, and Choi, Yejin. "PIQA: Reasoning about Physical Commonsense in Natural Language". In: *Thirty-Fourth AAAI Conference on Artificial Intelligence*. 2020.

[23] Blattmann, Andreas, Rombach, Robin, Oktay, Kaan, Müller, Jonas, and Ommer, Björn. *Semi-Parametric Neural Image Synthesis*. 2022. DOI: 10.48550/ARXIV.2204.11824. URL:

[24] Boreiko, Valentyn, Panfilov, Alexander, Voracek, Vaclav, Hein, Matthias, and Geiping, Jonas. *A Realistic Threat Model for Large Language Model Jailbreaks*. 2024. arXiv: 2410.16222 [cs.LG]. URL:

[25] Borji, Ali. "Pros and cons of GAN evaluation measures: New developments". In: *Computer Vision and Image Understanding* 215 (2022), p. 103329. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2021.103329. URL:

[26] Brown, Tom B., Mann, Benjamin, Ryder, Nick, Subbiah, Melanie, Kaplan, Jared, Dhariwal, Prafulla, Neelakantan, Arvind, Shyam, Pranav, Sastry, Girish, Askell, Amanda, Agarwal, Sandhini, Herbert-Voss, Ariel, Krueger, Gretchen, Henighan, T. J., Child, Rewon, Ramesh, Aditya, Ziegler, Daniel M., Wu, Jeff, Winter, Clemens, Hesse, Christopher, Chen, Mark, Sigler, Eric, Litwin, Mateusz, Gray, Scott, Chess, Benjamin, Clark, Jack, Berner, Christopher, McCandlish, Sam, Radford, Alec, Sutskever, Ilya, and Amodei, Dario. "Language Models are Few-Shot Learners". In: *ArXiv* abs/2005.14165 (2020). URL:

[27] Browne, Kieran and Swift, Ben. "Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks". In: *arXiv preprint arXiv:2012.10076* (2020).

[28] Buhrmester, Vanessa, MÃ¼nch, David, and Arens, Michael. "Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey". In: *Machine Learning and Knowledge Extraction* 3.4 (2021), pp. 966–989. ISSN: 2504-4990. DOI: 10.3390/make3040048. URL:

[29] Burkard, Rainer Ernst and Çela, Eranda. "Linear assignment problems and extensions". English. In: *Handbook of Combinatorial Optimization*. 1st ed. Supplement Volume A. Netherlands: Kluwer Academic Publishers, 1999, pp. 75–149.

[30] Byrne, Ruth MJ. "Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning." In: *IJCAI*. 2019, pp. 6276–6282.

[31] Calderon, Nitay, Ben-David, Eyal, Feder, Amir, and Reichart, Roi. "DoCoGen: Domain Counterfactual Generation for Low Resource Domain Adaptation". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 7727–7746. DOI: 10.18653/v1/2022.acl-long.533. URL:

[32] Chai, Lucy, Wulff, Jonas, and Isola, Phillip. “Using latent space regression to analyze and leverage compositionality in GANs”. In: *ArXiv* abs/2103.10426 (2021).

[33] Chemmengath, Saneem, Azad, Amar Prakash, Luss, Ronny, and Dhurandhar, Amit. *Let the CAT out of the bag: Contrastive Attributed explanations for Text.* 2022. arXiv: 2109.07983 [cs.CL].

[34] Chen, Banghao, Zhang, Zhaofeng, Langren'e, Nicolas, and Zhu, Shengxin. “Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review”. In: *ArXiv* abs/2310.14735 (2023). URL:

[35] Chen, Hong, Han, Rujun, Wu, Te-Lin, Nakayama, Hideki, and Peng, Nanyun. “Character-centric Story Visualization via Visual Planning and Token Alignment”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing.* 2022, pp. 8259–8272.

[36] Chen, Liang-Chieh, Papandreou, George, Kokkinos, Iasonas, Murphy, Kevin, and Yuille, Alan L. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[37] Chen, Mark, Tworek, Jerry, Jun, Heewoo, Yuan, Qiming, Ponde, Henrique, Kaplan, Jared, Edwards, Harrison, Burda, Yura, Joseph, Nicholas, Brockman, Greg, Ray, Alex, Puri, Raul, Krueger, Gretchen, Petrov, Michael, Khlaaf, Heidy, Sastry, Girish, Mishkin, Pamela, Chan, Brooke, Gray, Scott, Ryder, Nick, Pavlov, Mikhail, Power, Alethea, Kaiser, Lukasz, Bavarian, Mohammad, Winter, Clemens, Tillet, Philippe, Such, Felipe Petroski, Cummings, David W., Plappert, Matthias, Chantzis, Fotios, Barnes, Elizabeth, Herbert-Voss, Ariel, Guss, William H., Nichol, Alex, Babuschkin, Igor, Balaji, Suchir, Jain, Shantanu, Carr, Andrew, Leike, Jan, Achiam, Joshua, Misra, Vedant, Morikawa, Evan, Radford, Alec, Knight, Matthew M., Brundage, Miles, Murati, Mira, Mayer, Katie, Welinder, Peter, McGrew, Bob, Amodei, Dario, McCandlish, Sam, Sutskever, Ilya, and Zaremba, Wojciech. “Evaluating Large Language Models Trained on Code”. In: *ArXiv* abs/2107.03374 (2021). URL:

[38] Chen, Nuo, Liu, Jiqun, Dong, Xiaoyu, Liu, Qijiong, Sakai, Tetsuya, and Wu, Xiao-Ming. *AI Can Be Cognitively Biased: An Exploratory Study on Threshold Priming in LLM-Based Batch Relevance Assessment.* 2024. arXiv: 2409.16022 [cs.CL]. URL:

[39] Chen, Wenhu, Ma, Xueguang, Wang, Xinyi, and Cohen, William W. “Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks”. In: *ArXiv* abs/2211.12588 (2022). URL:

[40] Chen, Yangyi, Sikka, Karan, Cogswell, Michael, Ji, Heng, and Divakaran, Ajay. *Measuring and Improving Chain-of-Thought Reasoning in Vision-Language Models.* 2023. arXiv: 2309.04461 [cs.CL].

[41] Chen, Zeming, Gao, Qiyue, Bosselut, Antoine, Sabharwal, Ashish, and Richardson, Kyle. “Disco: distilling counterfactuals with large language models”. In: *arXiv preprint arXiv:2212.10534* (2022).

[42] Chen, Zeming, Gao, Qiyue, Bosselut, Antoine, Sabharwal, Ashish, and Richardson, Kyle. “DISCO: Distilling Counterfactuals with Large Language Models”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 5514–5528. DOI: 10.18653/v1/2023.acl-long.302. URL:

[43] Chen, Zeming, Gao, Qiyue, Bosselut, Antoine, Sabharwal, Ashish, and Richardson, Kyle. *DISCO: Distilling Counterfactuals with Large Language Models.* 2023. arXiv: 2212.10534 [cs.CL].

[44] Chu, Zheng, Chen, Jingchang, Chen, Qianglong, Yu, Weijiang, He, Tao, Wang, Haotian, Peng, Weihua, Liu, Ming, Qin, Bing, and Liu, Ting. “A Survey of Chain of Thought Reasoning: Advances, Frontiers and Future”. In: *ArXiv* abs/2309.15402 (2023). URL:

[45] Cognitive Science Laboratory, Princeton University. *WordNet - A lexical database for the English language.* , last accessed on March 8th, 2006. 2006.

[46] Cong, Yuren, Yang, Michael Ying, and Rosenhahn, Bodo. “RelTR: Relation Transformer for Scene Graph Generation”. In: *CoRR* abs/2201.11460 (2022). arXiv: 2201.11460. URL:

[47] Cong, Yuren, Yang, Michael Ying, and Rosenhahn, Bodo. “Reltr: Relation transformer for scene graph generation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[48] Creswell, Antonia, Shanahan, Murray, and Higgins, Irina. “Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning”. In: *ArXiv* abs/2205.09712 (2022). URL:

[49] d'Amato, Claudia, Fanizzi, Nicola, and Esposito, Floriana. “A semantic similarity measure for expressive description logics”. In: *arXiv preprint arXiv:0911.5043* (2009).

[50] d'Amato, Claudia, Mahon, Louis, Monnin, Pierre, and Stamou, Giorgos. "Machine Learning and Knowledge Graphs: Existing Gaps and Future Research Challenges". In: *Transactions on Graph Data and Knowledge* 1.1 (2023), pp. 1–35.

[51] Dai, Wenliang, Li, Junnan, Li, Dongxu, Tiong, Anthony Meng Huat, Zhao, Junqi, Wang, Weisheng, Li, Boyang, Fung, Pascale N, and Hoi, Steven. "Instructblip: Towards general-purpose vision-language models with instruction tuning". In: *Advances in Neural Information Processing Systems* 36 (2024).

[52] Dai, Wenliang, Liu, Zihan, Ji, Ziwei, Su, Dan, and Fung, Pascale. "Plausible May Not Be Faithful: Probing Object Hallucination in Vision-Language Pre-training". In: *ArXiv* abs/2210.07688 (2022). URL:

[53] Danilevsky, Marina, Qian, Kun, Aharonov, Ranit, Katsis, Yannis, Kawas, Ban, and Sen, Prithviraj. "A Survey of the State of Explainable AI for Natural Language Processing". In: *AACL*. 2020. URL:

[54] Del, Maksym and Fishel, Mark. "True Detective: A Deep Abductive Reasoning Benchmark Undoable for GPT-3 and Challenging for GPT-4". In: *STARSEM*. 2022. URL:

[55] Deldjoo, Yashar, He, Zhankui, McAuley, Julian, Korikov, Anton, Sanner, Scott, Ramisa, Arnau, Vidal, René, Sathiamoorthy, Maheswaran, Kasirzadeh, Atoosa, and Milano, Silvia. *A Review of Modern Recommender Systems Using Generative Models (Gen-RecSys)*. 2024. arXiv: 2404.00579 [cs.IR]. URL:

[56] Dervakos, Edmund, Filandrianos, Giorgos, and Stamou, Giorgos. "Heuristics for evaluation of AI generated music". In: *2020 25th international conference on pattern recognition (ICPR)*. IEEE. 2021, pp. 9164–9171.

[57] Dervakos, Edmund, Menis-Mastromichalakis, Orfeas, Chortaras, Alexandros, and Stamou, Giorgos. "Computing Rule-Based Explanations of Machine Learning Classifiers using Knowledge Graphs". In: *arXiv preprint arXiv:2202.03971* (2022).

[58] Dervakos, Edmund, Thomas, Konstantinos, Filandrianos, Giorgos, and Stamou, Giorgos. "Choose your Data Wisely: A Framework for Semantic Counterfactuals". In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. Ed. by Edith Elkind. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 382–390. DOI: 10.24963/ijcai.2023/43. URL:

[59] Dervakos, Edmund-Grigoris. "Knowledge Graph Based Explanation and Evaluation of Machine Learning Systems". In: (2024).

[60] Dervakosa, Edmund, Filandrianosa, Giorgos, Thomasa, Konstantinos, Mandaliosa, Alexios, Zervaa, Chrysoula, and Stamoua, Giorgos. "Semantic Enrichment of Pretrained Embedding Output for Unsupervised IR". In: (2021).

[61] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL:

[62] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL:

[63] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL:

[64] Dijkstra, Edsger W. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[65] Dimitriou, Angeliki, Lymperaiou, Maria, Filandrianos, Giorgos, Thomas, Konstantinos, and Stamou, Giorgos. *Structure Your Data: Towards Semantic Graph Counterfactuals*. 2024. arXiv: 2403.06514 [cs.CV]. URL:

[66] Ding, Ruomeng, Zhang, Chaoyun, Wang, Lu, Xu, Yong, Ma, Ming-Jie, Zhang, Wei, Qin, Si, Rajmohan, S., Lin, Qingwei, and Zhang, Dongmei. "Everything of Thoughts: Defying the Law of Penrose Triangle for Thought Generation". In: *ArXiv* abs/2311.04254 (2023). URL:

[67] Dong, Qingxiu, Li, Lei, Dai, Damai, Zheng, Ce, Ma, Jingyuan, Li, Rui, Xia, Heming, Xu, Jingjing, Wu, Zhiyong, Chang, Baobao, Sun, Xu, Li, Lei, and Sui, Zhifang. *A Survey on In-context Learning.* 2024. arXiv: 2301.00234 [cs.CL]. URL:

[68] Dong, Qingxiu, Li, Lei, Dai, Damai, Zheng, Ce, Wu, Zhiyong, Chang, Baobao, Sun, Xu, Xu, Jingjing, Li, Lei, and Sui, Zhifang. *A Survey on In-context Learning.* 2023. arXiv: 2301.00234 [cs.CL].

[69] Ebrahimi, Javid, Rao, Anyi, Lowd, Daniel, and Dou, Dejing. "HotFlip: White-Box Adversarial Examples for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 31–36. DOI: 10.18653/v1/P18-2006. URL:

[70] Echterhoff, Jessica Maria, Liu, Yao, Alessa, Abeer, McAuley, Julian, and He, Zexue. "Cognitive Bias in Decision-Making with LLMs". In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 12640–12653. DOI: 10.18653/v1/2024.findings-emnlp.739. URL:

[71] Efrat, Avia, Shaham, Uri, Kilman, Dan, and Levy, Omer. "Cryptonite: A Cryptic Crossword Benchmark for Extreme Ambiguity in Language". In: *ArXiv* abs/2103.01242 (2021). URL:

[72] Evangelatos, Andreas, Filandrianos, Giorgos, Lymperaiou, Maria, Voulodimos, Athanasios, and Stamou, Giorgos. "AILS-NTUA at SemEval-2025 Task 8: Language-to-Code prompting and Error Fixing for Tabular Question Answering". In: *arXiv preprint arXiv:2503.00435* (2025).

[73] Fang, Yuxin, Liao, Bencheng, Wang, Xinggang, Fang, Jiemin, Qi, Jiyang, Wu, Rui, Niu, Jianwei, and Liu, Wenyu. "You Only Look at One Sequence: Rethinking Transformer in Vision through Object Detection". In: *CoRR* abs/2106.00666 (2021). arXiv: 2106.00666. URL:

[74] Fankhauser, Stefan, Riesen, Kaspar, and Bunke, Horst. "Speeding up graph edit distance computation through fast bipartite matching". In: *International Workshop on Graph-Based Representations in Pattern Recognition*. Springer. 2011, pp. 102–111.

[75] Fellbaum, Christiane. "WordNet: An Electronic Lexical Database". In: (1998).

[76] Feng, Jiazhan, Xu, Ruochen, Hao, Junheng, Sharma, Hiteshi, Shen, Yelong, Zhao, Dongyan, and Chen, Weizhu. "Language Models can be Logical Solvers". In: *ArXiv* abs/2311.06158 (2023). URL:

[77] Feng, Xidong, Luo, Yicheng, Wang, Ziyan, Tang, Hongrui, Yang, Mengyue, Shao, Kun, Mguni, David Henry, Du, Yali, and Wang, Jun. "ChessGPT: Bridging Policy Learning and Language Modeling". In: *ArXiv* abs/2306.09200 (2023). URL:

[78] Feng, Zhangyin, Ren, Yuchen, Yu, Xinmiao, Feng, Xiaocheng, Tang, Duyu, Shi, Shuming, and Qin, Bing. "Improved Visual Story Generation with Adaptive Context Modeling". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 4939–4955. DOI: 10.18653/v1/2023.findings-acl.305. URL:

[79] Fern, Xiaoli and Pope, Quintin. "Text counterfactuals via latent optimization and shapley-guided search". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 5578–5593.

[80] Filandrianos, George, Dervakos, Edmund, Menis Mastromichalakis, Orfeas, Zerva, Chrysoula, and Stamou, Giorgos. "Counterfactuals of Counterfactuals: a back-translation-inspired approach to analyse counterfactual editors". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 9507–9525. DOI: 10.18653/v1/2023.findings-acl.606. URL:

[81] Filandrianos, George, Dervakos, Edmund, Menis Mastromichalakis, Orfeas, Zerva, Chrysoula, and Stamou, Giorgos. "Counterfactuals of Counterfactuals: a back-translation-inspired approach to analyse counterfactual editors". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 9507–9525. DOI: 10.18653/v1/2023.findings-acl.606. URL:

[82] Filandrianos, Giorgos, Dimitriou, Angeliki, Lymperaiou, Maria, Thomas, Konstantinos, and Stamou, Giorgos. "Bias Beware: The Impact of Cognitive Biases on LLM-Driven Product Recommendations". In: *arXiv preprint arXiv:2502.01349* (2025).

[83] Filandrianos, Giorgos, Kotsani, Natalia, Dervakos, Edmund G, Stamou, Giorgos, Amprazis, Vaios, and Kiourtzoglou, Panagiotis. “Brainwaves-driven Effects Automation in Musical Performance”. In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. 2020, pp. 545–546.

[84] Filandrianos, Giorgos, Thomas, Konstantinos, Dervakos, Edmund, and Stamou, Giorgos. “Conceptual Edits as Counterfactual Explanations”. In: *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering* (2022).

[85] Flach, Peter A. and Kakas, Antonis C. “Abductive and inductive reasoning: background and issues”. In: 2000. URL:

[86] Fu, Yao, Peng, Hao, Sabharwal, Ashish, Clark, Peter, and Khot, Tushar. *Complexity-Based Prompting for Multi-Step Reasoning*. 2023. arXiv: 2210.00720 [cs.CL]. URL:

[87] Fu, Yao, Peng, Hao-Chun, Sabharwal, Ashish, Clark, Peter, and Khot, Tushar. “Complexity-Based Prompting for Multi-Step Reasoning”. In: *ArXiv* abs/2210.00720 (2022). URL:

[88] Gao, Luyu, Madaan, Aman, Zhou, Shuyan, Alon, Uri, Liu, Pengfei, Yang, Yiming, Callan, Jamie, and Neubig, Graham. “PAL: Program-aided Language Models”. In: *ArXiv* abs/2211.10435 (2022). URL:

[89] Gao, Yunfan, Sheng, Tao, Xiang, Youlin, Xiong, Yun, Wang, Haofen, and Zhang, Jiawei. *Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System*. 2023. arXiv: 2303.14524 [cs.IR]. URL:

[90] Gardner, Matt, Artzi, Yoav, Basmov, Victoria, Berant, Jonathan, Bogin, Ben, Chen, Sihao, Dasigi, Pradeep, Dua, Dheeru, Elazar, Yanai, Gottumukkala, Ananth, Gupta, Nitish, Hajishirzi, Hannaneh, Ilharco, Gabriel, Khashabi, Daniel, Lin, Kevin, Liu, Jiangming, Liu, Nelson F., Mulcaire, Phoebe, Ning, Qiang, Singh, Sameer, Smith, Noah A., Subramanian, Sanjay, Tsarfaty, Reut, Wallace, Eric, Zhang, Ally, and Zhou, Ben. “Evaluating Models’ Local Decision Boundaries via Contrast Sets”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, Nov. 2020, pp. 1307–1323. DOI: 10.18653/v1/2020.findings-emnlp.117. URL:

[91] Garg, Siddhant and Ramakrishnan, Goutham. “BAE: BERT-based Adversarial Examples for Text Classification”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, Nov. 2020, pp. 6174–6181. DOI: 10.18653/v1/2020.emnlp-main.498. URL:

[92] Genovese, Angelo, Piuri, Vincenzo, and Scotti, Fabio. “Towards Explainable Face Aging with Generative Adversarial Networks”. In: *2019 IEEE International Conference on Image Processing (ICIP)*. 2019, pp. 3806–3810. DOI: 10.1109/ICIP.2019.8803616.

[93] Ghandi, Taraneh, Pourreza, Hamid Reza, and Mahyar, Hamidreza. “Deep Learning Approaches on Image Captioning: A Review”. In: *ACM Computing Surveys* 56 (2022), pp. 1–39. URL:

[94] Giadikiaroglou, Panagiotis, Lymperaiou, Maria, Filandrianos, Giorgos, and Stamou, Giorgos. “Puzzle Solving using Reasoning of Large Language Models: A Survey”. In: *arXiv preprint arXiv:2402.11291* (2024).

[95] Giadikiaroglou, Panagiotis, Lymperaiou, Maria, Filandrianos, Giorgos, and Stamou, Giorgos. *Puzzle Solving using Reasoning of Large Language Models: A Survey*. 2024. arXiv: 2402.11291 [cs.CL]. URL:

[96] Gilo, Daniel and Markovitch, Shaul. “A General Search-Based Framework for Generating Textual Counterfactual Explanations”. In: *AAAI Conference on Artificial Intelligence*. 2022. URL:

[97] Ginsberg, Matthew L. “Counterfactuals”. In: *Artificial intelligence* 30.1 (1986), pp. 35–79.

[98] Gomez, Oscar, Holter, Steffen, Yuan, Jun, and Bertini, Enrico. “Vice: Visual counterfactual explanations for machine learning models”. In: *Proceedings of the 25th international conference on intelligent user interfaces*. 2020, pp. 531–535.

[99] Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., 2014. URL:

[100] Goyal, Yash, Wu, Ziyan, Ernst, Jan, Batra, Dhruv, Parikh, Devi, and Lee, Stefan. “Counterfactual visual explanations”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2376–2384.

[101] Grattafiori, Aaron, Dubey, Abhimanyu, Jauhri, Abhinav, Pandey, Abhinav, Kadian, Abhishek, Al-Dahle, Ahmad, Letman, Aiesha, Mathur, Akhil, Schelten, Alan, Vaughan, Alex, Yang, Amy, Fan,

Angela, Goyal, Anirudh, Hartshorn, Anthony, Yang, Aobo, Mitra, Archi, Sravankumar, Archie, Korenev, Artem, Hinsvark, Arthur, Rao, Arun, Zhang, Aston, Rodriguez, Aurelien, Gregerson, Austen, Spataru, Ava, Roziere, Baptiste, Biron, Bethany, Tang, Binh, Chern, Bobbie, Caucheteux, Charlotte, Nayak, Chaya, Bi, Chloe, Marra, Chris, McConnell, Chris, Keller, Christian, Touret, Christophe, Wu, Chunyang, Wong, Corinne, Ferrer, Cristian Canton, Nikolaidis, Cyrus, Allonsius, Damien, Song, Daniel, Pintz, Danielle, Livshits, Danny, Wyatt, Danny, Esiobu, David, Choudhary, Dhruv, Mahajan, Dhruv, Garcia-Olano, Diego, Perino, Diego, Hupkes, Dieuwke, Lakomkin, Egor, AlBadawy, Ehab, Lobanova, Elina, Dinan, Emily, Smith, Eric Michael, Radenovic, Filip, Guzmán, Francisco, Zhang, Frank, Synnaeve, Gabriel, Lee, Gabrielle, Anderson, Georgia Lewis, Thattai, Govind, Nail, Graeme, Mialon, Gregoire, Pang, Guan, Cucurell, Guillem, Nguyen, Hailey, Korevaar, Hannah, Xu, Hu, Touvron, Hugo, Zarov, Iliyan, Ibarra, Imanol Arrieta, Kloumann, Isabel, Misra, Ishan, Evtimov, Ivan, Zhang, Jack, Copet, Jade, Lee, Jaewon, Geffert, Jan, Vranes, Jana, Park, Jason, Mahadeokar, Jay, Shah, Jeet, Linde, Jelmer van der, Billock, Jennifer, Hong, Jenny, Lee, Jenya, Fu, Jeremy, Chi, Jianfeng, Huang, Jianyu, Liu, Jiawen, Wang, Jie, Yu, Jiecao, Bitton, Joanna, Spisak, Joe, Park, Jongsoo, Rocca, Joseph, Johnstun, Joshua, Saxe, Joshua, Jia, Junteng, Alwala, Kalyan Vasuden, Prasad, Karthik, Upasani, Kartikeya, Plawiak, Kate, Li, Ke, Heafield, Kenneth, Stone, Kevin, El-Arini, Khalid, Iyer, Krithika, Malik, Kshitiz, Chiu, Kuenley, Bhalla, Kunal, Lakhotia, Kushal, Rantala-Yeary, Lauren, Maaten, Laurens van der, Chen, Lawrence, Tan, Liang, Jenkins, Liz, Martin, Louis, Madaan, Lovish, Malo, Lubo, Blecher, Lukas, Landzaat, Lukas, Oliveira, Luke de, Muzzi, Madeline, Pasupuleti, Mahesh, Singh, Mannat, Paluri, Manohar, Kardas, Marcin, Tsimpoukelli, Maria, Oldham, Mathew, Rita, Mathieu, Pavlova, Maya, Kambadur, Melanie, Lewis, Mike, Si, Min, Singh, Mitesh Kumar, Hassan, Mona, Goyal, Naman, Torabi, Narjes, Bashlykov, Nikolay, Bogoychev, Nikolay, Chatterji, Niladri, Zhang, Ning, Duchenne, Olivier, Çelebi, Onur, Alrassy, Patrick, Zhang, Pengchuan, Li, Pengwei, Vasic, Petar, Weng, Peter, Bhargava, Prajjwal, Dubal, Pratik, Krishnan, Praveen, Koura, Punit Singh, Xu, Puxin, He, Qing, Dong, Qingxiao, Srinivasan, Ragavan, Ganapathy, Raj, Calderer, Ramon, Cabral, Ricardo Silveira, Stojnic, Robert, Raileanu, Roberta, Maheswari, Rohan, Girdhar, Rohit, Patel, Rohit, Sauvestre, Romain, Polidoro, Ronnie, Sumbaly, Roshan, Taylor, Ross, Silva, Ruan, Hou, Rui, Wang, Rui, Hosseini, Saghar, Chennabasappa, Sahana, Singh, Sanjay, Bell, Sean, Kim, Seohyun Sonia, Edunov, Sergey, Nie, Shaoliang, Narang, Sharan, Raparthy, Sharath, Shen, Sheng, Wan, Shengye, Bhosale, Shruti, Zhang, Shun, Vandenhende, Simon, Batra, Soumya, Whitman, Spencer, Sootla, Sten, Collot, Stephane, Gururangan, Suchin, Borodinsky, Sydney, Herman, Tamar, Fowler, Tara, Sheasha, Tarek, Georgiou, Thomas, Scialom, Thomas, Speckbacher, Tobias, Mihaylov, Todor, Xiao, Tong, Karn, Ujjwal, Goswami, Vedanuj, Gupta, Vibhor, Ramanathan, Vignesh, Kerkez, Viktor, Gonguet, Vincent, Do, Virginie, Vogeti, Vish, Albiero, Vítor, Petrovic, Vladan, Chu, Weiwei, Xiong, Wenhan, Fu, Wenyin, Meers, Whitney, Martinet, Xavier, Wang, Xiaodong, Wang, Xiaofang, Tan, Xiaoqing Ellen, Xia, Xide, Xie, Xinfeng, Jia, Xuchao, Wang, Xuewei, Goldschlag, Yaelle, Gaur, Yashesh, Babaei, Yasmine, Wen, Yi, Song, Yiwen, Zhang, Yuchen, Li, Yue, Mao, Yuning, Coudert, Zacharie Delpierre, Yan, Zheng, Chen, Zhengxing, Papakipos, Zoe, Singh, Aaditya, Srivastava, Aayushi, Jain, Abha, Kelsey, Adam, Shajnfeld, Adam, Gangidi, Adithya, Victoria, Adolfo, Goldstand, Ahuva, Menon, Ajay, Sharma, Ajay, Boesenberg, Alex, Baevski, Alexei, Feinstein, Allie, Kallet, Amanda, Sangani, Amit, Teo, Amos, Yunus, Anam, Lupu, Andrei, Alvarado, Andres, Caples, Andrew, Gu, Andrew, Ho, Andrew, Poulton, Andrew, Ryan, Andrew, Ramchandani, Ankit, Dong, Annie, Franco, Annie, Goyal, Anuj, Saraf, Aparajita, Chowdhury, Arkabandhu, Gabriel, Ashley, Bharambe, Ashwin, Eisenman, Assaf, Yazdan, Azadeh, James, Beau, Maurer, Ben, Leonhardi, Benjamin, Huang, Bernie, Loyd, Beth, Paola, Beto De, Paranjape, Bhargavi, Liu, Bing, Wu, Bo, Ni, Boyu, Hancock, Braden, Wasti, Bram, Spence, Brandon, Stojkovic, Brani, Gamido, Brian, Montalvo, Britt, Parker, Carl, Burton, Carly, Mejia, Catalina, Liu, Ce, Wang, Changhan, Kim, Changkyu, Zhou, Chao, Hu, Chester, Chu, Ching-Hsiang, Cai, Chris, Tindal, Chris, Feichtenhofer, Christoph, Gao, Cynthia, Civin, Damon, Beaty, Dana, Kreymer, Daniel, Li, Daniel, Adkins, David, Xu, David, Testuggine, Davide, David, Delia, Parikh, Devi, Liskovich, Diana, Foss, Didem, Wang, Dingkang, Le, Duc, Holland, Dustin, Dowling, Edward, Jamil, Eissa, Montgomery, Elaine, Presani, Eleonora, Hahn, Emily, Wood, Emily, Le, Eric-Tuan, Brinkman, Erik, Arcaute, Esteban, Dunbar, Evan, Smothers, Evan, Sun, Fei, Kreuk, Felix, Tian, Feng, Kokkinos, Filippos, Ozgenel, Firat, Caggioni, Francesco, Kanayet, Frank, Seide, Frank, Florez, Gabriela Medina, Schwarz, Gabriella, Badeer, Gada, Swee, Georgia, Halpern, Gil, Herman, Grant, Sizov, Grigory, Guangyi, Zhang, Lakshminarayanan, Guna, Inan, Hakan, Shojanazeri, Hamid, Zou, Han, Wang, Hannah, Zha, Hanwen, Habeeb, Haroun, Rudolph, Harrison, Suk, Helen, Aspegren, Henry, Goldman, Hunter, Zhan, Hongyuan, Damlaj, Ibrahim, Molybog, Igor, Tufanov, Igor, Leontiadis, Ilias, Veliche, Irina-Elena, Gat, Itai, Weissman, Jake, Geboski, James, Kohli, James, Lam, Janice, Asher, Japhet, Gaya, Jean-Baptiste, Marcus, Jeff, Tang, Jeff, Chan, Jennifer, Zhen, Jenny, Reizenstein, Jeremy, Teboul, Jeremy, Zhong, Jessica, Jin, Jian, Yang, Jingyi, Cummings, Joe, Carvill, Jon, Shepard, Jon, McPhie, Jonathan, Torres, Jonathan, Ginsburg, Josh, Wang, Junjie, Wu, Kai, U, Kam Hou, Saxena, Karan, Khandelwal, Kartikay, Zand, Katayoun, Matosich, Kathy, Veeraraghavan, Kaushik, Michelena, Kelly, Li, Keqian, Jagadeesh, Kiran, Huang, Kun, Chawla, Kunal, Huang, Kyle, Chen, Lailin, Garg, Lakshya, A, Lavender, Silva, Leandro, Bell, Lee, Zhang, Lei, Guo, Liangpeng, Yu, Licheng, Moshkovich, Liron, Wehrstedt, Luca, Khabsa, Madian, Avalani, Manav, Bhatt, Manish, Mankus, Martynas, Hasson, Matan, Lennie, Matthew, Reso, Matthias, Groshev, Maxim, Naumov, Maxim, Lathi, Maya, Keneally, Meghan, Liu, Miao, Seltzer, Michael L., Valko, Michal, Restrepo, Michelle, Patel, Mihir, Vyatskov, Mik, Samvelyan, Mikayel, Clark, Mike, Macey, Mike, Wang, Mike, Hermoso, Miquel Jubert, Metanat, Mo, Rastegari, Mohammad, Bansal, Munish, Santhanam, Nandhini, Parks, Natascha, White, Natasha, Bawa, Navyata, Singhal, Nayan, Egebo, Nick, Usunier, Nicolas, Mehta, Nikhil, Laptev, Nikolay Pavlovich, Dong, Ning, Cheng, Norman, Chernoguz, Oleg, Hart, Olivia, Salpekar, Omkar, Kalinli, Ozlem, Kent, Parkin, Parekh, Parth, Saab, Paul, Balaji, Pavan, Rittner, Pedro, Bontrager, Philip, Roux, Pierre, Dollar, Piotr, Zvyagina, Polina, Ratanchandani, Prashant, Yuvraj, Pritish, Liang, Qian, Alao, Rachad, Rodriguez, Rachel, Ayub, Rafi, Murthy, Raghotham, Nayani, Raghu, Mitra, Rahul, Parthasarathy, Rangaprabhu, Li, Raymond, Hogan, Rebekkah, Battey, Robin, Wang, Rocky, Howes, Russ, Rinott, Ruty, Mehta, Sachin, Siby, Sachin, Bondu, Sai Jayesh, Datta, Samyak, Chugh, Sara, Hunt, Sara, Dhillon, Sargun, Sidorov, Sasha, Pan, Satadru, Mahajan, Saurabh, Verma, Saurabh, Yamamoto, Seiji, Ramaswamy, Sharadh, Lindsay, Shaun, Lindsay, Shaun, Feng, Sheng, Lin, Shenghao, Zha, Shengxin Cindy, Patil, Shishir, Shankar, Shiva, Zhang, Shuqiang, Zhang, Shuqiang, Wang, Sinong, Agarwal, Sneha, Sajuyigbe, Soji, Chintala, Soumith, Max, Stephanie, Chen, Stephen, Kehoe, Steve, Satterfield, Steve, Govindaprasad, Sudarshan, Gupta, Sumit, Deng, Summer, Cho, Sungmin, Virk, Sunny, Subramanian, Suraj, Choudhury, Sy, Goldman, Sydney, Remez, Tal, Glaser, Tamar, Best, Tamara, Koehler, Thilo, Robinson, Thomas, Li, Tianhe, Zhang, Tianjun, Matthews, Tim, Chou, Timothy, Shaked, Tzook, Vontimitta, Varun, Ajayi, Victoria, Montanez, Victoria, Mohan, Vijai, Kumar, Vinay Satish, Mangla, Vishal, Ionescu, Vlad, Poenaru, Vlad, Mihailescu, Vlad Tiberiu, Ivanov, Vladimir, Li, Wei, Wang, Wenchen, Jiang, Wenwen, Bouaziz, Wes, Constable, Will, Tang, Xiaocheng, Wu, Xiaojian, Wang, Xiaolan, Wu, Xilun, Gao, Xinbo, Kleinman, Yaniv, Chen, Yanjun, Hu, Ye, Jia, Ye, Qi, Ye, Li, Yenda, Zhang, Yilin, Zhang, Ying, Adi, Yossi, Nam, Youngjin, Yu, Wang, Zhao, Yu, Hao, Yuchen, Qian, Yundi, Li, Yunlu, He, Yuzi, Rait, Zach, DeVito, Zachary, Rosnbrick, Zef, Wen, Zhaoduo, Yang, Zhenyu, Zhao, Zhiwei, and Ma, Zhiyu. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI].

[102] Greshake, Kai, Abdelnabi, Sahar, Mishra, Shailesh, Endres, Christoph, Holz, Thorsten, and Fritz, Mario. "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection". In: *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*. AISec '23. Copenhagen, Denmark: Association for Computing Machinery, 2023, pp. 79–90. ISBN: 9798400702600. DOI: 10.1145/3605764.3623985. URL:

[103] Grigoriadou, Natalia, Lymperaiou, Maria, Filandrianos, George, and Stamou, Giorgos. "AILS-NTUA at SemEval-2024 Task 6: Efficient model tuning for hallucination detection and analysis". In: *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Ed. by Atul Kr. Ojha, A. Seza Doğruöz, Harish Tayyar Madabushi, Giovanni Da San Martino, Sara Rosenthal, and Aiala Rosá. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 1549–1560. URL:

[104] Grigoriadou, Natalia, Lymperaiou, Maria, Filandrianos, George, and Stamou, Giorgos. "AILS-NTUA at SemEval-2024 Task 6: Efficient model tuning for hallucination detection and analysis". In: *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. 2024, pp. 1549–1560.

[105] Gu, Zhouhong, Li, Zihan, Zhang, Lin, Xiong, Zhuozhi, Jiang, Sihang, Zhu, Xiaoxuan, Wang, Shusen, Wang, Zili, Wang, Jianchen, Ye, Haoning, Huang, Wenhao, Zhang, Yikai, Feng, Hongwei, and Xiao, Yanghua. "Go Beyond The Obvious: Probing the gap of INFORMAL reasoning ability between Humanity and LLMs by Detective Reasoning Puzzle Benchmark". In: 2023. URL:

[106] Guidotti, Riccardo. “Counterfactual explanations and how to find them: literature review and benchmarking”. In: *Data Mining and Knowledge Discovery* (2022), pp. 1–55.

[107] Guidotti, Riccardo, Monreale, Anna, Ruggieri, Salvatore, Turini, Franco, Giannotti, Fosca, and Pedreschi, Dino. “A survey of methods for explaining black box models”. In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[108] Guo, Jiaxian, Yang, Bo, Yoo, Paul, Lin, Bill Yuchen, Iwasawa, Yusuke, and Matsuo, Yutaka. “Suspicion-Agent: Playing Imperfect Information Games with Theory of Mind Aware GPT-4”. In: *ArXiv* abs/2309.17277 (2023). URL:

[109] Gupta, Akshat. “Are ChatGPT and GPT-4 Good Poker Players? - A Pre-Flop Analysis”. In: *ArXiv* abs/2308.12466 (2023). URL:

[110] Hagberg, Aric and Conway, Drew. “Networkx: Network analysis with python”. In: *URL: https://networkx. github. io* (2020).

[111] Hagendorff, Thilo, Dasgupta, Ishita, Binz, Marcel, Chan, Stephanie C. Y., Lampinen, Andrew, Wang, Jane X., Akata, Zeynep, and Schulz, Eric. *Machine Psychology.* 2024. arXiv: 2303.13988 [cs.CL]. URL:

[112] Han, Simon Jerome, Ransom, Keith J., Perfors, Andrew, and Kemp, Charles. “Inductive reasoning in humans and large language models”. In: *Cognitive Systems Research* 83 (2024), p. 101155. ISSN: 1389-0417. DOI: https://doi.org/10.1016/j.cogsys.2023.101155. URL:

[113] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778.

[114] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778.

[115] Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Advances in Neural Information Processing Systems.* Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL:

[116] Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium.* 2018. arXiv: 1706.08500 [cs.LG].

[117] Ho, Jonathan, Jain, Ajay, and Abbeel, Pieter. *Denoising Diffusion Probabilistic Models.* 2020. DOI: 10.48550/ARXIV.2006.11239. URL:

[118] Horé, Alain and Ziou, Djemel. “Image Quality Metrics: PSNR vs. SSIM”. In: *2010 20th International Conference on Pattern Recognition.* 2010, pp. 2366–2369. DOI: 10.1109/ICPR.2010.579.

[119] Hou, Yupeng, Li, Jiacheng, He, Zhankui, Yan, An, Chen, Xiusi, and McAuley, Julian. “Bridging Language and Items for Retrieval and Recommendation”. In: *arXiv preprint arXiv:2403.03952* (2024).

[120] Hu, Weixiong, Gu, Zhaoquan, Zhang, Chuanjing, Wang, Le, and Tang, Keke. “Adversarial Examples Generation System Based on Gradient Shielding of Restricted Region”. In: *Artificial Intelligence and Security: 6th International Conference, ICAIS 2020, Hohhot, China, July 17–20, 2020, Proceedings, Part III 6.* Springer. 2020, pp. 81–91.

[121] Huang, Chenghao, Cao, Yanbo, Wen, Yinlong, Zhou, Tao, and Zhang, Yanru. “PokerGPT: An End-to-End Lightweight Solver for Multi-Player Texas Hold’em via Large Language Model”. In: *ArXiv* abs/2401.06781 (2024). URL:

[122] Huang, Jie and Chang, Kevin Chen-Chuan. “Towards Reasoning in Large Language Models: A Survey”. In: *ArXiv* abs/2212.10403 (2022). URL:

[123] Huang, Lei, Yu, Weijiang, Ma, Weitao, Zhong, Weihong, Feng, Zhangyin, Wang, Haotian, Chen, Qianglong, Peng, Weihua, Feng, Xiaocheng, Qin, Bing, and Liu, Ting. *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions.* 2023. arXiv: 2311.05232 [cs.CL].

[124] Huang, Shulin, Ma, Shirong, Li, Yinghui, Huang, Mengzuo, Zou, Wuhe, Zhang, Weidong, and Zheng, Haitao. “LatEval: An Interactive LLMs Evaluation Benchmark with Incomplete Information from Lateral Thinking Puzzles”. In: *ArXiv* abs/2308.10855 (2023). URL:

[125] *Industrial-Strength Natural Language Processing.* spaCy. URL:

[126] Ishay, Adam, Yang, Zhun, and Lee, Joohyung. "Leveraging Large Language Models to Generate Answer Set Programs". In: *ArXiv* abs/2307.07699 (2023). URL:

[127] Iyyer, Mohit, Wieting, John, Gimpel, Kevin, and Zettlemoyer, Luke. "Adversarial Example Generation with Syntactically Controlled Paraphrase Networks". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1875–1885. DOI: 10.18653/v1/N18-1170. URL:

[128] Jia, Robin and Liang, Percy. "Adversarial Examples for Evaluating Reading Comprehension Systems". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2021–2031. DOI: 10.18653/v1/D17-1215. URL:

[129] Jiang, Albert Q., Sablayrolles, Alexandre, Mensch, Arthur, Bamford, Chris, Chaplot, Devendra Singh, Casas, Diego de las, Bressand, Florian, Lengyel, Gianna, Lample, Guillaume, Saulnier, Lucile, Lavaud, Lélio Renard, Lachaux, Marie-Anne, Stock, Pierre, Scao, Teven Le, Lavril, Thibaut, Wang, Thomas, Lacroix, Timothée, and Sayed, William El. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL]. URL:

[130] Jiang, Yifan, Ilievski, Filip, and Ma, Kaixin. "BRAINTEASER: Lateral Thinking Puzzles for Large Language Models". In: *Conference on Empirical Methods in Natural Language Processing*. 2023. URL:

[131] Jiang, Yifan, Ilievski, Filip, Ma, Kaixin, and Sourati, Zhivar. "BRAINTEASER: Lateral Thinking Puzzles for Large Language Models". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 14317–14332. DOI: 10.18653/v1/2023.emnlp-main.885. URL:

[132] Jin, Di, Jin, Zhijing, Zhou, Joey Tianyi, and Szolovits, Peter. *Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment*. 2020. arXiv: 1907.11932 [cs.CL]. URL:

[133] Jin, Di, Jin, Zhijing, Zhou, Joey Tianyi, and Szolovits, Peter. "Is bert really robust? a strong baseline for natural language attack on text classification and entailment". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 05. 2020, pp. 8018–8025.

[134] Jin, Haibo, Chen, Ruoxi, Zhou, Andy, Zhang, Yang, and Wang, Haohan. *GUARD: Role-playing to Generate Natural-language Jailbreakings to Test Guideline Adherence of Large Language Models*. 2024. arXiv: 2402.03299 [cs.LG]. URL:

[135] Jing, Liqiang, Li, Ruosen, Chen, Yunmo, Jia, Mengzhao, and Du, Xinya. *FAITHSCORE: Evaluating Hallucinations in Large Vision-Language Models*. 2023. arXiv: 2311.01477 [cs.CV].

[136] Jing, Liqiang, Li, Ruosen, Chen, Yunmo, Jia, Mengzhao, and Du, Xinya. "FAITHSCORE: Evaluating Hallucinations in Large Vision-Language Models". In: *ArXiv* abs/2311.01477 (2023). URL:

[137] Jocher, Glenn, Chaurasia, Ayush, and Qiu, Jing. *YOLO by Ultralytics*. Jan. 2023. URL:

[138] John, Vineet, Mou, Lili, Bahuleyan, Hareesh, and Vechtomova, Olga. "Disentangled representation learning for non-parallel text style transfer". In: *arXiv preprint arXiv:1808.04339* (2018).

[139] Johnson, Justin, Hariharan, Bharath, Maaten, Laurens van der, Fei-Fei, Li, Zitnick, C. Lawrence, and Girshick, Ross B. "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 1988–1997.

[140] Jonker, Roy and Volgenant, Anton. "A shortest augmenting path algorithm for dense and sparse linear assignment problems". In: *Computing* 38.4 (1987), pp. 325–340.

[141] Karkani, Dimitra, Lymperaiou, Maria, Filandrianos, Giorgos, Spanos, Nikolaos, Voulodimos, Athanasios, and Stamou, Giorgos. "AILS-NTUA at SemEval-2025 Task 3: Leveraging Large Language Models and Translation Strategies for Multilingual Hallucination Detection". In: *arXiv preprint arXiv:2503.02442* (2025).

[142] Karp, R.M. *An Algorithm to Solve the mxn Assignment Problem in Expected Time O (mn log n)*. Tech. rep. UCB/ERL M78/67. EECS Department, University of California, Berkeley, Sept. 1978. URL:

[143] Karp, Richard M. "An algorithm to solve the m× n assignment problem in expected time O (mn log n)". In: *Networks* 10.2 (1980), pp. 143–152.

[144] Karras, Tero, Aittala, Miika, Laine, Samuli, Härkönen, Erik, Hellsten, Janne, Lehtinen, Jaakko, and Aila, Timo. "Alias-free generative adversarial networks". In: *Advances in neural information processing systems* 34 (2021), pp. 852–863.

[145] Karras, Tero, Laine, Samuli, and Aila, Timo. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.

[146] Karras, Tero, Laine, Samuli, Aittala, Miika, Hellsten, Janne, Lehtinen, Jaakko, and Aila, Timo. "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8110–8119.

[147] Kaushik, Divyansh, Hovy, Eduard, and Lipton, Zachary C. *Learning the Difference that Makes a Difference with Counterfactually-Augmented Data*. 2020. arXiv: 1909.12434 [cs.CL]. URL:

[148] Kazani, Aggeliki, Filandrianos, George, Symeonaki, Maria, and Stamou, Giorgos. "Semantic integration of data: From theory to social research practice". In: *Quantitative demography and health estimates: Healthy life expectancy, templates for direct estimates from life tables and other applications*. Springer, 2023, pp. 303–314.

[149] Kazemi, Mehran, Yuan, Quan, Bhatia, Deepti, Kim, Najoung, Xu, Xin, Imbrasaite, Vaiva, and Ramachandran, Deepak. "BoardgameQA: A Dataset for Natural Language Reasoning with Contradictory Information". In: *ArXiv* abs/2306.07934 (2023). URL:

[150] Keane, Mark T, Kenny, Eoin M, Delaney, Eoin, and Smyth, Barry. "If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques". In: *arXiv preprint arXiv:2103.01035* (2021).

[151] Ketsekioulafis, Ioannis, Filandrianos, Giorgos, Katsos, Konstantinos, Thomas, Konstantinos, Spiliopoulou, Chara, Stamou, Giorgos, and Sakelliadis, Emmanouil I. "Artificial Intelligence in Forensic Sciences: A Systematic Review of Past and Current Applications and Future Perspectives". In: *Cureus* 16.9 (2024).

[152] Kim, Jiha and Park, Hyunhee. "Limited Discriminator GAN using explainable AI model for overfitting problem". In: *ICT Express* (2022). ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2021.12.014. URL:

[153] Kim, Jiwook and Lee, Minhyeok. *Class-Continuous Conditional Generative Neural Radiance Field*. 2023. DOI: 10.48550/ARXIV.2301.00950. URL:

[154] Kipf, Thomas N and Welling, Max. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[155] Kipf, Thomas N. and Welling, Max. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations (ICLR)*. 2017.

[156] Kipf, Thomas N. and Welling, Max. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations*. 2017. URL:

[157] Kojima, Takeshi, Gu, Shixiang Shane, Reid, Machel, Matsuo, Yutaka, and Iwasawa, Yusuke. "Large Language Models are Zero-Shot Reasoners". In: *ArXiv* abs/2205.11916 (2022). URL:

[158] Kojima, Takeshi, Gu, Shixiang Shane, Reid, Machel, Matsuo, Yutaka, and Iwasawa, Yusuke. *Large Language Models are Zero-Shot Reasoners*. 2023. arXiv: 2205.11916 [cs.CL]. URL:

[159] Koo, Ryan, Lee, Minhwa, Raheja, Vipul, Park, Jong Inn, Kim, Zae Myung, and Kang, Dongyeop. "Benchmarking Cognitive Biases in Large Language Models as Evaluators". In: *Findings of the Association for Computational Linguistics: ACL 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 517–545. DOI: 10.18653/v1/2024.findings-acl.29. URL:

[160] Koulakos, Alexandros, Lymperaiou, Maria, Filandrianos, Giorgos, and Stamou, Giorgos. "Enhancing adversarial robustness in Natural Language Inference using explanations". In: *The 7th BlackboxNLP Workshop*.

[161] Krishna, Ranjay, Zhu, Yuke, Groth, Oliver, Johnson, Justin, Hata, Kenji, Kravitz, Joshua, Chen, Stephanie, Kalantidis, Yannis, Li, Li-Jia, Shamma, David A, et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: *International journal of computer vision* 123.1 (2017), pp. 32–73.

[162] Kuhn, H. W. "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97. DOI: https://doi.org/10.1002/nav.3800020109. eprint: URL:

[163] Kuhn, Lorenz, Gal, Yarin, and Farquhar, Sebastian. "Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation". In: *The Eleventh International Conference on Learning Representations*.

[164] Kulshreshtha, Saurabh, Kovaleva, Olga, Shivagunde, Namrata, and Rumshisky, Anna. "Down and Across: Introducing Crossword-Solving as a New NLP Benchmark". In: *ArXiv* abs/2205.10442 (2022). URL:

[165] Kumar, Aounon and Lakkaraju, Himabindu. *Manipulating Large Language Models to Increase Product Visibility*. 2024. arXiv: 2404.07981 [cs.IR]. URL:

[166] Lan, Yihuai, Hu, Zhiqiang, Wang, Lei, Wang, Yang, Ye, De-Yong, Zhao, Peilin, Lim, Ee-Peng, Xiong, Hui, and Wang, Hao. "LLM-Based Agent Society Investigation: Collaboration and Confrontation in Avalon Gameplay". In: *ArXiv* abs/2310.14985 (2023). URL:

[167] Lan, Zhenzhong, Chen, Mingda, Goodman, Sebastian, Gimpel, Kevin, Sharma, Piyush, and Soricut, Radu. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *ArXiv* abs/1909.11942 (2019). URL:

[168] Lang, Ken. "NewsWeeder: learning to filter netnews". In: *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*. ICML'95. Tahoe City, California, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 331–339. ISBN: 1558603778.

[169] Lang, Ken. "Newsweeder: Learning to filter netnews". In: *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 331–339.

[170] Lee, Jinhyuk, Yoon, Wonjin, Kim, Sungdong, Kim, Donghyeon, Kim, Sunkyu, So, Chan Ho, and Kang, Jaewoo. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (Sept. 2019). Ed. by Jonathan Wren, pp. 1234–1240. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btz682. URL:

[171] Lei, Bin, Lin, Pei-Hung, Liao, Chunhua, and Ding, Caiwen. "Boosting Logical Reasoning in Large Language Models through a New Framework: The Graph of Thought". In: *ArXiv* abs/2308.08614 (2023). URL:

[172] Leiter, Christoph, Lertvittayakumjorn, Piyawat, Fomicheva, Marina, Zhao, Wei, Gao, Yang, and Eger, Steffen. *Towards Explainable Evaluation Metrics for Natural Language Generation*. 2022. DOI: 10.48550/ARXIV.2203.11131. URL:

[173] Leng, Sicong, Zhang, Hang, Chen, Guanzheng, Li, Xin, Lu, Shijian, Miao, Chunyan, and Bing, Lidong. *Mitigating Object Hallucinations in Large Vision-Language Models through Visual Contrastive Decoding*. 2023. arXiv: 2311.16922 [cs.CV].

[174] Levy, Mosh, Jacoby, Alon, and Goldberg, Yoav. *Same Task, More Tokens: the Impact of Input Length on the Reasoning Performance of Large Language Models*. 2024. arXiv: 2402.14848 [cs.CL]. URL:

[175] Lewis, Mike, Liu, Yinhan, Goyal, Naman, Ghazvininejad, Marjan, Mohamed, Abdel-rahman, Levy, Omer, Stoyanov, Veselin, and Zettlemoyer, Luke. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Annual Meeting of the Association for Computational Linguistics*. 2019. URL:

[176] Lewis, Mike, Liu, Yinhan, Goyal, Naman, Ghazvininejad, Marjan, Mohamed, Abdelrahman, Levy, Omer, Stoyanov, Ves, and Zettlemoyer, Luke. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL]. URL:

[177] Li, Dianqi, Zhang, Yizhe, Peng, Hao, Chen, Liqun, Brockett, Chris, Sun, Ming-Ting, and Dolan, Bill. "Contextualized Perturbation for Textual Adversarial Attack". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Online: Association for Computational Linguistics, June 2021, pp. 5053–5069. DOI: 10.18653/v1/2021.naacl-main.400. URL:

[178] Li, Jinming, Zhang, Wentao, Wang, Tian, Xiong, Guanglei, Lu, Alan, and Medioni, Gerard. *GPT4Rec: A Generative Framework for Personalized Recommendation and User Interests Interpretation*. 2023. arXiv: 2304.03879 [cs.IR]. URL:

[179] Li, Jiwei, Galley, Michel, Brockett, Chris, Gao, Jianfeng, and Dolan, Bill. "A Diversity-Promoting Objective Function for Neural Conversation Models". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Tech-*

nologies. Ed. by Kevin Knight, Ani Nenkova, and Owen Rambow. San Diego, California: Association for Computational Linguistics, June 2016, pp. 110–119. DOI: 10.18653/v1/N16-1014. URL:

[180] Li, Junnan, Li, Dongxu, Savarese, Silvio, and Hoi, Steven C. H. "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". In: *International Conference on Machine Learning*. 2023. URL:

[181] Li, Junnan, Li, Dongxu, Xiong, Caiming, and Hoi, Steven. "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation". In: *ICML*. 2022.

[182] Li, Linyang, Ma, Ruotian, Guo, Qipeng, Xue, Xiangyang, and Qiu, Xipeng. *BERT-ATTACK: Adversarial Attack Against BERT Using BERT*. 2020. arXiv: 2004.09984 [cs.CL]. URL:

[183] Li, Linyang, Ma, Ruotian, Guo, Qipeng, Xue, Xiangyang, and Qiu, Xipeng. "Bert-attack: Adversarial attack against bert using bert". In: *arXiv preprint arXiv:2004.09984* (2020).

[184] Li, Yandong, Cheng, Yu, Gan, Zhe, Yu, Licheng, Wang, Liqiang, and Liu, Jingjing. *BachGAN: High-Resolution Image Synthesis from Salient Object Layout*. 2020. arXiv: 2003.11690 [cs.CV].

[185] Li, Yifan, Du, Yifan, Zhou, Kun, Wang, Jinpeng, Zhao, Wayne Xin, and Wen, Ji-rong. "Evaluating Object Hallucination in Large Vision-Language Models". In: *ArXiv* abs/2305.10355 (2023). URL:

[186] Li, Yinghao, Wang, Haorui, and Zhang, Chao. "Assessing Logical Puzzle Solving in Large Language Models: Insights from a Minesweeper Case Study". In: *ArXiv* abs/2311.07387 (2023). URL:

[187] Li, Yitong, Gan, Zhe, Shen, Yelong, Liu, Jingjing, Cheng, Yu, Wu, Yuexin, Carin, Lawrence, Carlson, David Edwin, and Gao, Jianfeng. "StoryGAN: A Sequential Conditional GAN for Story Visualization". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 6322–6331.

[188] Li, Yongqi, Lin, Xinyu, Wang, Wenjie, Feng, Fuli, Pang, Liang, Li, Wenjie, Nie, Liqiang, He, Xiangnan, and Chua, Tat-Seng. *A Survey of Generative Search and Recommendation in the Era of Large Language Models*. 2024. arXiv: 2404.16924 [cs.IR]. URL:

[189] Li, Yongqi, Xu, Mayi, Miao, Xin, Zhou, Shen, and Qian, Tieyun. "Prompting Large Language Models for Counterfactual Generation: An Empirical Study". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Ed. by Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue. Torino, Italia: ELRA and ICCL, May 2024, pp. 13201–13221. URL:

[190] Li, Zekun, Peng, Baolin, He, Pengcheng, and Yan, Xifeng. *Evaluating the Instruction-Following Robustness of Large Language Models to Prompt Injection*. 2023. arXiv: 2308.10819 [cs.CL]. URL:

[191] Li, Zewen, Liu, Fan, Yang, Wenjie, Peng, Shouheng, and Zhou, Jun. "A survey of convolutional neural networks: analysis, applications, and prospects". In: *IEEE transactions on neural networks and learning systems* 33.12 (2021), pp. 6999–7019.

[192] Liartis, Jason, Dervakos, Edmund, Menis-Mastromichalakis, Orfeas, Chortaras, Alexandros, and Stamou, Giorgos. "Semantic Queries Explaining Opaque Machine Learning Classifiers". In: *DAO-XAI*. Vol. 2998. CEUR Workshop Proceedings. CEUR-WS.org, 2021.

[193] Liartis, Jason, Dervakos, Edmund, Menis-Mastromichalakis, Orfeas, Chortaras, Alexandros, and Stamou, Giorgos. "Searching for explanations of black-box classifiers in the space of semantic queries". In: *Semantic Web* Preprint (2023), pp. 1–42.

[194] Lin, Bill Yuchen, Wu, Ziyi, Yang, Yichi, Lee, Dong-Ho, and Ren, Xiang. "RiddleSense: Reasoning about Riddle Questions Featuring Linguistic Creativity and Commonsense Knowledge". In: *Findings*. 2021. URL:

[195] Lin, Bill Yuchen, Wu, Ziyi, Yang, Yichi, Lee, Dong-Ho, and Ren, Xiang. "RiddleSense: Reasoning about Riddle Questions Featuring Linguistic Creativity and Commonsense Knowledge". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, Aug. 2021, pp. 1504–1515. DOI: 10.18653/v1/2021.findings-acl.131. URL:

[196] Lin, Chin-Yew. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL:

[197] Lin, Jianghao, Dai, Xinyi, Xi, Yunjia, Liu, Weiwen, Chen, Bo, Zhang, Hao, Liu, Yong, Wu, Chuhan, Li, Xiangyang, Zhu, Chenxu, Guo, Huifeng, Yu, Yong, Tang, Ruiming, and Zhang, Weinan. *How Can Recommender Systems Benefit from Large Language Models: A Survey*. 2024. arXiv: 2306.05817 [cs.IR]. URL:

[198] Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[199] Lin, Tsung-Yi, Maire, Michael, Belongie, Serge J., Bourdev, Lubomir D., Girshick, Ross B., Hays, James, Perona, Pietro, Ramanan, Deva, Doll’a r, Piotr, and Zitnick, C. Lawrence. “Microsoft COCO: Common Objects in Context”. In: *CoRR* abs/1405.0312 (2014). arXiv: 1405.0312. URL:

[200] Liu, Fuxiao, Lin, Kevin, Li, Linjie, Wang, Jianfeng, Yacoob, Yaser, and Wang, Lijuan. *Mitigating Hallucination in Large Multi-Modal Models via Robust Instruction Tuning*. 2023. arXiv: 2306.14565 [cs.CV].

[201] Liu, Hanmeng, Ning, Ruoxi, Teng, Zhiyang, Liu, Jian, Zhou, Qiji, and Zhang, Yuexin. “Evaluating the Logical Reasoning Ability of ChatGPT and GPT-4”. In: *ArXiv* abs/2304.03439 (2023). URL:

[202] Liu, Hanmeng, Teng, Zhiyang, Ning, Ruoxi, Liu, Jian, Zhou, Qiji, and Zhang, Yuexin. “GLoRE: Evaluating Logical Reasoning of Large Language Models”. In: *ArXiv* abs/2310.09107 (2023). URL:

[203] Liu, Haotian, Li, Chunyuan, Li, Yuheng, and Lee, Yong Jae. *Improved Baselines with Visual Instruction Tuning*. 2023.

[204] Liu, Haotian, Li, Chunyuan, Wu, Qingyang, and Lee, Yong Jae. *Visual Instruction Tuning*. 2023. arXiv: 2304.08485 [cs.CV].

[205] Liu, He, Wang, Tao, Lang, Congyan, Feng, Songhe, Jin, Yi, and Li, Yidong. “GLAN: A graph-based linear assignment network”. In: *Pattern Recognition* 155 (June 2024), p. 110694. DOI: 10.1016/j.patcog.2024.110694.

[206] Liu, Jiachang, Shen, Dinghan, Zhang, Yizhe, Dolan, Bill, Carin, Lawrence, and Chen, Weizhu. “What Makes Good In-Context Examples for GPT-3?” In: *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Ed. by Eneko Agirre, Marianna Apidianaki, and Ivan Vulić. Dublin, Ireland and Online: Association for Computational Linguistics, May 2022, pp. 100–114. DOI: 10.18653/v1/2022.deelio-1.10. URL:

[207] Liu, Pengfei, Yuan, Weizhe, Fu, Jinlan, Jiang, Zhengbao, Hayashi, Hiroaki, and Neubig, Graham. “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”. In: *ACM Computing Surveys* 55 (2021), pp. 1–35. URL:

[208] Liu, Xiaogeng, Xu, Nan, Chen, Muhao, and Xiao, Chaowei. *AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models*. 2024. arXiv: 2310.04451 [cs.CL]. URL:

[209] Liu, Xiaogeng, Yu, Zhiyuan, Zhang, Yizhe, Zhang, Ning, and Xiao, Chaowei. *Automatic and Universal Prompt Injection Attacks against Large Language Models*. 2024. arXiv: 2403.04957 [cs.AI]. URL:

[210] Liu, Yinhan, Ott, Myle, Goyal, Naman, Du, Jingfei, Joshi, Mandar, Chen, Danqi, Levy, Omer, Lewis, Mike, Zettlemoyer, Luke, and Stoyanov, Veselin. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL]. URL:

[211] Liu, Yinhan, Ott, Myle, Goyal, Naman, Du, Jingfei, Joshi, Mandar, Chen, Danqi, Levy, Omer, Lewis, Mike, Zettlemoyer, Luke, and Stoyanov, Veselin. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *ArXiv* abs/1907.11692 (2019). URL:

[212] Long, Jieyi. “Large Language Model Guided Tree-of-Thought”. In: *ArXiv* abs/2305.08291 (2023). URL:

[213] Lou, Jiaxu and Sun, Yifan. *Anchoring Bias in Large Language Models: An Experimental Study*. 2024. arXiv: 2412.06593 [cs.CL]. URL:

[214] Lovenia, Holy, Dai, Wenliang, Cahyawijaya, Samuel, Ji, Ziwei, and Fung, Pascale. *Negative Object Presence Evaluation (NOPE) to Measure Object Hallucination in Vision-Language Models*. 2023. arXiv: 2310.05338 [cs.CV].

[215] Lu, Yao, Bartolo, Max, Moore, Alastair, Riedel, Sebastian, and Stenetorp, Pontus. “Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8086–8098. DOI: 10.18653/v1/2022.acl-long.556. URL:

[216] Luo, Liangchen, Liu, Yinxiao, Liu, Rosanne, Phatale, Samrat, Lara, Harsh, Li, Yunxuan, Shu, Lei, Zhu, Yun, Meng, Lei, Sun, Jiao, and Rastogi, Abhinav. *Improve Mathematical Reasoning in Language Models by Automated Process Supervision*. 2024. arXiv: 2406.06592 [cs.CL]. URL:

[217] Luo, Man, Kumbhar, Shrinidhi, shen, Ming, Parmar, Mihir, Varshney, Neeraj, Banerjee, Pratyay, Aditya, Somak, and Baral, Chitta. "Towards LogiGLUE: A Brief Survey and A Benchmark for Analyzing Logical Reasoning Capabilities of Language Models". In: *ArXiv* abs/2310.00836 (2023). URL:

[218] Lymperaiou, Maria, Filandrianos, Giorgos, Thomas, Konstantinos, and Stamou, Giorgos. "Counterfactual Edits for Generative Evaluation". In: *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering* (2023).

[219] Lymperaiou, Maria, Filandrianos, Giorgos, Thomas, Konstantinos, and Stamou, Giorgos. *Counterfactual Edits for Generative Evaluation.* 2023. arXiv: 2303.01555 [cs.CV].

[220] Lymperaiou, Maria, Manoliadis, George, Menis Mastromichalakis, Orfeas, Dervakos, Edmund G., and Stamou, Giorgos. "Towards Explainable Evaluation of Language Models on the Semantic Similarity of Visual Concepts". In: *Proceedings of the 29th International Conference on Computational Linguistics.* Ed. by Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 3639–3658. URL:

[221] Lymperaiou, Maria, Manoliadis, George, Menis Mastromichalakis, Orfeas, Dervakos, Edmund G., and Stamou, Giorgos. "Towards Explainable Evaluation of Language Models on the Semantic Similarity of Visual Concepts". In: *Proceedings of the 29th International Conference on Computational Linguistics.* Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 3639–3658. URL:

[222] Lymperopoulos, Dimitris, Lymperaiou, Maria, Filandrianos, Giorgos, and Stamou, Giorgos. *Optimal and efficient text counterfactuals using Graph Neural Networks.* 2024. arXiv: 2408.01969 [cs.CL]. URL:

[223] Lyu, Hanjia, Jiang, Song, Zeng, Hanqing, Xia, Yinglong, Wang, Qifan, Zhang, Si, Chen, Ren, Leung, Christopher, Tang, Jiajie, and Luo, Jiebo. *LLM-Rec: Personalized Recommendation via Prompting Large Language Models.* 2024. arXiv: 2307.15780 [cs.CL]. URL:

[224] Lyu, Yougang, Zhang, Xiaoyu, Ren, Zhaochun, and Rijke, Maarten de. *Cognitive Biases in Large Language Models for News Recommendation.* 2024. arXiv: 2410.02897 [cs.IR]. URL:

[225] Ma, Ke, Zhao, Bo, and Sigal, Leonid. *Attribute-guided image generation from layout.* 2020. arXiv: 2008.11932 [cs.CV].

[226] Maas, Andrew L., Daly, Raymond E., Pham, Peter T., Huang, Dan, Ng, Andrew Y., and Potts, Christopher. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. URL:

[227] Maas, Andrew L., Daly, Raymond E., Pham, Peter T., Huang, Dan, Ng, Andrew Y., and Potts, Christopher. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. URL:

[228] Madaan, Aman, Tandon, Niket, Gupta, Prakhar, Hallinan, Skyler, Gao, Luyu, Wiegreffe, Sarah, Alon, Uri, Dziri, Nouha, Prabhumoye, Shrimai, Yang, Yiming, Welleck, Sean, Majumder, Bodhisattwa Prasad, Gupta, Shashank, Yazdanbakhsh, Amir, and Clark, Peter. "Self-Refine: Iterative Refinement with Self-Feedback". In: *ArXiv* abs/2303.17651 (2023). URL:

[229] Madaan, Nishtha, Padhi, Inkit, Panwar, Naveen, and Saha, Diptikalyan. "Generate your counterfactuals: Towards controlled counterfactual generation for text". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 35. 15. 2021, pp. 13516–13524.

[230] Madsen, Andreas, Reddy, Siva, and Chandar, Sarath. "Post-hoc Interpretability for Neural NLP: A Survey". In: *ACM Comput. Surv.* 55.8 (Dec. 2022). ISSN: 0360-0300. DOI: 10.1145/3546577. URL:

[231] Madsen, Andreas, Reddy, Siva, and Chandar, Sarath. "Post-hoc interpretability for neural nlp: A survey". In: *ACM Computing Surveys* 55.8 (2022), pp. 1–42.

[232] Maharana, Adyasha and Bansal, Mohit. "Integrating Visuospatial, Linguistic, and Commonsense Structure into Story Visualization". In: *ArXiv* abs/2110.10834 (2021).

[233] Maharana, Adyasha, Hannan, Darryl, and Bansal, Mohit. "Improving Generation and Evaluation of Visual Stories via Semantic Consistency". In: *ArXiv* abs/2105.10026 (2021).

[234] Malberg, Simon, Poletukhin, Roman, Schuster, Carolin M., and Groh, Georg. *A Comprehensive Evaluation of Cognitive Biases in LLMs*. 2024. arXiv: 2410.15413 [cs.CL]. URL:

[235] Mastromichalakis, Orfeas Menis, Filandrianos, Giorgos, Symeonaki, Maria, and Stamou, Giorgos. "Assumed Identities: Quantifying Gender Bias in Machine Translation of Ambiguous Occupational Terms". In: *arXiv preprint arXiv:2503.04372* (2025).

[236] Mastromichalakis, Orfeas Menis, Filandrianos, Giorgos, Tsouparopoulou, Eva, Parsanoglou, Dimitris, Symeonaki, Maria, and Stamou, Giorgos. "GOSt-MT: A Knowledge Graph for Occupation-related Gender Biases in Machine Translation". In: *arXiv preprint arXiv:2409.10989* (2024).

[237] Mastromichalakis, Orfeas Menis, Liartis, Jason, and Stamou, Giorgos. *Beyond One-Size-Fits-All: Adapting Counterfactual Explanations to User Objectives*. 2024. arXiv: 2404.08721 [cs.LG].

[238] McKenzie, Ian R., Lyzhov, Alexander, Pieler, Michael, Parrish, Alicia, Mueller, Aaron, Prabhu, Ameya, McLean, Euan, Kirtland, Aaron, Ross, Alexis, Liu, Alisa, Gritsevskiy, Andrew, Wurgaft, Daniel, Kauffman, Derik, Recchia, Gabriel, Liu, Jiacheng, Cavanagh, Joe, Weiss, Max, Huang, Sicong, Droid, The Floating, Tseng, Tom, Korbak, Tomasz, Shen, Xudong, Zhang, Yuhui, Zhou, Zhengping, Kim, Najoung, Bowman, Samuel R., and Perez, Ethan. *Inverse Scaling: When Bigger Isn't Better*. 2024. arXiv: 2306.09479 [cs.CL]. URL:

[239] Menis Mastromichalakis, Orfeas, Filandrianos, Giorgos, Liartis, Jason, Dervakos, Edmund, and Stamou, Giorgos. "Semantic Prototypes: Enhancing Transparency Without Black Boxes". In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2024, pp. 1680–1688.

[240] Michel, Paul, Li, Xian, Neubig, Graham, and Pino, Juan Miguel. *On Evaluation of Adversarial Perturbations for Sequence-to-Sequence Models*. 2019. arXiv: 1903.06620 [cs.CL]. URL:

[241] Mihalcea, Rada and Csomai, Andras. "Wikify! Linking documents to encyclopedic knowledge". In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 2007, pp. 233–242.

[242] Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL]. URL:

[243] Miller, George A. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[244] Miller, George A. "WordNet: A Lexical Database for English". In: *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*. 1992. URL:

[245] Min, Sewon, Lyu, Xinxi, Holtzman, Ari, Artetxe, Mikel, Lewis, Mike, Hajishirzi, Hannaneh, and Zettlemoyer, Luke. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11048–11064. DOI: 10.18653/v1/2022.emnlp-main.759. URL:

[246] Mirza, Mehdi and Osindero, Simon. "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784 (2014). arXiv: 1411.1784. URL:

[247] Mitra, Arindam and Baral, Chitta. "Learning to Automatically Solve Logic Grid Puzzles". In: *Conference on Empirical Methods in Natural Language Processing*. 2015. URL:

[248] Miyato, Takeru, Kataoka, Toshiki, Koyama, Masanori, and Yoshida, Yuichi. *Spectral Normalization for Generative Adversarial Networks*. 2018. arXiv: 1802.05957 [cs.LG].

[249] Miyato, Takeru and Koyama, Masanori. *cGANs with Projection Discriminator*. 2018. arXiv: 1802.05637 [cs.LG].

[250] Mo, Shentong and Xin, Miao. "Tree of Uncertain Thoughts Reasoning for Large Language Models". In: *ArXiv* abs/2309.07694 (2023). URL:

[251] Morris, John X, Lifland, Eli, Yoo, Jin Yong, Grigsby, Jake, Jin, Di, and Qi, Yanjun. "Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp". In: *arXiv preprint arXiv:2005.05909* (2020).

[252] Morris, John X., Lifland, Eli, Yoo, Jin Yong, Grigsby, Jake, Jin, Di, and Qi, Yanjun. *TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP*. 2020. arXiv: 2005.05909 [cs.CL]. URL:

[253] Mothilal, Ramaravind K, Sharma, Amit, and Tan, Chenhao. “Explaining machine learning classifiers through diverse counterfactual explanations”. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 607–617.

[254] Mozes, Maximilian, Kleinberg, Bennett, and Griffin, Lewis D. “Identifying Human Strategies for Generating Word-Level Adversarial Examples”. In: *Conference on Empirical Methods in Natural Language Processing*. 2022. URL:

[255] Nagisetty, Vineel, Graves, Laura, Scott, Joseph, and Ganesh, Vijay. *xAI-GAN: Enhancing Generative Adversarial Networks via Explainable AI Systems*. 2020. DOI: 10.48550/ARXIV.2002.10438. URL:

[256] Nestaas, Fredrik, Debenedetti, Edoardo, and Tramèr, Florian. *Adversarial Search Engine Optimization for Large Language Models*. 2024. arXiv: 2406.18382 [cs.CR]. URL:

[257] Niu, Qian, Liu, Junyu, Bi, Ziqian, Feng, Pohsun, Peng, Benji, Chen, Keyu, Li, Ming, Yan, Lawrence KQ, Zhang, Yichao, Yin, Caitlyn Heqi, Fei, Cheng, Wang, Tianyang, Wang, Yunze, Chen, Silin, and Liu, Ming. *Large Language Models and Cognitive Science: A Comprehensive Review of Similarities, Differences, and Challenges*. 2024. arXiv: 2409.02387 [cs.AI]. URL:

[258] Noever, David A. and Burdick, Ryerson. “Puzzle Solving without Search or Human Knowledge: An Unnatural Language Approach”. In: *ArXiv* abs/2109.02797 (2021). URL:

[259] Odena, Augustus, Olah, Christopher, and Shlens, Jonathon. *Conditional Image Synthesis With Auxiliary Classifier GANs*. 2017. arXiv: 1610.09585 [stat.ML].

[260] Olausson, Theo X., Gu, Alex, Lipkin, Benjamin, Zhang, Cedegao, Solar-Lezama, Armando, Tenenbaum, Josh, and Levy, Roger. “LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers”. In: *Conference on Empirical Methods in Natural Language Processing*. 2023. URL:

[261] Opedal, Andreas, Stolfo, Alessandro, Shirakami, Haruki, Jiao, Ying, Cotterell, Ryan, Schölkopf, Bernhard, Saparov, Abulhair, and Sachan, Mrinmaya. *Do Language Models Exhibit the Same Cognitive Biases in Problem Solving as Human Learners?* 2024. arXiv: 2401.18070 [cs.CL]. URL:

[262] OpenAI, : Achiam, Josh, Adler, Steven, Agarwal, Sandhini, Ahmad, Lama, Akkaya, Ilge, Aleman, Florencia Leoni, Almeida, Diogo, Altenschmidt, Janko, Altman, Sam, Anadkat, Shyamal, Avila, Red, Babuschkin, Igor, Balaji, Suchir, Balcom, Valerie, Baltescu, Paul, Bao, Haiming, Bavarian, Mo, Belgum, Jeff, Bello, Irwan, Berdine, Jake, Bernadett-Shapiro, Gabriel, Berner, Christopher, Bogdonoff, Lenny, Boiko, Oleg, Boyd, Madelaine, Brakman, Anna-Luisa, Brockman, Greg, Brooks, Tim, Brundage, Miles, Button, Kevin, Cai, Trevor, Campbell, Rosie, Cann, Andrew, Carey, Brittany, Carlson, Chelsea, Carmichael, Rory, Chan, Brooke, Chang, Che, Chantzis, Fotis, Chen, Derek, Chen, Sully, Chen, Ruby, Chen, Jason, Chen, Mark, Chess, Ben, Cho, Chester, Chu, Casey, Chung, Hyung Won, Cummings, Dave, Currier, Jeremiah, Dai, Yunxing, Decareaux, Cory, Degry, Thomas, Deutsch, Noah, Deville, Damien, Dhar, Arka, Dohan, David, Dowling, Steve, Dunning, Sheila, Ecoffet, Adrien, Eleti, Atty, Eloundou, Tyna, Farhi, David, Fedus, Liam, Felix, Niko, Fishman, Simón Posada, Forte, Juston, Fulford, Isabella, Gao, Leo, Georges, Elie, Gibson, Christian, Goel, Vik, Gogineni, Tarun, Goh, Gabriel, Gontijo-Lopes, Rapha, Gordon, Jonathan, Grafstein, Morgan, Gray, Scott, Greene, Ryan, Gross, Joshua, Gu, Shixiang Shane, Guo, Yufei, Hallacy, Chris, Han, Jesse, Harris, Jeff, He, Yuchen, Heaton, Mike, Heidecke, Johannes, Hesse, Chris, Hickey, Alan, Hickey, Wade, Hoeschele, Peter, Houghton, Brandon, Hsu, Kenny, Hu, Shengli, Hu, Xin, Huizinga, Joost, Jain, Shantanu, Jain, Shawn, Jang, Joanne, Jiang, Angela, Jiang, Roger, Jin, Haozhun, Jin, Denny, Jomoto, Shino, Jonn, Billie, Jun, Heewoo, Kaftan, Tomer, Kaiser, Łukasz, Kamali, Ali, Kanitscheider, Ingmar, Keskar, Nitish Shirish, Khan, Tabarak, Kilpatrick, Logan, Kim, Jong Wook, Kim, Christina, Kim, Yongjik, Kirchner, Hendrik, Kiros, Jamie, Knight, Matt, Kokotajlo, Daniel, Kondraciuk, Łukasz, Kondrich, Andrew, Konstantinidis, Aris, Kosic, Kyle, Krueger, Gretchen, Kuo, Vishal, Lampe, Michael, Lan, Ikai, Lee, Teddy, Leike, Jan, Leung, Jade, Levy, Daniel, Li, Chak Ming, Lim, Rachel, Lin, Molly, Lin, Stephanie, Litwin, Mateusz, Lopez, Theresa, Lowe, Ryan, Lue, Patricia, Makanju, Anna, Malfacini, Kim, Manning, Sam, Markov, Todor, Markovski, Yaniv, Martin, Bianca, Mayer, Katie, Mayne, Andrew, McGrew, Bob, McKinney, Scott Mayer, McLeavey, Christine, McMillan, Paul, McNeil, Jake, Medina, David, Mehta, Aalok, Menick, Jacob, Metz, Luke, Mishchenko, Andrey, Mishkin, Pamela, Monaco, Vinnie, Morikawa, Evan, Mossing, Daniel, Mu, Tong, Murati, Mira, Murk, Oleg, Mély, David, Nair, Ashvin, Nakano, Reiichiro, Nayak, Rajeev, Neelakantan, Arvind, Ngo, Richard, Noh, Hyeonwoo, Ouyang, Long, O’Keefe, Cullen, Pachocki, Jakub, Paino, Alex, Palermo, Joe, Pantuliano, Ashley, Parascandolo, Giambattista, Parish, Joel, Parparita, Emy, Passos, Alex, Pavlov, Mikhail, Peng, An-

drew, Perelman, Adam, Avila Belbute Peres, Filipe de, Petrov, Michael, Oliveira Pinto, Henrique Ponde de, Michael, Pokorny, Pokrass, Michelle, Pong, Vitchyr, Powell, Tolly, Power, Alethea, Power, Boris, Proehl, Elizabeth, Puri, Raul, Radford, Alec, Rae, Jack, Ramesh, Aditya, Raymond, Cameron, Real, Francis, Rimbach, Kendra, Ross, Carl, Rotsted, Bob, Roussez, Henri, Ryder, Nick, Saltarelli, Mario, Sanders, Ted, Santurkar, Shibani, Sastry, Girish, Schmidt, Heather, Schnurr, David, Schulman, John, Selsam, Daniel, Sheppard, Kyla, Sherbakov, Toki, Shieh, Jessica, Shoker, Sarah, Shyam, Pranav, Sidor, Szymon, Sigler, Eric, Simens, Maddie, Sitkin, Jordan, Slama, Katarina, Sohl, Ian, Sokolowsky, Benjamin, Song, Yang, Staudacher, Natalie, Such, Felipe Petroski, Summers, Natalie, Sutskever, Ilya, Tang, Jie, Tezak, Nikolas, Thompson, Madeleine, Tillet, Phil, Tootoonchian, Amin, Tseng, Elizabeth, Tuggle, Preston, Turley, Nick, Tworek, Jerry, Uribe, Juan Felipe Cerón, Vallone, Andrea, Vijayvergiya, Arun, Voss, Chelsea, Wainwright, Carroll, Wang, Justin Jay, Wang, Alvin, Wang, Ben, Ward, Jonathan, Wei, Jason, Weinmann, CJ, Welihinda, Akila, Welinder, Peter, Weng, Jiayi, Weng, Lilian, Wiethoff, Matt, Willner, Dave, Winter, Clemens, Wolrich, Samuel, Wong, Hannah, Workman, Lauren, Wu, Sherwin, Wu, Jeff, Wu, Michael, Xiao, Kai, Xu, Tao, Yoo, Sarah, Yu, Kevin, Yuan, Qiming, Zaremba, Wojciech, Zellers, Rowan, Zhang, Chong, Zhang, Marvin, Zhao, Shengjia, Zheng, Tianhao, Zhuang, Juntang, Zhuk, William, and Zoph, Barret. *GPT-4 Technical Report.* 2023. arXiv: 2303.08774 [cs.CL].

[263] OpenAI, Achiam, Josh, Adler, Steven, Agarwal, Sandhini, Ahmad, Lama, Akkaya, Ilge, Aleman, Florencia Leoni, Almeida, Diogo, Altenschmidt, Janko, Altman, Sam, Anadkat, Shyamal, Avila, Red, Babuschkin, Igor, Balaji, Suchir, Balcom, Valerie, Baltescu, Paul, Bao, Haiming, Bavarian, Mohammad, Belgum, Jeff, Bello, Irwan, Berdine, Jake, Bernadett-Shapiro, Gabriel, Berner, Christopher, Bogdonoff, Lenny, Boiko, Oleg, Boyd, Madelaine, Brakman, Anna-Luisa, Brockman, Greg, Brooks, Tim, Brundage, Miles, Button, Kevin, Cai, Trevor, Campbell, Rosie, Cann, Andrew, Carey, Brittany, Carlson, Chelsea, Carmichael, Rory, Chan, Brooke, Chang, Che, Chantzis, Fotis, Chen, Derek, Chen, Sully, Chen, Ruby, Chen, Jason, Chen, Mark, Chess, Ben, Cho, Chester, Chu, Casey, Chung, Hyung Won, Cummings, Dave, Currier, Jeremiah, Dai, Yunxing, Decareaux, Cory, Degry, Thomas, Deutsch, Noah, Deville, Damien, Dhar, Arka, Dohan, David, Dowling, Steve, Dunning, Sheila, Ecoffet, Adrien, Eleti, Atty, Eloundou, Tyna, Farhi, David, Fedus, Liam, Felix, Niko, Fishman, Simón Posada, Forte, Juston, Fulford, Isabella, Gao, Leo, Georges, Elie, Gibson, Christian, Goel, Vik, Gogineni, Tarun, Goh, Gabriel, Gontijo-Lopes, Rapha, Gordon, Jonathan, Grafstein, Morgan, Gray, Scott, Greene, Ryan, Gross, Joshua, Gu, Shixiang Shane, Guo, Yufei, Hallacy, Chris, Han, Jesse, Harris, Jeff, He, Yuchen, Heaton, Mike, Heidecke, Johannes, Hesse, Chris, Hickey, Alan, Hickey, Wade, Hoeschele, Peter, Houghton, Brandon, Hsu, Kenny, Hu, Shengli, Hu, Xin, Huizinga, Joost, Jain, Shantanu, Jain, Shawn, Jang, Joanne, Jiang, Angela, Jiang, Roger, Jin, Haozhun, Jin, Denny, Jomoto, Shino, Jonn, Billie, Jun, Heewoo, Kaftan, Tomer, Kaiser, Łukasz, Kamali, Ali, Kanitscheider, Ingmar, Keskar, Nitish Shirish, Khan, Tabarak, Kilpatrick, Logan, Kim, Jong Wook, Kim, Christina, Kim, Yongjik, Kirchner, Jan Hendrik, Kiros, Jamie, Knight, Matt, Kokotajlo, Daniel, Kondraciuk, Łukasz, Kondrich, Andrew, Konstantinidis, Aris, Kosic, Kyle, Krueger, Gretchen, Kuo, Vishal, Lampe, Michael, Lan, Ikai, Lee, Teddy, Leike, Jan, Leung, Jade, Levy, Daniel, Li, Chak Ming, Lim, Rachel, Lin, Molly, Lin, Stephanie, Litwin, Mateusz, Lopez, Theresa, Lowe, Ryan, Lue, Patricia, Makanju, Anna, Malfacini, Kim, Manning, Sam, Markov, Todor, Markovski, Yaniv, Martin, Bianca, Mayer, Katie, Mayne, Andrew, McGrew, Bob, McKinney, Scott Mayer, McLeavey, Christine, McMillan, Paul, McNeil, Jake, Medina, David, Mehta, Aalok, Menick, Jacob, Metz, Luke, Mishchenko, Andrey, Mishkin, Pamela, Monaco, Vinnie, Morikawa, Evan, Mossing, Daniel, Mu, Tong, Murati, Mira, Murk, Oleg, Mély, David, Nair, Ashvin, Nakano, Reiichiro, Nayak, Rajeev, Neelakantan, Arvind, Ngo, Richard, Noh, Hyeonwoo, Ouyang, Long, O'Keefe, Cullen, Pachocki, Jakub, Paino, Alex, Palermo, Joe, Pantuliano, Ashley, Parascandolo, Giambattista, Parish, Joel, Parparita, Emy, Passos, Alex, Pavlov, Mikhail, Peng, Andrew, Perelman, Adam, Avila Belbute Peres, Filipe de, Petrov, Michael, Oliveira Pinto, Henrique Ponde de, Michael, Pokorny, Pokrass, Michelle, Pong, Vitchyr H., Powell, Tolly, Power, Alethea, Power, Boris, Proehl, Elizabeth, Puri, Raul, Radford, Alec, Rae, Jack, Ramesh, Aditya, Raymond, Cameron, Real, Francis, Rimbach, Kendra, Ross, Carl, Rotsted, Bob, Roussez, Henri, Ryder, Nick, Saltarelli, Mario, Sanders, Ted, Santurkar, Shibani, Sastry, Girish, Schmidt, Heather, Schnurr, David, Schulman, John, Selsam, Daniel, Sheppard, Kyla, Sherbakov, Toki, Shieh, Jessica, Shoker, Sarah, Shyam, Pranav, Sidor, Szymon, Sigler, Eric, Simens, Maddie, Sitkin, Jordan, Slama, Katarina, Sohl, Ian, Sokolowsky, Benjamin, Song, Yang, Staudacher, Natalie, Such, Felipe Petroski, Summers, Natalie,

Sutskever, Ilya, Tang, Jie, Tezak, Nikolas, Thompson, Madeleine B., Tillet, Phil, Tootoonchian, Amin, Tseng, Elizabeth, Tuggle, Preston, Turley, Nick, Tworek, Jerry, Uribe, Juan Felipe Cerón, Vallone, Andrea, Vijayvergiya, Arun, Voss, Chelsea, Wainwright, Carroll, Wang, Justin Jay, Wang, Alvin, Wang, Ben, Ward, Jonathan, Wei, Jason, Weinmann, CJ, Welihinda, Akila, Welinder, Peter, Weng, Jiayi, Weng, Lilian, Wiethoff, Matt, Willner, Dave, Winter, Clemens, Wolrich, Samuel, Wong, Hannah, Workman, Lauren, Wu, Sherwin, Wu, Jeff, Wu, Michael, Xiao, Kai, Xu, Tao, Yoo, Sarah, Yu, Kevin, Yuan, Qiming, Zaremba, Wojciech, Zellers, Rowan, Zhang, Chong, Zhang, Marvin, Zhao, Shengjia, Zheng, Tianhao, Zhuang, Juntang, Zhuk, William, and Zoph, Barret. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774.

[264] Opitz, Juri and Frank, Anette. "Towards a Decomposable Metric for Explainable Evaluation of Text Generation from AMR". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 1504–1518. DOI: 10.18653/v1/2021.eacl-main.129. URL:

[265] Orlandic, Lara, Teijeiro, Tomas, and Atienza, David. "The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms". In: *Scientific Data* 8.1 (2021), p. 156.

[266] Pan, Liangming, Albalak, Alon, Wang, Xinyi, and Wang, William Yang. "Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning". In: *ArXiv* abs/2305.12295 (2023). URL:

[267] Pan, Xichen, Qin, Pengda, Li, Yuhong, Xue, Hui, and Chen, Wenhu. "Synthesizing coherent story with auto-regressive latent diffusion models". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 2920–2930.

[268] Panagiotopoulos, Ioannis, Filandrianos, George, Lymperaiou, Maria, and Stamou, Giorgos. "AILS-NTUA at SemEval-2024 Task 9: Cracking Brain Teasers: Transformer Models for Lateral Thinking Puzzles". In: *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Ed. by Atul Kr. Ojha, A. Seza Doğruöz, Harish Tayyar Madabushi, Giovanni Da San Martino, Sara Rosenthal, and Aiala Rosá. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 1733–1746. URL:

[269] Panagiotopoulos, Ioannis, Filandrianos, Giorgos, Lymperaiou, Maria, and Stamou, Giorgos. "RISCORE: Enhancing In-Context Riddle Solving in Language Models through Context-Reconstructed Example Augmentation". In: *arXiv preprint arXiv:2409.16383* (2024).

[270] Papadimitriou, Christos, Filandrianos, Giorgos, Lymperaiou, Maria, and Stamou, Giorgos. *Masked Generative Story Transformer with Character Guidance and Caption Augmentation*. 2024. arXiv: 2403.08502 [cs.CV]. URL:

[271] Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL:

[272] Park, Taesung, Liu, Ming-Yu, Wang, Ting-Chun, and Zhu, Jun-Yan. *Semantic Image Synthesis with Spatially-Adaptive Normalization*. 2019. arXiv: 1903.07291 [cs.CV].

[273] Parmar, Gaurav, Zhang, Richard, and Zhu, Jun-Yan. "On Aliased Resizing and Surprising Subtleties in GAN Evaluation". In: *CVPR*. 2022.

[274] Parsanoglou, Dimitris, Mifsud, Louise, Ayllón, Sara, Brugarolas, Pablo, Hyggen, Christer, and Hornung, Helena. "Combining innovative methodological tools to approach digital transformations in leisure among children and young people". In: (2022).

[275] Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[276] Pourcel, Julien, Colas, Cédric, Oudeyer, Pierre-Yves, and Teodorescu, Laetitia. "ACES: Generating Diverse Programming Puzzles with Autotelic Language Models and Semantic Descriptors". In: *ArXiv* abs/2310.10692 (2023). URL:

[277] Poyiadzi, Rafael, Sokol, Kacper, Santos-Rodriguez, Raul, De Bie, Tijl, and Flach, Peter. "FACE: Feasible and Actionable Counterfactual Explanations". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. AIES '20. ACM, Feb. 2020. DOI: 10.1145/3375627.3375850. URL:

[278] Premptis, Iraklis, Lymperaiou, Maria, Filandrianos, Giorgos, Mastromichalakis, Orfeas Menis, Voulodimos, Athanasios, and Stamou, Giorgos. "AILS-NTUA at SemEval-2025 Task 4: Parameter-Efficient Unlearning for Large Language Models using Data Chunking". In: *arXiv preprint arXiv:2503.02443* (2025).

[279] *Protogen x3.4*. Huggingface. URL:

[280] *Protogen x5.8*. Huggingface. URL:

[281] Qiao, Shuofei, Ou, Yixin, Zhang, Ningyu, Chen, Xiang, Yao, Yunzhi, Deng, Shumin, Tan, Chuanqi, Huang, Fei, and Chen, Huajun. "Reasoning with Language Model Prompting: A Survey". In: *ArXiv* abs/2212.09597 (2022). URL:

[282] Qiao, Shuofei, Ou, Yixin, Zhang, Ningyu, Chen, Xiang, Yao, Yunzhi, Deng, Shumin, Tan, Chuanqi, Huang, Fei, and Chen, Huajun. "Reasoning with Language Model Prompting: A Survey". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 5368–5393. DOI: 10.18653/v1/2023.acl-long.294. URL:

[283] Qin, Chengwei, Zhang, Aston, Chen, Chen, Dagar, Anirudh, and Ye, Wenming. *In-Context Learning with Iterative Demonstration Selection*. 2024. arXiv: 2310.09881 [cs.CL]. URL:

[284] Radford, Alec, Kim, Jong, Hallacy, Chris, Ramesh, Aditya, Goh, Gabriel, Agarwal, Sandhini, Sastry, Girish, Askell, Amanda, Mishkin, Pamela, Clark, Jack, Krueger, Gretchen, and Sutskever, Ilya. *Learning Transferable Visual Models From Natural Language Supervision*. Feb. 2021.

[285] Radford, Alec, Wu, Jeff, Child, Rewon, Luan, David, Amodei, Dario, and Sutskever, Ilya. "Language Models are Unsupervised Multitask Learners". In: 2019. URL:

[286] Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei, and Liu, Peter J. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL:

[287] Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei, and Liu, Peter J. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL:

[288] Raffel, Colin, Shazeer, Noam M., Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei, and Liu, Peter J. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *J. Mach. Learn. Res.* 21 (2019), 140:1–140:67. URL:

[289] Rahman, Tanzila, Lee, Hsin-Ying, Ren, Jian, Tulyakov, Sergey, Mahajan, Shweta, and Sigal, Leonid. "Make-a-story: Visual memory conditioned consistent story generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2493–2502.

[290] Ramesh, Aditya, Dhariwal, Prafulla, Nichol, Alex, Chu, Casey, and Chen, Mark. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. 2022. DOI: 10.48550/ARXIV.2204.06125. URL:

[291] Ramesh, Aditya, Pavlov, Mikhail, Goh, Gabriel, Gray, Scott, Voss, Chelsea, Radford, Alec, Chen, Mark, and Sutskever, Ilya. "Zero-shot text-to-image generation". In: *International conference on machine learning*. Pmlr. 2021, pp. 8821–8831.

[292] Ravi, Nikhila, Gabeur, Valentin, Hu, Yuan-Ting, Hu, Ronghang, Ryali, Chaitanya, Ma, Tengyu, Khedr, Haitham, Rädle, Roman, Rolland, Chloe, Gustafson, Laura, et al. "Sam 2: Segment anything in images and videos". In: *arXiv preprint arXiv:2408.00714* (2024).

[293] Reimers, Nils and Gurevych, Iryna. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL:

[294] Ren, Shaoqing, He, Kaiming, Girshick, Ross, and Sun, Jian. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL:

[295] Ren, Shuhuai, Deng, Yihe, He, Kun, and Che, Wanxiang. "Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís

Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1085–1097. DOI: 10.18653/v1/P19-1103. URL:

[296] Robeer, Marcel, Bex, Floris, and Feelders, Ad. "Generating Realistic Natural Language Counterfactuals". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3611–3625. DOI: 10.18653/v1/2021.findings-emnlp.306. URL:

[297] Robeer, Marcel, Bex, Floris, and Feelders, Ad. "Generating realistic natural language counterfactuals". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2021, pp. 3611–3625.

[298] Rohrbach, Anna, Hendricks, Lisa Anne, Burns, Kaylee, Darrell, Trevor, and Saenko, Kate. *Object Hallucination in Image Captioning*. 2019. arXiv: 1809.02156 [cs.CL].

[299] Rombach, Robin, Blattmann, Andreas, Lorenz, Dominik, Esser, Patrick, and Ommer, Björn. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. arXiv: 2112.10752 [cs.CV].

[300] Rombach, Robin, Blattmann, Andreas, Lorenz, Dominik, Esser, Patrick, and Ommer, Björn. "High-Resolution Image Synthesis With Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10684–10695.

[301] Rombach, Robin, Blattmann, Andreas, Lorenz, Dominik, Esser, Patrick, and Ommer, Björn. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.

[302] Ross, Alexis, Marasović, Ana, and Peters, Matthew. "Explaining NLP Models via Minimal Contrastive Editing (MiCE)". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, Aug. 2021, pp. 3840–3852. DOI: 10.18653/v1/2021.findings-acl.336. URL:

[303] Ross, Alexis, Marasović, Ana, and Peters, Matthew. "Explaining NLP Models via Minimal Contrastive Editing (MiCE)". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, Aug. 2021, pp. 3840–3852. DOI: 10.18653/v1/2021.findings-acl.336. URL:

[304] Ross, Alexis, Wu, Tongshuang, Peng, Hao, Peters, Matthew, and Gardner, Matt. "Tailor: Generating and Perturbing Text with Semantic Controls". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3194–3213. DOI: 10.18653/v1/2022.acl-long.228. URL:

[305] Ross, Alexis, Wu, Tongshuang, Peng, Hao, Peters, Matthew, and Gardner, Matt. "Tailor: Generating and Perturbing Text with Semantic Controls". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3194–3213. DOI: 10.18653/v1/2022.acl-long.228. URL:

[306] Rozner, Josh, Potts, Christopher, and Mahowald, Kyle. "Decrypting Cryptic Crosswords: Semantically Complex Wordplay Puzzles as a Target for NLP". In: *ArXiv* abs/2104.08620 (2021). URL:

[307] Rudin, Cynthia. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature machine intelligence* 1.5 (2019), pp. 206–215.

[308] Ruiz, Nataniel, Li, Yuanzhen, Jampani, Varun, Pritch, Yael, Rubinstein, Michael, and Aberman, Kfir. *DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation*. 2022. DOI: 10.48550/ARXIV.2208.12242. URL:

[309] Sachdeva, Rachneet, Tutek, Martin, and Gurevych, Iryna. "CATfOOD: Counterfactual Augmented Training for Improving Out-of-Domain Performance and Calibration". In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Yvette Graham and Matthew Purver. St. Julian's, Malta: Association for Computational Linguistics, Mar. 2024, pp. 1876–1898. URL:

[310] Saharia, Chitwan, Chan, William, Saxena, Saurabh, Li, Lala, Whang, Jay, Denton, Emily, Ghasemipour, Seyed Kamyar Seyed, Ayan, Burcu Karagol, Mahdavi, S. Sara, Lopes, Rapha Gontijo, Salimans, Tim, Ho, Jonathan, Fleet, David J, and Norouzi, Mohammad. *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. 2022. arXiv: 2205.11487 [cs.CV].

[311] Saharia, Chitwan, Chan, William, Saxena, Saurabh, Li, Lala, Whang, Jay, Denton, Emily, Ghasemipour, Seyed Kamyar Seyed, Ayan, Burcu Karagol, Mahdavi, S. Sara, Lopes, Rapha Gontijo,

Salimans, Tim, Ho, Jonathan, Fleet, David J, and Norouzi, Mohammad. *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. 2022. DOI: 10.48550/ARXIV.2205.11487. URL:

[312] Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, Chen, Xi, and Chen, Xi. "Improved Techniques for Training GANs". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc., 2016. URL:

[313] Sanyal, Soumya, Singh, Harman, and Ren, Xiang. "FaiRR: Faithful and Robust Deductive Reasoning over Natural Language". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1075–1093. DOI: 10.18653/v1/2022.acl-long.77. URL:

[314] Sap, Maarten, Le Bras, Ronan, Allaway, Emily, Bhagavatula, Chandra, Lourie, Nicholas, Rashkin, Hannah, Roof, Brendan, Smith, Noah A., and Choi, Yejin. "ATOMIC: an atlas of machine commonsense for if-then reasoning". In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'19/IAAI'19/EAAI'19. Honolulu, Hawaii, USA: AAAI Press, 2019. ISBN: 978-1-57735-809-1. DOI: 10.1609/aaai.v33i01.33013027. URL:

[315] Sap, Maarten, Shwartz, Vered, Bosselut, Antoine, Choi, Yejin, and Roth, Dan. "Commonsense Reasoning for Natural Language Processing". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Ed. by Agata Savary and Yue Zhang. Online: Association for Computational Linguistics, July 2020, pp. 27–33. DOI: 10.18653/v1/2020.acl-tutorials.7. URL:

[316] Savelka, Jaromir, Agarwal, Arav, Bogart, Christopher, and Sakr, Majd. *Large Language Models (GPT) Struggle to Answer Multiple-Choice Questions about Code*. 2023. arXiv: 2303.08033 [cs.CL].

[317] Scarselli, Franco, Gori, Marco, Tsoi, Ah Chung, Hagenbuchner, Markus, and Monfardini, Gabriele. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[318] Schuster, Tal, Kalyan, A., Polozov, Oleksandr, and Kalai, Adam Tauman. "Programming Puzzles". In: *ArXiv* abs/2106.05784 (2021). URL:

[319] Shaki, Jonathan, Kraus, Sarit, and Wooldridge, Michael. "Cognitive Effects in Large Language Models". In: *ECAI 2023*. IOS Press, Sept. 2023. ISBN: 9781643684376. DOI: 10.3233/faia230505. URL:

[320] Sharma, Neeraj, Krishnan, Prashant, Kumar, Rohit, Ramoji, Shreyas, Chetupalli, Srikanth Raj, Ghosh, Prasanta Kumar, Ganapathy, Sriram, et al. "Coswara–a database of breathing, cough, and voice sounds for COVID-19 diagnosis". In: *arXiv preprint arXiv:2005.10548* (2020).

[321] Shayegani, Erfan, Mamun, Md Abdullah Al, Fu, Yu, Zaree, Pedram, Dong, Yue, and Abu-Ghazaleh, Nael. *Survey of Vulnerabilities in Large Language Models Revealed by Adversarial Attacks*. 2023. arXiv: 2310.10844 [cs.CL]. URL:

[322] Shen, Yujun, Gu, Jinjin, Tang, Xiaoou, and Zhou, Bolei. "Interpreting the Latent Space of GANs for Semantic Face Editing". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 9240–9249.

[323] Shen, Yujun, Yang, Ceyuan, Tang, Xiaoou, and Zhou, Bolei. "InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44 (2020), pp. 2004–2018.

[324] Simonyan, Karen, Vedaldi, Andrea, and Zisserman, Andrew. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[325] Soloveitchik, Michael, Diskin, Tzvi, Morin, Efrat, and Wiesel, Ami. *Conditional Frechet Inception Distance*. 2021. DOI: 10.48550/ARXIV.2103.11521. URL:

[326] Song, Tianyi, Cao, Jiuxin, Wang, Kun, Liu, Bo, and Zhang, Xiaofeng. *Causal-Story: Local Causal Attention Utilizing Parameter-Efficient Tuning For Visual Story Synthesis*. 2023. arXiv: 2309.09553 [cs.CV].

[327] Speer, Robyn, Chin, Joshua, and Havasi, Catherine. "ConceptNet 5.5: an open multilingual graph of general knowledge". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, pp. 4444–4451.

[328] Speer, Robyn, Chin, Joshua, and Havasi, Catherine. “ConceptNet 5.5: an open multilingual graph of general knowledge”. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI’17. San Francisco, California, USA: AAAI Press, 2017, pp. 4444–4451.

[329] *Stable Diffusion 2 base*. Huggingface. URL:

[330] *Stable Diffusion v1.4*. Huggingface. URL:

[331] Stammer, Wolfgang, Schramowski, Patrick, and Kersting, Kristian. “Right for the Right Concept: Revising Neuro-Symbolic Concepts by Interacting With Their Explanations”. In: *CVPR*. Computer Vision Foundation / IEEE, 2021, pp. 3619–3629.

[332] Stringli, Elena, Lymperaiou, Maria, Filandrianos, Giorgos, and Stamou, Giorgos. “Pitfalls of Scale: Investigating the Inverse Task of Redefinition in Large Language Models”. In: *arXiv preprint arXiv:2502.12821* (2025).

[333] Sultan, Oren and Shahaf, Dafna. “Life is a Circus and We are the Clowns: Automatically Finding Analogies between Situations and Processes”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 3547–3562. DOI: 10.18653/v1/2022.emnlp-main.232. URL:

[334] Sumita, Yasuaki, Takeuchi, Koh, and Kashima, Hisashi. *Cognitive Biases in Large Language Models: A Survey and Mitigation Experiments*. 2024. arXiv: 2412.00323 [cs.CL]. URL:

[335] Sun, Wei and Wu, Tianfu. *Learning Layout and Style Reconfigurable GANs for Controllable Image Synthesis*. 2021. arXiv: 2003.11571 [cs.CV].

[336] Symeonaki, Maria, Filandrianos, George, and Stamou, Giorgos. “Visualising key information and communication technologies (ICT) indicators for children and young individuals in Europe”. In: *Humanities and Social Sciences Communications* 9.1 (2022), pp. 1–12.

[337] Szomiu, Roxana and Groza, Adrian. “A Puzzle-Based Dataset for Natural Language Inference”. In: *ArXiv* abs/2112.05742 (2021). URL:

[338] Talmor, Alon, Herzig, Jonathan, Lourie, Nicholas, and Berant, Jonathan. “CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge”. In: *ArXiv* abs/1811.00937 (2019). URL:

[339] Tang, Jiabin, Yang, Yuhao, Wei, Wei, Shi, Lei, Su, Lixin, Cheng, Suqi, Yin, Dawei, and Huang, Chao. “Graphgpt: Graph instruction tuning for large language models”. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2024, pp. 491–500.

[340] Team, Gemini, Anil, Rohan, Borgeaud, Sebastian, Alayrac, Jean-Baptiste, Yu, Jiahui, Soricut, Radu, Schalkwyk, Johan, Dai, Andrew M., Hauth, Anja, Millican, Katie, Silver, David, Johnson, Melvin, Antonoglou, Ioannis, Schrittwieser, Julian, Glaese, Amelia, Chen, Jilin, Pitler, Emily, Lillicrap, Timothy, Lazaridou, Angeliki, Firat, Orhan, Molloy, James, Isard, Michael, Barham, Paul R., Hennigan, Tom, Lee, Benjamin, Viola, Fabio, Reynolds, Malcolm, Xu, Yuanzhong, Doherty, Ryan, Collins, Eli, Meyer, Clemens, Rutherford, Eliza, Moreira, Erica, Ayoub, Kareem, Goel, Megha, Krawczyk, Jack, Du, Cosmo, Chi, Ed, Cheng, Heng-Tze, Ni, Eric, Shah, Purvi, Kane, Patrick, Chan, Betty, Faruqui, Manaal, Severyn, Aliaksei, Lin, Hanzhao, Li, YaGuang, Cheng, Yong, Ittycheriah, Abe, Mahdieh, Mahdis, Chen, Mia, Sun, Pei, Tran, Dustin, Bagri, Sumit, Lakshminarayanan, Balaji, Liu, Jeremiah, Orban, Andras, Güra, Fabian, Zhou, Hao, Song, Xinying, Boffy, Aurelien, Ganapathy, Harish, Zheng, Steven, Choe, HyunJeong, Weisz, Ágoston, Zhu, Tao, Lu, Yifeng, Gopal, Siddharth, Kahn, Jarrod, Kula, Maciej, Pitman, Jeff, Shah, Rushin, Taropa, Emanuel, Merey, Majd Al, Baeuml, Martin, Chen, Zhifeng, Shafey, Laurent El, Zhang, Yujing, Sercinoglu, Olcan, Tucker, George, Piqueras, Enrique, Krikun, Maxim, Barr, Iain, Savinov, Nikolay, Danihelka, Ivo, Roelofs, Becca, White, Anaïs, Andreassen, Anders, Glehn, Tamara von, Yagati, Lakshman, Kazemi, Mehran, Gonzalez, Lucas, Khalman, Misha, Sygnowski, Jakub, Frechette, Alexandre, Smith, Charlotte, Culp, Laura, Proleev, Lev, Luan, Yi, Chen, Xi, Lottes, James, Schucher, Nathan, Lebron, Federico, Rrustemi, Alban, Clay, Natalie, Crone, Phil, Kocisky, Tomas, Zhao, Jeffrey, Perz, Bartek, Yu, Dian, Howard, Heidi, Bloniarz, Adam, Rae, Jack W., Lu, Han, Sifre, Laurent, Maggioni, Marcello, Alcober, Fred, Garrette, Dan, Barnes, Megan, Thakoor, Shantanu, Austin, Jacob, Barth-Maron, Gabriel, Wong, William, Joshi, Rishabh, Chaabouni, Rahma, Fatiha, Deeni, Ahuja, Arun, Tomar, Gaurav Singh, Senter, Evan, Chadwick, Martin, Kornakov, Ilya, Attaluri, Nithya, Iturrate, Iñaki, Liu, Ruibo, Li, Yunxuan, Cogan, Sarah, Chen, Jeremy, Jia, Chao, Gu, Chenjie, Zhang, Qiao, Grimstad, Jordan, Hartman, Ale Jakse, Garcia,

Xavier, Pillai, Thanumalayan Sankaranarayana, Devlin, Jacob, Laskin, Michael, Las Casas, Diego de, Valter, Dasha, Tao, Connie, Blanco, Lorenzo, Badia, Adrià Puigdomènech, Reitter, David, Chen, Mianna, Brennan, Jenny, Rivera, Clara, Brin, Sergey, Iqbal, Shariq, Surita, Gabriela, Labanowski, Jane, Rao, Abhi, Winkler, Stephanie, Parisotto, Emilio, Gu, Yiming, Olszewska, Kate, Addanki, Ravi, Miech, Antoine, Louis, Annie, Teplyashin, Denis, Brown, Geoff, Catt, Elliot, Balaguer, Jan, Xiang, Jackie, Wang, Pidong, Ashwood, Zoe, Briukhov, Anton, Webson, Albert, Ganapathy, Sanjay, Sanghavi, Smit, Kannan, Ajay, Chang, Ming-Wei, Stjerngren, Axel, Djolonga, Josip, Sun, Yuting, Bapna, Ankur, Aitchison, Matthew, Pejman, Pedram, Michalewski, Henryk, Yu, Tianhe, Wang, Cindy, Love, Juliette, Ahn, Junwhan, Bloxwich, Dawn, Han, Kehang, Humphreys, Peter, Sellam, Thibault, Bradbury, James, Godbole, Varun, Samangooei, Sina, Damoc, Bogdan, Kaskasoli, Alex, Arnold, Sébastien M. R., Vasudevan, Vijay, Agrawal, Shubham, Riesa, Jason, Lepikhin, Dmitry, Tanburn, Richard, Srinivasan, Srivatsan, Lim, Hyeontaek, Hodkinson, Sarah, Shyam, Pranav, Ferret, Johan, Hand, Steven, Garg, Ankush, Paine, Tom Le, Li, Jian, Li, Yujia, Giang, Minh, Neitz, Alexander, Abbas, Zaheer, York, Sarah, Reid, Machel, Cole, Elizabeth, Chowdhery, Aakanksha, Das, Dipanjan, Rogozińska, Dominika, Nikolaev, Vitaliy, Sprechmann, Pablo, Nado, Zachary, Zilka, Lukas, Prost, Flavien, He, Luheng, Monteiro, Marianne, Mishra, Gaurav, Welty, Chris, Newlan, Josh, Jia, Dawei, Allamanis, Miltiadis, Hu, Clara Huiyi, Liedekerke, Raoul de, Gilmer, Justin, Saroufim, Carl, Rijhwani, Shruti, Hou, Shaobo, Shrivastava, Disha, Baddepudi, Anirudh, Goldin, Alex, Ozturel, Adnan, Cassirer, Albin, Xu, Yunhan, Sohn, Daniel, Sachan, Devendra, Amplayo, Reinald Kim, Swanson, Craig, Petrova, Dessie, Narayan, Shashi, Guez, Arthur, Brahma, Siddhartha, Landon, Jessica, Patel, Miteyan, Zhao, Ruizhe, Villela, Kevin, Wang, Luyu, Jia, Wenhao, Rahtz, Matthew, Giménez, Mai, Yeung, Legg, Keeling, James, Georgiev, Petko, Mincu, Diana, Wu, Boxi, Haykal, Salem, Saputro, Rachel, Vodrahalli, Kiran, Qin, James, Cankara, Zeynep, Sharma, Abhanshu, Fernando, Nick, Hawkins, Will, Neyshabur, Behnam, Kim, Solomon, Hutter, Adrian, Agrawal, Priyanka, Castro-Ros, Alex, Driessche, George van den, Wang, Tao, Yang, Fan, Chang, Shuo-yiin, Komarek, Paul, McIlroy, Ross, Lučić, Mario, Zhang, Guodong, Farhan, Wael, Sharman, Michael, Natsev, Paul, Michel, Paul, Bansal, Yamini, Qiao, Siyuan, Cao, Kris, Shakeri, Siamak, Butterfield, Christina, Chung, Justin, Rubenstein, Paul Kishan, Agrawal, Shivani, Mensch, Arthur, Soparkar, Kedar, Lenc, Karel, Chung, Timothy, Pope, Aedan, Maggiore, Loren, Kay, Jackie, Jhakra, Priya, Wang, Shibo, Maynez, Joshua, Phuong, Mary, Tobin, Taylor, Tacchetti, Andrea, Trebacz, Maja, Robinson, Kevin, Katariya, Yash, Riedel, Sebastian, Bailey, Paige, Xiao, Kefan, Ghelani, Nimesh, Aroyo, Lora, Slone, Ambrose, Houlsby, Neil, Xiong, Xuehan, Yang, Zhen, Gribovskaya, Elena, Adler, Jonas, Wirth, Mateo, Lee, Lisa, Li, Music, Kagohara, Thais, Pavagadhi, Jay, Bridgers, Sophie, Bortsova, Anna, Ghemawat, Sanjay, Ahmed, Zafarali, Liu, Tianqi, Powell, Richard, Bolina, Vijay, Iinuma, Mariko, Zablotskaia, Polina, Besley, James, Chung, Da-Woon, Dozat, Timothy, Comanescu, Ramona, Si, Xiance, Greer, Jeremy, Su, Guolong, Polacek, Martin, Kaufman, Raphaël Lopez, Tokumine, Simon, Hu, Hexiang, Buchatskaya, Elena, Miao, Yingjie, Elhawaty, Mohamed, Siddhant, Aditya, Tomasev, Nenad, Xing, Jinwei, Greer, Christina, Miller, Helen, Ashraf, Shereen, Roy, Aurko, Zhang, Zizhao, Ma, Ada, Filos, Angelos, Besta, Milos, Blevins, Rory, Klimenko, Ted, Yeh, Chih-Kuan, Changpinyo, Soravit, Mu, Jiaqi, Chang, Oscar, Pajarskas, Mantas, Muir, Carrie, Cohen, Vered, Lan, Charline Le, Haridasan, Krishna, Marathe, Amit, Hansen, Steven, Douglas, Sholto, Samuel, Rajkumar, Wang, Mingqiu, Austin, Sophia, Lan, Chang, Jiang, Jiepu, Chiu, Justin, Lorenzo, Jaime Alonso, Sjösund, Lars Lowe, Cevey, Sébastien, Gleicher, Zach, Avrahami, Thi, Boral, Anudhyan, Srinivasan, Hansa, Selo, Vittorio, May, Rhys, Aisopos, Konstantinos, Hussenot, Léonard, Soares, Livio Baldini, Baumli, Kate, Chang, Michael B., Recasens, Adrià, Caine, Ben, Pritzel, Alexander, Pavetic, Filip, Pardo, Fabio, Gergely, Anita, Frye, Justin, Ramasesh, Vinay, Horgan, Dan, Badola, Kartikeya, Kassner, Nora, Roy, Subhrajit, Dyer, Ethan, Campos, Víctor Campos, Tomala, Alex, Tang, Yunhao, Badawy, Dalia El, White, Elspeth, Mustafa, Basil, Lang, Oran, Jindal, Abhishek, Vikram, Sharad, Gong, Zhitao, Caelles, Sergi, Hemsley, Ross, Thornton, Gregory, Feng, Fangxiaoyu, Stokowiec, Wojciech, Zheng, Ce, Thacker, Phoebe, Ünlü, Çağlar, Zhang, Zhishuai, Saleh, Mohammad, Svensson, James, Bileschi, Max, Patil, Piyush, Anand, Ankesh, Ring, Roman, Tsihlas, Katerina, Vezer, Arpi, Selvi, Marco, Shevlane, Toby, Rodriguez, Mikel, Kwiatkowski, Tom, Daruki, Samira, Rong, Keran, Dafoe, Allan, FitzGerald, Nicholas, Gu-Lemberg, Keren, Khan, Mina, Hendricks, Lisa Anne, Pellat, Marie, Feinberg, Vladimir, Cobon-Kerr, James, Sainath, Tara, Rauh, Maribeth, Hashemi, Sayed Hadi, Ives, Richard, Hasson, Yana, Noland, Eric, Cao, Yuan, Byrd, Nathan, Hou, Le, Wang, Qingze, Sottiaux, Thibault, Paganini, Michela, Lespiau, Jean-Baptiste, Moufarek, Alexandre, Hassan, Samer, Shivakumar, Kaushik, Amersfoort, Joost van, Mandhane, Amol, Joshi, Pratik, Goyal, Anirudh, Tung, Matthew, Brock, Andrew, Sheahan, Hannah, Misra, Vedant, Li, Cheng, Rakićević, Nemanja, Dehghani, Mostafa, Liu, Fangyu, Mittal, Sid, Oh, Junhyuk, Noury, Seb, Sezener, Eren, Huot, Fantine, Lamm, Matthew, Cao, Nicola De, Chen, Charlie, Mudgal, Siddharth, Stella, Romina, Brooks, Kevin, Vasudevan, Gautam, Liu, Chenxi, Chain, Mainak, Melinkeri, Nivedita, Cohen, Aaron, Wang, Venus, Seymore, Kristie, Zubkov, Sergey, Goel, Rahul, Yue, Summer, Krishnakumaran, Sai, Albert, Brian, Hurley, Nate, Sano, Motoki, Mohananey, Anhad, Joughin, Jonah, Filonov, Egor, Kępa, Tomasz, Eldawy, Yomna, Lim, Jiawern, Rishi, Rahul, Badiezadegan, Shirin, Bos, Taylor, Chang, Jerry, Jain, Sanil, Padmanabhan, Sri Gayatri Sundara, Puttagunta, Subha, Krishna, Kalpesh, Baker, Leslie, Kalb, Norbert, Bedapudi, Vamsi, Kurzrok, Adam, Lei, Shuntong, Yu, Anthony, Litvin, Oren, Zhou, Xiang, Wu, Zhichun, Sobell, Sam, Siciliano, Andrea, Papir, Alan, Neale, Robby, Bragagnolo, Jonas, Toor, Tej, Chen, Tina, Anklin, Valentin, Wang, Feiran, Feng, Richie, Gholami, Milad, Ling, Kevin, Liu, Lijuan, Walter, Jules, Moghaddam, Hamid, Kishore, Arun, Adamek, Jakub, Mercado, Tyler, Mallinson, Jonathan, Wandekar, Siddhinita, Cagle, Stephen, Ofek, Eran, Garrido, Guillermo, Lombriser, Clemens, Mukha, Maksim, Sun, Botu, Mohammad, Hafeezul Rahman, Matak, Josip, Qian, Yadi, Peswani, Vikas, Janus, Pawel, Yuan, Quan, Schelin, Leif, David, Oana, Garg, Ankur, He, Yifan, Duzhyi, Oleksii, Älgmyr, Anton, Lottaz, Timothée, Li, Qi, Yadav, Vikas, Xu, Luyao, Chinien, Alex, Shivanna, Rakesh, Chuklin, Aleksandr, Li, Josie, Spadine, Carrie, Wolfe, Travis, Mohamed, Kareem, Das, Subhabrata, Dai, Zihang, He, Kyle, Dincklage, Daniel von, Upadhyay, Shyam, Maurya, Akanksha, Chi, Luyan, Krause, Sebastian, Salama, Khalid, Rabinovitch, Pam G, M, Pavan Kumar Reddy, Selvan, Aarush, Dektiarev, Mikhail, Ghiasi, Golnaz, Guven, Erdem, Gupta, Himanshu, Liu, Boyi, Sharma, Deepak, Shtacher, Idan Heimlich, Paul, Shachi, Akerlund, Oscar, Aubet, François-Xavier, Huang, Terry, Zhu, Chen, Zhu, Eric, Teixeira, Elico, Fritze, Matthew, Bertolini, Francesco, Marinescu, Liana-Eleonora, Bölle, Martin, Paulus, Dominik, Gupta, Khyatti, Latkar, Tejasi, Chang, Max, Sanders, Jason, Wilson, Roopa, Wu, Xuewei, Tan, Yi-Xuan, Thiet, Lam Nguyen, Doshi, Tulsee, Lall, Sid, Mishra, Swaroop, Chen, Wanming, Luong, Thang, Benjamin, Seth, Lee, Jasmine, Andrejczuk, Ewa, Rabiej, Dominik, Ranjan, Vipul, Styrc, Krzysztof, Yin, Pengcheng, Simon, Jon, Harriott, Malcolm Rose, Bansal, Mudit, Robsky, Alexei, Bacon, Geoff, Greene, David, Mirylenka, Daniil, Zhou, Chen, Sarvana, Obaid, Goyal, Abhimanyu, Andermatt, Samuel, Siegler, Patrick, Horn, Ben, Israel, Assaf, Pongetti, Francesco, Chen, Chih-Wei "Louis", Selvatici, Marco, Silva, Pedro, Wang, Kathie, Tolins, Jackson, Guu, Kelvin, Yogev, Roey, Cai, Xiaochen, Agostini, Alessandro, Shah, Maulik, Nguyen, Hung, Donnaile, Noah Ó, Pereira, Sébastien, Friso, Linda, Stambler, Adam, Kurzrok, Adam, Kuang, Chenkai, Romanikhin, Yan, Geller, Mark, Yan, ZJ, Jang, Kane, Lee, Cheng-Chun, Fica, Wojciech, Malmi, Eric, Tan, Qijun, Banica, Dan, Balle, Daniel, Pham, Ryan, Huang, Yanping, Avram, Diana, Shi, Hongzhi, Singh, Jasjot, Hidey, Chris, Ahuja, Niharika, Saxena, Pranab, Dooley, Dan, Potharaju, Srividya Pranavi, O'Neill, Eileen, Gokulchandran, Anand, Foley, Ryan, Zhao, Kai, Dusenberry, Mike, Liu, Yuan, Mehta, Pulkit, Kotikalapudi, Ragha, Safranek-Shrader, Chalence, Goodman, Andrew, Kessinger, Joshua, Globen, Eran, Kolhar, Prateek, Gorgolewski, Chris, Ibrahim, Ali, Song, Yang, Eichenbaum, Ali, Brovelli, Thomas, Potluri, Sahitya, Lahoti, Preethi, Baetu, Cip, Ghorbani, Ali, Chen, Charles, Crawford, Andy, Pal, Shalini, Sridhar, Mukund, Gurita, Petru, Mujika, Asier, Petrovski, Igor, Cedoz, Pierre-Louis, Li, Chenmei, Chen, Shiyuan, Santo, Niccolò Dal, Goyal, Siddharth, Punjabi, Jitesh, Kappaganthu, Karthik, Kwak, Chester, LV, Pallavi, Velury, Sarmishta, Choudhury, Himadri, Hall, Jamie, Shah, Premal, Figueira, Ricardo, Thomas, Matt, Lu, Minjie, Zhou, Ting, Kumar, Chintu, Jurdi, Thomas, Chikkerur, Sharat, Ma, Yenai, Yu, Adams, Kwak, Soo, Ähdel, Victor, Rajayogam, Sujeevan, Choma, Travis, Liu, Fei, Barua, Aditya, Ji, Colin, Park, Ji Ho, Hellendoorn, Vincent, Bailey, Alex, Bilal, Taylan, Zhou, Huanjie, Khatir, Mehrdad, Sutton, Charles, Rzadkowski, Wojciech, Macintosh, Fiona, Shagin, Konstantin, Medina, Paul, Liang, Chen, Zhou, Jinjing, Shah, Pararth, Bi, Yingying, Dankovics, Attila, Banga, Shipra, Lehmann, Sabine, Bredesen, Marissa, Lin, Zifan, Hoffmann, John Eric, Lai, Jonathan, Chung, Raynald, Yang, Kai, Balani, Nihal, Bražinskas, Arthur, Sozanschi, Andrei, Hayes, Matthew, Alcalde, Héctor Fernández, Makarov, Peter, Chen, Will, Stella, Antonio, Snijders, Liselotte, Mandl, Michael, Kärrman, Ante, Nowak, Paweł, Wu, Xinyi, Dyck, Alex, Vaidyanathan, Krishnan, R, Raghavender, Mallet, Jessica, Rudominer, Mitch, Johnston, Eric, Mittal, Sushil, Udathu, Akhil, Christensen, Janara, Verma, Vishal, Irving, Zach, Santucci, Andreas, Elsayed, Gamaleldin, Davoodi, Elnaz, Georgiev, Marin, Tenney, Ian, Hua, Nan, Cideron, Geoffrey, Leurent, Edouard, Alnahlawi, Mahmoud, Georgescu, Ionut, Wei, Nan, Zheng, Ivy, Scandinaro, Dylan, Jiang, Heinrich, Snoek, Jasper, Sundararajan, Mukund, Wang, Xuezhi, Ontiveros, Zack, Karo, Itay, Cole, Jeremy, Rajashekhar, Vinu, Tumeh, Lara, Ben-David, Eyal, Jain, Rishub, Uesato, Jonathan, Datta, Romina, Bunyan, Oskar, Wu, Shimu, Zhang, John, Stanczyk, Piotr, Zhang, Ye, Steiner, David, Naskar, Subhajit, Azzam, Michael, Johnson, Matthew, Paszke, Adam, Chiu, Chung-Cheng, Elias, Jaume Sanchez, Mohiuddin, Afroz, Muhammad, Faizan, Miao, Jin, Lee, Andrew, Vieillard, Nino, Park, Jane, Zhang, Jiageng, Stanway, Jeff, Garmon, Drew, Karmarkar, Abhijit, Dong, Zhe, Lee, Jong, Kumar, Aviral, Zhou, Luowei, Evens, Jonathan, Isaac, William, Irving, Geoffrey, Loper, Edward, Fink, Michael, Arkatkar, Isha, Chen, Nanxin, Shafran, Izhak, Petrychenko, Ivan, Chen, Zhe, Jia, Johnson, Levskaya, Anselm, Zhu, Zhenkai, Grabowski, Peter, Mao, Yu, Magni, Alberto, Yao, Kaisheng, Snaider, Javier, Casagrande, Norman, Palmer, Evan, Suganthan, Paul, Castaño, Alfonso, Giannoumis, Irene, Kim, Wooyeol, Rybiński, Mikołaj, Sreevatsa, Ashwin, Prendki, Jennifer, Soergel, David, Goedeckemeyer, Adrian, Gierke, Willi, Jafari, Mohsen, Gaba, Meenu, Wiesner, Jeremy, Wright, Diana Gage, Wei, Yawen, Vashisht, Harsha, Kulizhskaya, Yana, Hoover, Jay, Le, Maigo, Li, Lu, Iwuanyanwu, Chimezie, Liu, Lu, Ramirez, Kevin, Khorlin, Andrey, Cui, Albert, LIN, Tian, Wu, Marcus, Aguilar, Ricardo, Pallo, Keith, Chakladar, Abhishek, Perng, Ginger, Abellan, Elena Allica, Zhang, Mingyang, Dasgupta, Ishita, Kushman, Nate, Penchev, Ivo, Repina, Alena, Wu, Xihui, Weide, Tom van der, Ponnapalli, Priya, Kaplan, Caroline, Simsa, Jiri, Li, Shuangfeng, Dousse, Olivier, Yang, Fan, Piper, Jeff, Ie, Nathan, Pasumarthi, Rama, Lintz, Nathan, Vijayakumar, Anitha, Andor, Daniel, Valenzuela, Pedro, Lui, Minnie, Paduraru, Cosmin, Peng, Daiyi, Lee, Katherine, Zhang, Shuyuan, Greene, Somer, Nguyen, Duc Dung, Kurylowicz, Paula, Hardin, Cassidy, Dixon, Lucas, Janzer, Lili, Choo, Kiam, Feng, Ziqiang, Zhang, Biao, Singhal, Achintya, Du, Dayou, McKinnon, Dan, Antropova, Natasha, Bolukbasi, Tolga, Keller, Orgad, Reid, David, Finchelstein, Daniel, Raad, Maria Abi, Crocker, Remi, Hawkins, Peter, Dadashi, Robert, Gaffney, Colin, Franko, Ken, Bulanova, Anna, Leblond, Rémi, Chung, Shirley, Askham, Harry, Cobo, Luis C., Xu, Kelvin, Fischer, Felix, Xu, Jun, Sorokin, Christina, Alberti, Chris, Lin, Chu-Cheng, Evans, Colin, Dimitriev, Alek, Forbes, Hannah, Banarse, Dylan, Tung, Zora, Omernick, Mark, Bishop, Colton, Sterneck, Rachel, Jain, Rohan, Xia, Jiawei, Amid, Ehsan, Piccinno, Francesco, Wang, Xingyu, Banzal, Praseem, Mankowitz, Daniel J., Polozov, Alex, Krakovna, Victoria, Brown, Sasha, Bateni, MohammadHossein, Duan, Dennis, Firoiu, Vlad, Thotakuri, Meghana, Natan, Tom, Geist, Matthieu, Girgin, Ser tan, Li, Hui, Ye, Jiayu, Roval, Ofir, Tojo, Reiko, Kwong, Michael, Lee-Thorp, James, Yew, Christopher, Sinopalnikov, Danila, Ramos, Sabela, Mellor, John, Sharma, Abhishek, Wu, Kathy, Miller, David, Sonnerat, Nicolas, Vnukov, Denis, Greig, Rory, Beattie, Jennifer, Caveness, Emily, Bai, Libin, Eisenschlos, Julian, Korchemniy, Alex, Tsai, Tomy, Jasarevic, Mimi, Kong, Weize, Dao, Phuong, Zheng, Zeyu, Liu, Frederick, Yang, Fan, Zhu, Rui, Teh, Tian Huey, Sanmiya, Jason, Gladchenko, Evgeny, Trdin, Nejc, Toyama, Daniel, Rosen, Evan, Tavakkol, Sasan, Xue, Linting, Elkind, Chen, Woodman, Oliver, Carpenter, John, Papamakarios, George, Kemp, Rupert, Kafle, Sushant, Grunina, Tanya, Sinha, Rishika, Talbert, Alice, Wu, Diane, Owusu-Afriyie, Denese, Du, Cosmo, Thornton, Chloe, Pont-Tuset, Jordi, Narayana, Pradyumna, Li, Jing, Fatehi, Saaber, Wieting, John, Ajmeri, Omar, Uria, Benigno, Ko, Yeongil, Knight, Laura, Héliou, Amélie, Niu, Ning, Gu, Shane, Pang, Chenxi, Li, Yeqing, Levine, Nir, Stolovich, Ariel, Santamaria-Fernandez, Rebeca, Goenka, Sonam, Yustalim, Wenny, Strudel, Robin, Elqursh, Ali, Deck, Charlie, Lee, Hyo, Li, Zonglin, Levin, Kyle, Hoffmann, Raphael, Holtmann-Rice, Dan, Bachem, Olivier, Arora, Sho, Koh, Christy, Yeganeh, Soheil Hassas, Põder, Siim, Tariq, Mukarram, Sun, Yanhua, Ionita, Lucian, Seyedhosseini, Mojtaba, Tafti, Pouya, Liu, Zhiyu, Gulati, Anmol, Liu, Jasmine, Ye, Xinyu, Chrzaszcz, Bart, Wang, Lily, Sethi, Nikhil, Li, Tianrun, Brown, Ben, Singh, Shreya, Fan, Wei, Parisi, Aaron, Stanton, Joe, Koverkathu, Vinod, Choquette-Choo, Christopher A., Li, Yunjie, Lu, TJ, Ittycheriah, Abe, Shroff, Prakash, Varadarajan, Mani, Bahargam, Sanaz, Willoughby, Rob, Gaddy, David, Desjardins, Guillaume, Cornero, Marco, Robenek, Brona, Mittal, Bhavishya, Albrecht, Ben, Shenoy, Ashish, Moiseev, Fedor, Jacobsson, Henrik, Ghaffarkhah, Alireza, Rivière, Morgane, Walton, Alanna, Crepy, Clément, Parrish, Alicia, Zhou, Zongwei, Farabet, Clement, Radebaugh, Carey, Srinivasan, Praveen, Salm, Claudia van der, Fidjeland, Andreas, Scellato, Salvatore, Latorre-Chimoto, Eri, Klimczak-Plucińska, Hanna, Bridson, David, Cesare, Dario de, Hudson, Tom, Mendolicchio, Piermaria, Walker, Lexi, Morris, Alex, Mauger, Matthew, Guseynov, Alexey, Reid, Alison, Odoom, Seth, Loher, Lucia, Cotruta, Victor, Yenugula, Madhavi, Grewe, Dominik, Petrushkina, Anastasia, Duerig, Tom, Sanchez, Antonio, Yadlowsky, Steve, Shen, Amy, Globerson, Amir, Webb, Lynette, Dua, Sahil, Li, Dong, Bhupatiraju, Surya, Hurt, Dan, Qureshi, Haroon, Agarwal, Ananth, Shani, Tomer, Eyal, Matan, Khare, Anuj, Belle, Shreyas Rammohan, Wang, Lei, Tekur, Chetan, Kale, Mihir Sanjay, Wei, Jinliang, Sang, Ruoxin, Saeta, Brennan, Liechty, Tyler, Sun, Yi, Zhao, Yao, Lee, Stephan, Nayak, Pandu, Fritz, Doug, Vuyyuru, Manish Reddy, Aslanides, John, Vyas, Nidhi, Wicke, Martin, Ma, Xiao, Eltyshev, Evgenii, Martin, Nina, Cate, Hardie, Manyika, James, Amiri, Keyvan, Kim, Yelin, Xiong, Xi, Kang, Kai, Luisier, Florian, Tripuraneni, Nilesh, Madras, David, Guo, Mandy, Waters, Austin, Wang, Oliver, Ainslie, Joshua, Baldridge, Jason, Zhang, Han, Pruthi, Garima, Bauer, Jakob, Yang, Feng, Mansour, Riham, Gelman, Jason, Xu, Yang, Polovets, George, Liu, Ji, Cai, Honglong, Chen, Warren, Sheng, XiangHai, Xue, Emily, Ozair, Sherjil, Angermueller, Christof, Li, Xiaowei, Sinha, Anoop, Wang, Weiren, Wiesinger, Julia, Koukoumidis, Emmanouil, Tian, Yuan, Iyer, Anand, Gurumurthy, Madhu, Goldenson, Mark, Shah, Parashar, Blake, MK, Yu, Hongkun, Urbanowicz, Anthony, Palomaki, Jennimaria, Fernando, Chrisantha, Durden, Ken, Mehta, Harsh, Momchev, Nikola, Rahimtoroghi, Elahe, Georgaki, Maria, Raul, Amit, Ruder, Sebastian, Redshaw, Morgan, Lee, Jinhyuk, Zhou, Denny, Jalan, Komal, Li, Dinghua, Hechtman, Blake, Schuh, Parker, Nasr, Milad, Milan, Kieran, Mikulik, Vladimir, Franco, Juliana, Green, Tim, Nguyen, Nam, Kelley, Joe, Mahendru, Aroma, Hu, Andrea, Howland, Joshua, Vargas, Ben, Hui, Jeffrey, Bansal, Kshitij, Rao, Vikram, Ghiya, Rakesh, Wang, Emma, Ye, Ke, Sarr, Jean Michel, Preston, Melanie Moranski, Elish, Madeleine, Li, Steve, Kaku, Aakash, Gupta, Jigar, Pasupat, Ice, Juan, Da-Cheng, Someswar, Milan, M., Tejvi, Chen, Xinyun, Amini, Aida, Fabrikant, Alex, Chu, Eric, Dong, Xuanyi, Muthal, Amruta, Buthpitiya, Senaka, Jauhari, Sarthak, Hua, Nan, Khandelwal, Urvashi, Hitron, Ayal, Ren, Jie, Rinaldi, Larissa, Drath, Shahar, Dabush, Avigail, Jiang, Nan-Jiang, Godhia, Harshal, Sachs, Uli, Chen, Anthony, Fan, Yicheng, Taitelbaum, Hagai, Noga, Hila, Dai, Zhuyun, Wang, James, Liang, Chen, Hamer, Jenny, Ferng, Chun-Sung, Elkind, Chenel, Atias, Aviel, Lee, Paulina, Listík, Vít, Carlen, Mathias, Kerkhof, Jan van de, Pikus, Marcin, Zaher, Krunoslav, Müller, Paul, Zykova, Sasha, Stefanec, Richard, Gatsko, Vitaly, Hirnschall, Christoph, Sethi, Ashwin, Xu, Xingyu Federico, Ahuja, Chetan, Tsai, Beth, Stefanoiu, Anca, Feng, Bo, Dhandhania, Keshav, Katyal, Manish, Gupta, Akshay, Parulekar, Atharva, Pitta, Divya, Zhao, Jing, Bhatia, Vivaan, Bhavnani, Yashodha, Alhadlaq, Omar, Li, Xiaolin, Danenberg, Peter, Tu, Dennis, Pine, Alex, Filippova, Vera, Ghosh, Abhipso, Limonchik, Ben, Urala, Bhargava, Lanka, Chaitanya Krishna, Clive, Derik, Sun, Yi, Li, Edward, Wu, Hao, Hongtongsak, Kevin, Li, Ianna, Thakkar, Kalind, Omarov, Kuanysh, Majmundar, Kushal, Alverson, Michael, Kucharski, Michael, Patel, Mohak, Jain, Mudit, Zabelin, Maksim, Pelagatti, Paolo, Kohli, Rohan, Kumar, Saurabh, Kim, Joseph, Sankar, Swetha, Shah, Vineet, Ramachandruni, Lakshmi, Zeng, Xiangkai, Bariach, Ben, Weidinger, Laura, Vu, Tu, Andreev, Alek, He, Antoine, Hui, Kevin, Kashem, Sheleem, Subramanya, Amar, Hsiao, Sissie, Hassabis, Demis, Kavukcuoglu, Koray, Sadovsky, Adam, Le, Quoc, Strohman, Trevor, Wu, Yonghui, Petrov, Slav, Dean, Jeffrey, and Vinyals, Oriol. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805.

[341] Theodoropoulos, Nikitas, Filandrianos, Giorgos, Lyberatos, Vassilis, Lymperaiou, Maria, and Stamou, Giorgos. “BERTtime Stories: Investigating the Role of Synthetic Story Data in Language Pre-training”. In: *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*. Ed. by Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox. Miami, FL, USA: Association for Computational Linguistics, Nov. 2024, pp. 308–323. URL:

[342] Thomas, Konstantinos, Filandrianos, Giorgos, Lymperaiou, Maria, Zerva, Chrysoula, and Stamou, Giorgos. “” I Never Said That”: A dataset, taxonomy and baselines on response clarity classification”. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. 2024, pp. 5204–5233.

[343] Tong, Yongqi, Wang, Yifan, Li, Dawei, Wang, Sizhe, Lin, Zi, Han, Simeng, and Shang, Jingbo. “Eliminating Reasoning via Inferring with Planning: A New Framework to Guide LLMs’ Non-linear Thinking”. In: *ArXiv* abs/2310.12342 (2023). URL:

[344] Tonmoy, S. M Towhidul Islam, Zaman, S M Mehedi, Jain, Vinija, Rani, Anku, Rawte, Vipula, Chadha, Aman, and Das, Amitava. *A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models*. 2024. arXiv: 2401.01313 [cs.CL].

[345] Touvron, Hugo, Martin, Louis, Stone, Kevin, Albert, Peter, Almahairi, Amjad, Babaei, Yasmine, Bashlykov, Nikolay, Batra, Soumya, Bhargava, Prajjwal, Bhosale, Shruti, Bikel, Dan, Blecher, Lukas, Ferrer, Cristian Canton, Chen, Moya, Cucurull, Guillem, Esiobu, David, Fernandes, Jude, Fu, Jeremy, Fu, Wenyin, Fuller, Brian, Gao, Cynthia, Goswami, Vedanuj, Goyal, Naman, Hartshorn, Anthony, Hosseini, Saghar, Hou, Rui, Inan, Hakan, Kardas, Marcin, Kerkez, Viktor, Khabsa, Madian, Kloumann,

Isabel, Korenev, Artem, Koura, Punit Singh, Lachaux, Marie-Anne, Lavril, Thibaut, Lee, Jenya, Liskovich, Diana, Lu, Yinghai, Mao, Yuning, Martinet, Xavier, Mihaylov, Todor, Mishra, Pushkar, Molybog, Igor, Nie, Yixin, Poulton, Andrew, Reizenstein, Jeremy, Rungta, Rashi, Saladi, Kalyan, Schelten, Alan, Silva, Ruan, Smith, Eric Michael, Subramanian, Ranjan, Tan, Xiaoqing Ellen, Tang, Binh, Taylor, Ross, Williams, Adina, Kuan, Jian Xiang, Xu, Puxin, Yan, Zheng, Zarov, Iliyan, Zhang, Yuchen, Fan, Angela, Kambadur, Melanie, Narang, Sharan, Rodriguez, Aurelien, Stojnic, Robert, Edunov, Sergey, and Scialom, Thomas. *Llama 2: Open Foundation and Fine-Tuned Chat Models.* 2023. arXiv: 2307.09288.

[346] Tsakas, Nikolaos, Lymperaiou, Maria, Filandrianos, Giorgos, and Stamou, Giorgos. "An Impartial Transformer for Story Visualization". In: *arXiv preprint arXiv:2301.03563* (2023).

[347] Tsakas, Nikolaos, Lymperaiou, Maria, Filandrianos, Giorgos, and Stamou, Giorgos. *An Impartial Transformer for Story Visualization.* 2023. DOI: 10.48550/ARXIV.2301.03563. URL:

[348] Vandenhende, Simon, Mahajan, Dhruv, Radenovic, Filip, and Ghadiyaram, Deepti. "Making heads or tails: Towards semantically consistent visual counterfactuals". In: *European Conference on Computer Vision*. Springer. 2022, pp. 261–279.

[349] Vasilakes, Jake, Zerva, Chrysoula, Miwa, Makoto, and Ananiadou, Sophia. "Learning Disentangled Representations of Negation and Uncertainty". In: *arXiv preprint arXiv:2204.00511* (2022).

[350] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, and Polosukhin, Illia. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL:

[351] Vedantam, Ramakrishna, Zitnick, C. Lawrence, and Parikh, Devi. *CIDEr: Consensus-based Image Description Evaluation.* 2015. arXiv: 1411.5726 [cs.CV].

[352] Veličković, Petar, Cucurull, Guillem, Casanova, Arantxa, Romero, Adriana, Lio, Pietro, and Bengio, Yoshua. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[353] Verma, Sahil, Boonsanong, Varich, Hoang, Minh, Hines, Keegan E., Dickerson, John P., and Shah, Chirag. *Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review.* 2022. arXiv: 2010.10596 [cs.LG].

[354] Verma, Sahil, Dickerson, John, and Hines, Keegan. "Counterfactual explanations for machine learning: Challenges revisited". In: *arXiv preprint arXiv:2106.07756* (2021).

[355] Wah, Catherine, Branson, Steve, Welinder, Peter, Perona, Pietro, and Belongie, Serge. "The caltech-ucsd birds-200-2011 dataset". In: (2011).

[356] Wang, Alex and Cho, Kyunghyun. "BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model". In: *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. 2019, pp. 30–36.

[357] Wang, Ao, Chen, Hui, Liu, Lihao, Chen, Kai, Lin, Zijia, Han, Jungong, and Ding, Guiguang. "Yolov10: Real-time end-to-end object detection". In: *arXiv preprint arXiv:2405.14458* (2024).

[358] Wang, Haoyu, Ma, Guozheng, Yu, Cong, Gui, Ning, Zhang, Linrui, Huang, Zhiqi, Ma, Suwei, Chang, Yongzhe, Zhang, Sen, Shen, Li, Wang, Xueqian, Zhao, Peilin, and Tao, Dacheng. *Are Large Language Models Really Robust to Word-Level Perturbations?* 2023. arXiv: 2309.11166 [cs.CL]. URL:

[359] Wang, Jianfeng, Yang, Zhengyuan, Hu, Xiaowei, Li, Linjie, Lin, Kevin, Gan, Zhe, Liu, Zicheng, Liu, Ce, and Wang, Lijuan. "GIT: A Generative Image-to-text Transformer for Vision and Language". In: *ArXiv* abs/2205.14100 (2022). URL:

[360] Wang, Jindong, Hu, Xixu, Hou, Wenxin, Chen, Hao, Zheng, Runkai, Wang, Yidong, Yang, Linyi, Huang, Haojun, Ye, Wei, Geng, Xiubo, Jiao, Binxin, Zhang, Yue, and Xie, Xing. *On the Robustness of ChatGPT: An Adversarial and Out-of-distribution Perspective.* 2023. arXiv: 2302.12095 [cs.AI]. URL:

[361] Wang, Junyang, Zhou, Yiyang, Xu, Guohai, Shi, Pengcheng, Zhao, Chenlin, Xu, Haiyang, Ye, Qinghao, Yan, Ming, Zhang, Ji, Zhu, Jihua, Sang, Jitao, and Tang, Haoyu. *Evaluation and Analysis of Hallucination in Large Vision-Language Models.* 2023. arXiv: 2308.15126 [cs.LG].

[362] Wang, Lei, Xu, Wanyu, Lan, Yihuai, Hu, Zhiqiang, Lan, Yunshi, Lee, Roy Ka-Wei, and Lim, Ee-Peng. "Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models". In: *Annual Meeting of the Association for Computational Linguistics*. 2023. URL:

[363] Wang, Liang, Yang, Nan, and Wei, Furu. "Learning to Retrieve In-Context Examples for Large Language Models". In: *Proceedings of the 18th Conference of the European Chapter of the Association for*

Computational Linguistics (Volume 1: Long Papers). Ed. by Yvette Graham and Matthew Purver. St. Julian's, Malta: Association for Computational Linguistics, Mar. 2024, pp. 1752–1767. URL:

[364] Wang, Shenzhi, Liu, Chang, Zheng, Zilong, Qi, Siyuan, Chen, Shuo, Yang, Qisen, Zhao, Andrew, Wang, Chaofei, Song, Shiji, and Huang, Gao. "Avalon's Game of Thoughts: Battle Against Deception through Recursive Contemplation". In: *ArXiv* abs/2310.01320 (2023). URL:

[365] Wang, Wenhui, Bao, Hangbo, Dong, Li, Bjorck, Johan, Peng, Zhiliang, Liu, Qiang, Aggarwal, Kriti, Mohammed, Owais Khan, Singhal, Saksham, Som, Subhojit, and Wei, Furu. "Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks". In: *ArXiv* abs/2208.10442 (2022). URL:

[366] Wang, Xuezhi, Wei, Jason, Schuurmans, Dale, Le, Quoc, Chi, Ed Huai-hsin, and Zhou, Denny. "Self-Consistency Improves Chain of Thought Reasoning in Language Models". In: *ArXiv* abs/2203.11171 (2022). URL:

[367] Wang, Zhao and Culotta, Aron. "Robustness to spurious correlations in text classification via automatically generated counterfactuals". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 16. 2021, pp. 14024–14031.

[368] Wei, Alexander, Haghtalab, Nika, and Steinhardt, Jacob. *Jailbroken: How Does LLM Safety Training Fail?* 2023. arXiv: 2307.02483 [cs.LG]. URL:

[369] Wei, Jason, Tay, Yi, Bommasani, Rishi, Raffel, Colin, Zoph, Barret, Borgeaud, Sebastian, Yogatama, Dani, Bosma, Maarten, Zhou, Denny, Metzler, Donald, Chi, Ed H., Hashimoto, Tatsunori, Vinyals, Oriol, Liang, Percy, Dean, Jeff, and Fedus, William. *Emergent Abilities of Large Language Models*. 2022. arXiv: 2206.07682 [cs.CL]. URL:

[370] Wei, Jason, Wang, Xuezhi, Schuurmans, Dale, Bosma, Maarten, Chi, Ed Huai-hsin, Xia, F., Le, Quoc, and Zhou, Denny. "Chain of Thought Prompting Elicits Reasoning in Large Language Models". In: *ArXiv* abs/2201.11903 (2022). URL:

[371] Wei, Jason, Wang, Xuezhi, Schuurmans, Dale, Bosma, Maarten, Ichter, Brian, Xia, Fei, Chi, Ed, Le, Quoc, and Zhou, Denny. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL]. URL:

[372] Wen, Jeffrey, Benitez-Quiroz, Fabian, Feng, Qianli, and Martinez, Aleix M. "Diamond in the rough: Improving image realism by traversing the GAN latent space". In: *ArXiv* abs/2104.05518 (2021).

[373] Wu, Hao, Mao, Jiayuan, Zhang, Yufeng, Jiang, Yuning, Li, Lei, Sun, Weiwei, and Ma, Wei-Ying. "Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6609–6618.

[374] Wu, Tongshuang, Ribeiro, Marco Tulio, Heer, Jeffrey, and Weld, Daniel. "Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, Aug. 2021, pp. 6707–6723. DOI: 10.18653/v1/2021.acl-long.523. URL:

[375] Wu, Tongshuang, Ribeiro, Marco Tulio, Heer, Jeffrey, and Weld, Daniel S. "Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models". In: *arXiv preprint arXiv:2101.00288* (2021).

[376] Wu, Zhibiao and Palmer, Martha. "Verb semantics and lexical selection". In: *arXiv preprint cmp-lg/9406033* (1994).

[377] Wu, Zhiyong, Wang, Yaoxiang, Ye, Jiacheng, and Kong, Lingpeng. "Self-Adaptive In-Context Learning: An Information Compression Perspective for In-Context Example Selection and Ordering". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 1423–1436. DOI: 10.18653/v1/2023.acl-long.79. URL:

[378] Wu, Zonghan, Pan, Shirui, Chen, Fengwen, Long, Guodong, Zhang, Chengqi, and Yu, Philip S. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 32 (2019), pp. 4–24. URL:

[379] Xi, Yunjia, Liu, Weiwen, Lin, Jianghao, Cai, Xiaoling, Zhu, Hong, Zhu, Jieming, Chen, Bo, Tang, Ruiming, Zhang, Weinan, and Yu, Yong. "Towards Open-World Recommendation with Knowledge

Augmentation from Large Language Models". In: *Proceedings of the 18th ACM Conference on Recommender Systems*. RecSys '24. Bari, Italy: Association for Computing Machinery, 2024, pp. 12–22. ISBN: 9798400705052. DOI: 10.1145/3640457.3688104. URL:

[380] Xia, Tong, Spathis, Dimitris, Ch, J, Grammenos, Andreas, Han, Jing, Hasthanasombat, Apinan, Bondareva, Erika, Dang, Ting, Floto, Andres, Cicuta, Pietro, et al. "COVID-19 sounds: a large-scale audio dataset for digital respiratory screening". In: *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*. 2021.

[381] Xie, Cihang, Tan, Mingxing, Gong, Boqing, Wang, Jiang, Yuille, Alan, and Le, Quoc V. "Adversarial Examples Improve Image Recognition". In: 2020. URL:

[382] Xu, Fan, Zhang, Yunxiang, and Wan, Xiao-Yi. "CC-Riddle: A Question Answering Dataset of Chinese Character Riddles". In: *ArXiv* abs/2206.13778 (2022). URL:

[383] Xu, Fangzhi, Lin, Qika, Han, Jiawei, Zhao, Tianzhe, Liu, Jun, and Cambria, Erik. *Are Large Language Models Really Good Logical Reasoners? A Comprehensive Evaluation and Beyond*. 2023. arXiv: 2306.09841 [cs.CL].

[384] Xu, Keyulu, Hu, Weihua, Leskovec, Jure, and Jegelka, Stefanie. "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826* (2018).

[385] Xu, Yuzhuang, Wang, Shuo, Li, Peng, Luo, Fuwen, Wang, Xiaolong, Liu, Weidong, and Liu, Yang. "Exploring Large Language Models for Communication Games: An Empirical Study on Werewolf". In: *ArXiv* abs/2309.04658 (2023). URL:

[386] Yan, Junchi, Yin, Xu-Cheng, Lin, Weiyao, Deng, Cheng, Zha, Hongyuan, and Yang, Xiaokang. "A Short Survey of Recent Advances in Graph Matching". In: *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ICMR '16. New York, New York, USA: Association for Computing Machinery, 2016, pp. 167–174. ISBN: 9781450343596. DOI: 10.1145/2911996.2912035. URL:

[387] Yang, An, Yang, Baosong, Hui, Binyuan, Zheng, Bo, Yu, Bowen, Zhou, Chang, Li, Chengpeng, Li, Chengyuan, Liu, Dayiheng, Huang, Fei, Dong, Guanting, Wei, Haoran, Lin, Huan, Tang, Jialong, Wang, Jialin, Yang, Jian, Tu, Jianhong, Zhang, Jianwei, Ma, Jianxin, Yang, Jianxin, Xu, Jin, Zhou, Jingren, Bai, Jinze, He, Jinzheng, Lin, Junyang, Dang, Kai, Lu, Keming, Chen, Keqin, Yang, Kexin, Li, Mei, Xue, Mingfeng, Ni, Na, Zhang, Pei, Wang, Peng, Peng, Ru, Men, Rui, Gao, Ruize, Lin, Runji, Wang, Shijie, Bai, Shuai, Tan, Sinan, Zhu, Tianhang, Li, Tianhao, Liu, Tianyu, Ge, Wenbin, Deng, Xiaodong, Zhou, Xiaohuan, Ren, Xingzhang, Zhang, Xinyu, Wei, Xipin, Ren, Xuancheng, Liu, Xuejing, Fan, Yang, Yao, Yang, Zhang, Yichang, Wan, Yu, Chu, Yunfei, Liu, Yuqiong, Cui, Zeyu, Zhang, Zhenru, Guo, Zhifang, and Fan, Zhihao. *Qwen2 Technical Report*. 2024. arXiv: 2407.10671 [cs.CL]. URL:

[388] Yang, Fan, Chen, Zheng, Jiang, Ziyan, Cho, Eunah, Huang, Xiaojiang, and Lu, Yanbin. *PALR: Personalization Aware LLMs for Recommendation*. 2023. arXiv: 2305.07622 [cs.IR]. URL:

[389] Yang, Sen, Li, Xin, Cui, Leyang, Bing, Li, and Lam, Wai. "Neuro-Symbolic Integration Brings Causal and Reliable Reasoning Proofs". In: *ArXiv* abs/2311.09802 (2023). URL:

[390] Yang, Zonglin, Du, Xinya, Mao, Rui, Ni, Jinjie, and Cambria, E. "Logical Reasoning over Natural Language as Knowledge Representation: A Survey". In: *ArXiv* abs/2303.12023 (2023). URL:

[391] Yao, Shunyu, Yu, Dian, Zhao, Jeffrey, Shafran, Izhak, Griffiths, Thomas L., Cao, Yuan, and Narasimhan, Karthik. "Tree of Thoughts: Deliberate Problem Solving with Large Language Models". In: *ArXiv* abs/2305.10601 (2023). URL:

[392] Ye, Jiayi, Wang, Yanbo, Huang, Yue, Chen, Dongping, Zhang, Qihui, Moniz, Nuno, Gao, Tian, Geyer, Werner, Huang, Chao, Chen, Pin-Yu, Chawla, Nitesh V, and Zhang, Xiangliang. "Justice or Prejudice? Quantifying Biases in LLM-as-a-Judge". In: *arXiv preprint arXiv:2410.02736* (2024).

[393] Yow, Kai Siong and Luo, Siqiang. *Learning-Based Approaches for Graph Problems: A Survey*. Apr. 2022.

[394] Yu, Fei, Zhang, Hongbo, Tiwari, Prayag, and Wang, Benyou. *Natural Language Reasoning, A Survey*. 2023. arXiv: 2303.14725 [cs.CL].

[395] Yu, Zihan, He, Liang, Wu, Zhen, Dai, Xinyu, and Chen, Jiajun. "Towards Better Chain-of-Thought Prompting Strategies: A Survey". In: *ArXiv* abs/2310.04959 (2023). URL:

[396] Zang, Yuan, Qi, Fanchao, Yang, Chenghao, Liu, Zhiyuan, Zhang, Meng, Liu, Qun, and Sun, Maosong. "Word-level Textual Adversarial Attacking as Combinatorial Optimization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai,

Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 6066–6080. DOI: 10.18653/v1/2020.acl-main.540. URL:

[397] Zareian, Alireza, Wang, Zhecan, You, Haoxuan, and Chang, Shih-Fu. *Learning Visual Commonsense for Robust Scene Graph Generation*. 2020. arXiv: 2006.09623 [cs.CV].

[398] Zarkogianni, Konstantia, Dervakos, Edmund, Filandrianos, George, Ganitidis, Theofanis, Gkatzou, Vasiliki, Sakagianni, Aikaterini, Raghavendra, Raghu, Max Nikias, CL, Stamou, Giorgos, and Nikita, Konstantina S. "The smarty4covid dataset and knowledge base as a framework for interpretable physiological audio data analysis". In: *Scientific data* 10.1 (2023), p. 770.

[399] Zeinalipour, Kamyar, laquinta, Tommaso, Zanollo, Asya, Angelini, Giovanni, Rigutini, Leonardo, Maggini, Marco, and Gori, Marco. *Italian Crossword Generator: Enhancing Education through Interactive Word Puzzles*. 2023. arXiv: 2311.15723 [cs.CL].

[400] Zeinalipour, Kamyar, Saad, Mohamed, Maggini, Marco, and Gori, Marco. "ArabIcros: AI-Powered Arabic Crossword Puzzle Generation for Educational Applications". In: *Proceedings of ArabicNLP 2023*. Association for Computational Linguistics, 2023. DOI: 10.18653/v1/2023.arabicnlp-1.23. URL:

[401] Zeng, Zhiping, Tung, Anthony KH, Wang, Jianyong, Feng, Jianhua, and Zhou, Lizhu. "Comparing stars: On approximating graph edit distance". In: *Proceedings of the VLDB Endowment* 2.1 (2009), pp. 25–36.

[402] Zhang, Han, Goodfellow, Ian, Metaxas, Dimitris, and Odena, Augustus. *Self-Attention Generative Adversarial Networks*. 2019. arXiv: 1805.08318 [stat.ML].

[403] Zhang, Renrui, Han, Jiaming, Liu, Chris, Zhou, Aojun, Lu, Pan, Qiao, Yu, Li, Hongsheng, and Gao, Peng. "LLaMA-adapter: Efficient fine-tuning of large language models with zero-initialized attention". In: *The Twelfth International Conference on Learning Representations*. 2024.

[404] Zhang, Richard, Isola, Phillip, Efros, Alexei A, Shechtman, Eli, and Wang, Oliver. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *CVPR*. 2018.

[405] Zhang, Tianyi, Kishore, Varsha, Wu, Felix, Weinberger, Kilian Q., and Artzi, Yoav. "BERTScore: Evaluating Text Generation with BERT". In: *ArXiv* abs/1904.09675 (2019). URL:

[406] Zhang, Tianyi, Kishore, Varsha, Wu, Felix, Weinberger, Kilian Q., and Artzi, Yoav. "BERTScore: Evaluating Text Generation with BERT". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL:

[407] Zhang, Wei Emma, Sheng, Quan Z., Alhazmi, Ahoud, and Li, Chenliang. "Adversarial attacks on deep-learning models in natural language processing: a survey". English. In: *ACM Transactions on Intelligent Systems and Technology* 11.3 (June 2020), pp. 1–41. ISSN: 2157-6904. DOI: 10.1145/3374217.

[408] Zhang, Xin, Zhang, Yanzhao, Long, Dingkun, Xie, Wen, Dai, Ziqi, Tang, Jialong, Lin, Huan, Yang, Baosong, Xie, Pengjun, Huang, Fei, Zhang, Meishan, Li, Wenjie, and Zhang, Min. *mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval*. 2024. arXiv: 2407.19669 [cs.CL]. URL:

[409] Zhang, Yue, Li, Yafu, Cui, Leyang, Cai, Deng, Liu, Lemao, Fu, Tingchen, Huang, Xinting, Zhao, Enbo, Zhang, Yu, Chen, Yulong, Wang, Longyue, Luu, Anh Tuan, Bi, Wei, Shi, Freda, and Shi, Shuming. *Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models*. 2023. arXiv: 2309.01219 [cs.CL].

[410] Zhang, Yunxiang and Wan, Xiaojun. "BiRdQA: A Bilingual Dataset for Question Answering on Tricky Riddles". In: *ArXiv* abs/2109.11087 (2021). URL:

[411] Zhang, Yunxiang and Wan, Xiaojun. *BiRdQA: A Bilingual Dataset for Question Answering on Tricky Riddles*. 2022. arXiv: 2109.11087 [cs.CL]. URL:

[412] Zhang, Zhuosheng, Zhang, Aston, Li, Mu, and Smola, Alex. *Automatic Chain of Thought Prompting in Large Language Models*. 2022. arXiv: 2210.03493 [cs.CL]. URL:

[413] Zhang, Zhuosheng, Zhang, Aston, Li, Mu, and Smola, Alexander J. "Automatic Chain of Thought Prompting in Large Language Models". In: *ArXiv* abs/2210.03493 (2022). URL:

[414] Zhao, Jingmiao and Anderson, Carolyn Jane. *Solving and Generating NPR Sunday Puzzles with Large Language Models*. 2023. arXiv: 2306.12255 [cs.CL].

[415] Zhao, Wenqi, Oyama, Satoshi, and Kurihara, Masahito. "Generating natural counterfactual visual explanations". In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 2021, pp. 5204–5205.

[416] Zhao, Wenting, Chiu, Justin, Cardie, Claire, and Rush, Alexander. “Abductive Commonsense Reasoning Exploiting Mutually Exclusive Explanations”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 14883–14896. DOI: 10.18653/v1/2023.acl-long.831. URL:

[417] Zhao, Xuandong, Yang, Xianjun, Pang, Tianyu, Du, Chao, Li, Lei, Wang, Yu-Xiang, and Wang, William Yang. *Weak-to-Strong Jailbreaking on Large Language Models*. 2024. arXiv: 2401.17256 [cs.CL]. URL:

[418] Zhao, Zhiyuan, Wang, Bin, Ouyang, Linke, Dong, Xiaoyi, Wang, Jiaqi, and He, Conghui. *Beyond Hallucinations: Enhancing LVLMs through Hallucination-Aware Direct Preference Optimization*. 2023. arXiv: 2311.16839 [cs.CV].

[419] Zhou, Bolei, Lapedriza, Agata, Khosla, Aditya, Oliva, Aude, and Torralba, Antonio. “Places: A 10 Million Image Database for Scene Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1452–1464. DOI: 10.1109/TPAMI.2017.2723009.

[420] Zhou, Bolei, Lapedriza, Àgata, Khosla, Aditya, Oliva, Aude, and Torralba, Antonio. “Places: A 10 Million Image Database for Scene Recognition”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.6 (2018), pp. 1452–1464.

[421] Zhou, Yiyang, Cui, Chenhang, Yoon, Jaehong, Zhang, Linjun, Deng, Zhun, Finn, Chelsea, Bansal, Mohit, and Yao, Huaxiu. “Analyzing and Mitigating Object Hallucination in Large Vision-Language Models”. In: *ArXiv* abs/2310.00754 (2023). URL:

[422] Zhou, Yongchao, Muresanu, Andrei Ioan, Han, Ziwen, Paster, Keiran, Pitis, Silviu, Chan, Harris, and Ba, Jimmy. “Large Language Models Are Human-Level Prompt Engineers”. In: *ArXiv* abs/2211.01910 (2022). URL:

[423] Zhu, Deyao, Chen, Jun, Shen, Xiaoqian, Li, Xiang, and Elhoseiny, Mohamed. *MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models*. 2023. arXiv: 2304.10592 [cs.CV].

[424] Zhu, Hai, Zhao, Qingyang, and Wu, Yuren. “BeamAttack: Generating High-quality Textual Adversarial Examples through Beam Search and Mixed Semantic Spaces”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 2023. URL:

[425] Zhu, Yaoming, Lu, Sidi, Zheng, Lei, Guo, Jiaxian, Zhang, Weinan, Wang, Jun, and Yu, Yong. “Texygen: A benchmarking platform for text generation models”. In: *The 41st international ACM SIGIR conference on research & development in information retrieval*. 2018, pp. 1097–1100.

[426] Zugarini, Andrea, Zeinalipour, Kamyar, Kadali, Surya Sai, Maggini, Marco, Gori, Marco, and Rigutini, Leonardo. *Clue-Instruct: Text-Based Clue Generation for Educational Crossword Puzzles*. 2024. arXiv: 2404.06186 [cs.CL].