Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Εργαστηριο Ψηφιακης Επεξεργασιας Εικονας και Σηματων

Attention-based Story Visualization

Diploma Thesis

by

Nikos Tsakas

Επιβλέπων: Γεώργιος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2021

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Πληροφορικής
Εργαστήριο Ψηφιακής Επεξεργασίας Εικόνας και Σημάτων

Attention-based Story Visualization

Diploma Thesis

by

Nikos Tsakas

Επιβλέπων: Γεώργιος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1η Μαρτίου, 2021.

.....	.....	.....
Γεώργιος Στάμου	Αθανάσιος Βουλόδημος	Ανδρέας Γεώργιος Σταφυλοπάτης
Αν. Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2021

.....

Νικολαος Τσακας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η Οπτικοποίηση Ιστορίας είναι ένα πρόσφατα προταθέν πρόβλημα τεχνητής νοημοσύνης που συνδυάζει προκλήσεις από τα πεδία της όρασης υπολογιστών, της επεξεργασίας φυσικής γλώσσας και της γεννητικής τεχνητής νοημοσύνης. Ο στόχος είναι να δημιουργηθεί ένα σύστημα που να παράγει μια ακολουθία εικόνων από μια "ιστορία" εισόδου που αποτελείται από προτάσεις φυσικής γλώσσας ή άλλη σειριακή πληροφορία. Οι εικόνες πρέπει να αντιστοιχούν στις προτάσεις μία προς μία, να είναι ικανοποιητικά ρεαλιστικές και να διατηρούν μια αίσθηση συνέπειας και σειριακής προόδου. Αυτό το πρόβλημα σχετίζεται στενά με τα πεδία παραγωγής εικόνας από κείμενο και βίντεο από κείμενο.

Η εργασία που εισάγει το πρόβλημα [29] προτείνει επίσης την πρώτη αρχιτεκτονική που το προσεγγίζει: ένα Γεννητικό Ανταγωνιστικό Δίκτυο (GAN), του οποίου προηγείται ένα σχήμα κωδικοποίησης της ιστορίας βασισμένο σε Αναδρομικά Νευρωνικά Δίκτυα (RNN). Οι ενσωματωμένες προτάσεις τροφοδοτούνται διαδοχικά σε μια στοίβα αναδρομικών μονάδων μαζί με ολόκληρο το νοηματικό πλαίσιο εισόδου, για την παραγωγή διανυσμάτων συνθήκης με επίγνωση της ιστορίας. Αυτά στη συνέχεια παρέχονται σε ένα δίκτυο δημιουργίας εικόνων για την παραγωγή της ακολουθίας οπτικοποίησης.

Σε αυτή την εργασία πειραματιζόμαστε με παραλλαγές στο αρχικό δίκτυο με βάση τις πρόσφατες εξελίξεις στους τομείς της δημιουργίας εικόνας υπό συνθήκη και μεταγωγής ακολουθίας σε ακολουθία (εναλλακτικός τρόπος αντιμετώπισης του ίδιου προβλήματος). Δηλαδή χρησιμοποιούμε έναν Transformer Encoder [54] για να κωδικοποιήσουμε το νοηματικό πλαίσιο της ιστορίας και μια ενημερωμένη δομή GAN που βασίζεται στην αρχιτεκτονική του μοντέλου SAGAN [65] για πιο σταθερή εκπαίδευση και βελτιωμένη ποιότητα εικόνας.

Προτείνουμε επίσης δύο νέους μηχανισμούς προσοχής για ακολουθίες εικόνων, εμπνευσμένοι από τον Transformer Decoder, για να βοηθήσει το δίκτυο να μάθει τις εξαρτήσεις χαρακτηριστικών σε όλη την ακολουθία και να βελτιώσει τη συνοχή στο παραγόμενο αποτέλεσμα. Αφού περιγράψουμε το αρχικό πλαίσιο, μεταβάλλουμε έναν αριθμό από αυτές τις παραμέτρους και παρουσιάζουμε τα ευρήματά μας με την ελπίδα να προσεγγίσουμε μια επιτυχημένη αρχιτεκτονική τελευταίας τεχνολογίας για την Οπτικοποίηση Ιστορίας.

Λέξεις-κλειδιά — Γεννητικά Ανταγωνιστικά Δίκτυα, Οπτικοποίηση Ιστορίας, Δημιουργία Εικόνας από Κείμενο, Δημιουργία Βίντεο από Κείμενο, Μεταγωγή ακολουθίας σε ακολουθία, Transformer, Προσοχή

Abstract

Story Visualization is a recently proposed Artificial Intelligence task combining challenges from the fields of computer vision, natural language processing and generative AI. The aim is to create a system that produces a sequence of images given an input "story" consisting of natural language sentences or other information elements. The images should correspond to the sentences one-to-one, be satisfyingly realistic and maintain a sense of consistency and serial progression. This problem is closely related to the fields of Text-to-Image and Text-to-Video generation.

The paper introducing the task [29] also proposes the first architecture tackling it: a Generative Adversarial Network preceded by an RNN-based story encoding scheme. The embedded sentences are successively fed into a stack of recurrent cells along with the entire input context, to produce story-aware conditioning vectors. These vectors are then provided to an image generator network to create the visualization sequence.

In this thesis we experiment with variations on the original network based on recent advances in the fields of conditional image generation and sequence-to-sequence transduction (another way to view the same task). Namely we use a Transformer Encoder [54] to reason about the story context and an updated GAN structure based on SAGAN [65] for stabler training and improved image fidelity.

We also propose two novel attention mechanisms for image sequences inspired by the Transformer Decoder, to help the network learn feature dependencies across the sequence and improve consistency in the generated storyboard. After describing the initial framework we vary a number of these parameters and present our findings in hopes of approaching a successful state-of-the-art Story Visualization architecture.

Keywords — Generative Adversarial Networks, Story Visualization, Text-to-Image Generation, Text-to-Video Generation, Sequence-to-Sequence Transduction, Transformer, Attention

Ευχαριστίες

Σε επίσημο επίπεδο θα ήθελα να ευχαριστήσω τον κύριο Στάμου και τους κυρίους Σιόλα και Αλεξανδρίδη για την πολύτιμη βοήθεια τους στην εκπόνηση αυτής της διπλωματικής, και τη Μαρία Λυμπεραίου για τη στενή συνεργασία μας και την ανεκτίμητη συνεισφορά της καθ' όλη τη διαδικασία. Ευχαριστώ επίσης το GRNET για την παροχή των υπολογιστικών πόρων για την διεξαγωγή των πειραμάτων.

Στα πιο δακρύβρεχτα, θέλω να ευχαριστήσω τη μητέρα και τον πατέρα μου για όλη τους την αγάπη και υποστήριξη σε όλη μου τη σταδιοδρομία. Ευχαριστώ επίσης τους συμμάχους μου στο Παραδοσιακό, για τις καλές στιγμές που ζήσαμε τα χρόνια μας αυτά, καθώς ξεκινήσαμε μαζί και βγήκαμε μαζί αλώβητοι από την άλλη πλευρά. Τέλος θέλω να ευχαριστήσω τους υπόλοιπους φίλους μου, παιδικούς και φοιτητικούς γιατί χωρίς αυτούς δεν θα ήμουν τίποτα απ' όσα είμαι σήμερα.

Τσάκας Νικόλαος, Νοέμβριος 2021

Contents

Contents xiii

List of Figures xv

1 Εκτεταμένη Περίληψη στα Ελληνικά 1
- 1.1 Θεωρητικό Υπόβαθρο . . . 2
  - 1.1.1 Γεννητική Προσέγγιση . . . 2
  - 1.1.2 Γεννητικά μοντέλα υπό συνθήκη . . . 2
  - 1.1.3 Οπτικοποίηση ιστορίας . . . 3
  - 1.1.4 Ακολουθία σε ακολουθία . . . 3
- 1.2 Σχετικές Αρχιτεκτονικές . . . 4
  - 1.2.1 StackGAN . . . 4
  - 1.2.2 StoryGAN . . . 5
  - 1.2.3 SAGAN . . . 7
- 1.3 Οπτικοποίηση Ιστορίας με Προσοχή . . . 8
  - 1.3.1 Γεννήτορας . . . 8
  - 1.3.2 Διευκρινιστής Εικόνας . . . 11
  - 1.3.3 Διευκρινιστής Ιστορίας . . . 11
- 1.4 Πειράματα . . . 11
  - 1.4.1 Πείραμα: Κανόνας ενημέρωσης τριών χρονικών κλιμάκων . . . 11
  - 1.4.2 Πείραμα: Αμερόληπτος Κωδικοποιητής . . . 13
  - 1.4.3 Πείραμα: Προθέρμανση Ρυθμού Εκμάθησης . . . 16
  - 1.4.4 Πείραμα: Υπερπαράμετροι Transformer . . . 16
  - 1.4.5 Πείραμα: Μηχανισμοί Προσοχής . . . 17
- 1.5 Συμπέρασματα και Μελλοντικές κατευθύνσεις . . . 17

2 Introduction 21
- 2.1 Motivation . . . 22
  - 2.1.1 Background . . . 22
  - 2.1.2 Story Visualization . . . 22
  - 2.1.3 Sequence-to-Sequence . . . 23
- 2.2 Contribution . . . 23
- 2.3 Thesis Structure . . . 24

3 Generative Adversarial Networks 25
- 3.1 Original Formulation . . . 26
- 3.2 Conditional GANs . . . 26
- 3.3 DCGAN . . . 26

4 Relevant GAN Architectures 29
- 4.1 StackGAN . . . 30
  - 4.1.1 Conditioning Augmentation . . . 30
  - 4.1.2 Stage-I GAN . . . 30

4.1.3 Stage-II GAN . . . 31
4.2 StoryGAN . . . 31
4.2.1 Story Encoder . . . 32
4.2.2 Context Encoder . . . 32
4.2.3 Image Discriminator . . . 33
4.2.4 Story Discriminator . . . 33
4.2.5 Training . . . 33
4.3 SAGAN . . . 34
4.3.1 Model . . . 34
4.3.2 Stabilization . . . 35

5 The Transformer **37**
5.1 Architecture . . . 37
5.2 Attention . . . 37
5.3 Position-wise Feed-Forward Networks . . . 39
5.4 Positional Encoding . . . 39
5.5 Training . . . 40

6 Attention-based Story Visualization **41**
6.1 Generator . . . 41
6.1.1 Conditioning Augmentation . . . 41
6.1.2 Transformer Encoder . . . 43
6.1.3 Upsampling . . . 43
6.1.4 Spectral Normalization . . . 43
6.1.5 Attention . . . 44
6.2 Image Discriminator . . . 45
6.2.1 Transformer Encoder . . . 45
6.2.2 Downsampling . . . 46
6.2.3 Dropout . . . 46
6.2.4 Attention . . . 46
6.2.5 Output . . . 46
6.3 Story Discriminator . . . 46
6.4 Training . . . 48

7 Experiments **49**
7.1 Dataset . . . 49
7.2 Resources . . . 50
7.3 Experiment: Three Time-scale Update Rule . . . 50
7.4 Experiment: Impartial Transformer Encoder . . . 53
7.5 Experiment: Warmup Scheduler . . . 55
7.6 Experiment: Transformer Hyperparameters . . . 57
7.7 Experiment: Attention Mechanisms . . . 58

8 Conclusion Future Directions **59**

9 Bibliography **61**

List of Figures

1.1.1 Εικόνες που παράγονται από το StyleGANv2 [23] . . . 2
1.1.2 Σύγκριση μοντέλων παραγωγής εικόνας από κείμενο [67] . . . 3
1.2.1 Η αρχιτεκτονική του StackGAN. . . . 4
1.2.2 Η δομή του StoryGAN. . . . 5
1.2.3 Δομή του Story Discriminator . . . 7
1.3.1 Γεννήτορας . . . 9
1.3.2 Διευκρινιστής Εικόνας . . . 10
1.3.3 Διευκρινιστής Ιστορίας . . . 12
1.4.1 (αριστερά) Παραχθείσα ακολουθία (δεξιά) Πραγματική ακολουθία . . . 13
1.4.2 Αύξηση του ρυθμού εκμάθησης του Image Discriminator. . . . 14
1.4.3 Αύξηση του ρυθμού εκμάθησης του Story Discriminator. . . . 14
1.4.4 Η χρήση ενός αμερόληπτου κωδικοποιητή μετασχηματιστή φαίνεται να δίνει τα καλύτερα αποτελέσματα, παρόλο που Οι διαβαθμίσεις από το Story Discriminator βλάπτουν την απόδοση. . . . 15
1.4.5 Ένα άλλο παράδειγμα Αμερόληπτου Κωδικοποιητή που δείχνει τα ίδια συμπεράσματα. . . . 15
1.4.6 Ρυθμός εκμάθησης σε σχέση με τον αριθμό των βημάτων, για τη διάσταση μοντέλου 512. . . . 16
1.4.7 Σύγκριση μεθόδων προγραμματισμού του ρυθμού μάθησης. . . . 17
1.4.8 Σύγκριση παραμέτρων για τον Transformer Encoder. . . . 18
1.4.9 Όλες οι παραλλαγές των μηχανισμών προσοχής οδηγούν σε κατάρρευση συστήματος. . . . 19

2.1.1 Images generated by StyleGANv2 [23] . . . 22
2.1.2 Comparison of Text to Image methods "DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image" [67] . . . 23

3.3.1 The DCGAN generator. . . . 27
3.3.2 Comparison of transpose convolution to separate upsampling and filtering. "Deconvolution and Checkerboard Artifacts" [36] . . . 27

4.1.1 The architecture of StackGAN. . . . 30
4.2.1 The StoryGAN framework. . . . 32
4.2.2 Structure of the Story Discriminator . . . 33

5.1.1 The transformer architecture . . . 38
5.2.1 (left) Scaled Dot-Product Attention. (right) Multi-Head Attention. . . . 39
5.4.1 Positional Encodings as defined in the original transformer. (source: "The Illustrated Transformer", J. Alammar [1]) . . . 40

6.1.1 The generator network. In the embedding stage, the generator utilizes a transformer encoder. The attention block may contain any or all attention mechanisms described in this section, although their position relative to the upsampling blockscould vary. . . . 42
6.2.1 The Image Discriminator. The transformer encoder might be the same or a different one compared to the Generator, which is explored in Section **??**. Beyond that, D_{im} operates on each image individually. The attention block in this case refers to the Intra-image Attention module. . . . 45
6.3.1 The Story Discriminator. . . . 47

7.1.1 Example image from the CLEVR dataset 50
7.3.1 (left) Generated sequence (right) Ground truth Training the Generator with faster LR than both Discriminators causes mode collapse. 51
7.3.2 Increasing the learning rate of the Image Discriminator. 51
7.3.3 Increasing the learning rate of the Story Discriminator. 52
7.4.1 Using an Impartial Transformer Encoder seem to give the best results, even though gradients from the Story Discriminator hurt performance. 53
7.4.2 Another Impartial Encoder example illustrating the same conclusions. 54
7.5.1 Learning rate against number of steps, for model dimension 512. 55
7.5.2 Comparison of regular learning rate decay with the warmup scheduling proposed in the Transformer paper. 56
7.6.1 Comparison of different hyperparameter settings for the Transformer Encoder. 57
7.7.1 All variations of our attention mechanisms lead to mode collapse. 58

Chapter 1

Εκτεταμένη Περίληψη στα Ελληνικά

1.1 Θεωρητικό Υπόβαθρο

Η τεχνητή νοημοσύνη έχει προχωρήσει σημαντικά από την εμφάνιση της υπολογιστικής θεωρίας του Alan Turing [10] και τους πρώτους Turing-πλήρεις τεχνητούς νευρώνες [32]. Αυτό που κάποτε αποτελούσε μια προσπάθεια ενοποίησης των μαθηματικών κάτω από ένα ενιαίο θεωρητικό πλαίσιο έχει μετατραπεί τις τελευταίες δεκαετίες στην επιδίωξη της κατανόησης, μοντελοποίησης και υπέρβασης της ανθρώπινης ικανότητας σε κάθε τομέα. Η μηχανική μάθηση, και τα τελευταία χρόνια η επιτυχία της βαθιάς μάθησης, μας επέτρεψαν να το δημιουργήσουμε συστήματα που επιτυγχάνουν εντυπωσιακά αποτελέσματα ανθρώπινου ή και υπερ-ανθρώπινου επιπέδου στις διεργασίες τις οποίες ο εγκέφαλός μας έχει εξελιχθεί να εκτελεί.

Ένα από τα πιο σημαντικά πεδία έρευνας της τεχνητής νοημοσύνης αποτελεί η όραση υπολογιστών. Η όραση υπολογιστών είναι η μελέτη συστημάτων που αποσκοπούν να ταξινομήσουν και να μοντελοποιήσουν χαρακτηριστικά εικόνας σε διεργασίες που σχετίζονται με την ανθρώπινη οπτική αντίληψη. Διαχρονικα, οι εξελίξεις στον τομέα αυτό οδήγησαν σε πολλές σημαντικές καινοτομίες, όπως υψηλής ακρίβειας ταξινόμηση εικόνων [11] και ανίχνευση αντικειμένων[61], καθώς και σημαντικές προόδους στον τομέα της τεχνητής νοημοσύνης υγείας, που υποβοηθά την ιατρική διάγνωση και θεραπεία [66, 49, 2].

1.1.1 Γεννητική Προσέγγιση

Ενώ αυτή η ταξινομητική πλευρά της στατιστικής μάθησης είδε την εμφάνιση πολλών επιτυχημένων μοντέλων τις τελευταίες δεκαετίες, δεν μπορεί να ειπωθεί το ίδιο για τη γενικά πιο δύσκολη γεννητική προσέγγιση. Η μοντελοποίηση σύνθετων κατανομών δεδομένων, όπως είναι η φυσική γλώσσα ή οι ρεαλιστικές εικόνες, ιδιαίτερα σε βαθμό υλοποίησης ενός συστήματος ικανού να παράγει νέα δείγματα που φαίνονται πειστικά σε έναν άνθρωπο παρατηρητή παρουσιάζει μια σειρά από προκλήσεις, οι οποίες επίσης είναι μεταβλητής δυσκολίας ανάλογα με τον τύπο των δεδομένων. Καθώς η απόδοση του υλικού υπολογιστών (hardware) βελτιώνεται συνεχώς και τα νευρωνικά δίκτυα γίνονται πιο βαθιά και πιο περίπλοκα, η προσεγγιση μιας συχνά πολυτροπικής κατανομής σε αυτούς τους χώρους δεδομένων υψηλών διαστάσεων έχει εξελιχθεί σε ερευνητική περιοχή αυξανόμενου ενδιαφέροντος.

Η πρόοδος προ των τελευταίων ετών ήταν αργή, με δίκτυα όπως τα Restricted Boltzmann Machines [48] [14, §20.2] να αποδεικνύονται ασταθή στην εκπαίδευση, μη πρακτικά στη χρήση και ανεπαρκή σε αποτελέσματα. Η εμφάνιση των Variational Autoencoders (VAEs) [24] και των Generative Adversarial Networks [15] έχει επιφέρει μεγάλη βελτίωση σε διεργασίες όπως η παραγωγή εικόνων άνευ συνθήκης [22, 23] και υπό συνθήκη [57, 58], καθώς και η παραγωγή βίντεο [45].

Figure 1.1.1: Εικόνες που παράγονται από το StyleGANv2 [23]

1.1.2 Γεννητικά μοντέλα υπό συνθήκη

Στην περίπτωση σύνθεσης δεδομένων υπό συνθήκη, έχει καταβληθεί μεγάλη προσπάθεια για το σχεδιασμό μοντέλων για τη δημιουργία οπτικών δεδομένων παραγόμενων από κείμενο. Ένας αριθμός εργασιών που παρουσιάζουν εντυπωσιακά αποτελέσματα έχουν διακλαδιστεί σε διαφορετικές παραλλαγές, μαρτυρώντας μια μακρά πορεία αρχιτεκτονικής εξερεύνησης. Οι Reed και συνεργάτες [42, 43] παρουσίασαν μια πρώιμη αρχιτεκτονική που επηρέασε πολλά μεταγενέστερα σχέδια, χρησιμοποιώντας έναν νέο κωδικοποιητή RNN-CNN για το κείμενο και παραγοντας άμεσα εικόνες στην τελική τους ανάλυση. Η σταδιακή κλιμάκωση των χαρακτηριστικών όπως γίνεται απο το StackGAN [63] που παρουσίασαν ο Zhang και οι συνεργάτες του, ωοτόκησαν επίσης πολλαπλούς

επιτυχημένους απόγονους [64, 62, 67]. Υπήρξαν ακόμη και ορισμένα εγχειρήματα προς τη δημιουργία βίντεο από κείμενο [28], αν και ο τομέας της παραγωγής βίντεο είναι ακόμα σε πρώιμο στάδιο και πολύ πιο απαιτητικός, επομένως δεν έχουν ακόμη επιτευχθεί αξιοσημείωτα αποτελέσματα.

Figure 1.1.2: Σύγκριση μοντέλων παραγωγής εικόνας από κείμενο [67]

1.1.3 Οπτικοποίηση ιστορίας

Τα δύο προαναφερθέντα θέματα εμπνέουν ένα φυσικό ενδιάμεσο στο νέο τομέα της Οπτικοποίησης ιστορίας (Story Visualization - SV), ο οποίος περιγράφεται από τους Li et al. [29] ως η δημιουργία μιας ακολουθίας εικόνων που βασίζεται σε μια σύντομη ιστορία με προτάσεις φυσικής γλώσσας ή άλλες σημασιολογικές πληροφορίες. Η εργασία δανείζεται από τον τομέα της σύνθεσης εικόνας από κείμενο στην επιδίωξη της αντιστοιχίας κειμένου-εικόνας, καθώς και της σύνθεσης βίντεο από κείμενο λόγω της επιζήτησης συνέπειας μεταξύ των καρέ. Επί του παρόντος, λίγες βελτιώσεις έχουν προταθεί για αυτό το δύσκολο θέμα [27] και υπάρχει έλλειψη κατάλληλων συνόλων δεδομένων και μεθόδων αξιολόγησης.

1.1.4 Ακολουθία σε ακολουθία

Ένας άλλος τρόπος για να δει κανεις την οπτικοποίηση ιστορίας είναι ως πρόβλημα μεταγωγής ακολουθίας σε ακολουθία, όμοια με την αυτόματη μετάφραση. Τα μοντέλα μεταγωγής ακολουθίας σε ακολουθία είναι ένας τομέας μελέτης που που καλύπτεται κυρίως από την Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing - NLP), με την εστίαση να κινείται σταδιακά από τα αναδρομικά νευρωνικά δίκτυα [51, 8] προς μοντέλα που βασίζονται σε μηχανισμούς προσοχής [5]. Αυτή η τάση κορυφώθηκε με τον Transformer [54], ένα σημαντικό δίκτυο που εκτελεί εργασίες NLP χρησιμοποιώντας αποκλειστικά μηχανισμούς προσοχής. Από την εμφάνισή του, ο Transformer έχει προτιμηθεί στην έρευνα για την απλή του προσέγγιση, αποδοτική εκπαίδευση και εντυπωσιακά αποτελέσματα. Πολλά διαδεδομένα μοντέλα έχουν επεκταθεί πάνω στον αρχικό Transformer για αυτόματη μετάφραση [59] και μοντελοποίηση γλώσσας [12, 31, 41], συμπεριλαμβανομένου του GPT-3 [6], ενός μοντέλου φυσικής γλώσσας ικανό να εκτελέσει μία ποικιλία εργασιών σε σχεδόν ανθρώπινο επίπεδο. Transformers και άλλοι μηχανισμοί προσοχής έχουν επίσης χρησιμοποιηθεί σε εφαρμογές όρασης υπολογιστών όπου είναι ικανοί να το βελτιώσουν υπάρχουσες προσεγγίσεις, μαθαίνοντας πολύπλοκες εξαρτήσεις που οι συνήθεις μέθοδοι που βασίζονται στη συνέλιξη δεν καταφέρνουν να αποτυπώσουν [39, 65].

Είναι ο συνδυασμός αυτών των πρόσφατων προόδων που ενέπνευσαν την προσέγγισή μας στο έργο της οπτικοποίησης ιστορίας, με την ελπίδα να συνεισφέρουμε σε ένα μοντέλο που μπορεί να αποτυπώσει τις την παραγωγή ακολουθιών εικόνων και την αντιστοίχιση γλώσσικής-οπτικής πληροφορίας.

1.2 Σχετικές Αρχιτεκτονικές

Σε αυτό το κεφάλαιο εξετάζουμε τρεις αρχιτεκτονικές GAN που θεμελιώνουν τη συνεισφορά μας. Αυτά τα μοντέλα έχουν σχεδιαστεί για διαφορετικές εφαρμογές παραγωγής εικόνας υπό συνθήκη, αλλά συνολικά παρέχουν τα βασικά στοιχεία για την κατανόηση του κεφαλαίου 6, όπου παρουσιάζουμε τις ιδέες μας για ένα σύγχρονο GAN οπτικοποίησης ιστορίας που βασίζεται στην προσοχή.

Το StackGAN είναι ένα σημαντικό ορόσημο για τη δημιουργία εικόνας από κείμενο που έχει επηρεάσει πολλές εργασίες για το θέμα τόσο από άποψη δομής όσο και εκπαιδευτικής προσέγγισης. Έτσι, το θεωρούμε φυσικό προκάτοχο του StoryGAN, του πρωτότυπου μοντέλο οπτικοποίησης ιστορίας, που έχει δανειστεί σε μεγάλο βαθμό από αυτό. Επιπλέον το Self-Attention GAN (SAGAN) παρουσιάζεται ως ένα πρόσφατο μοντέλο παραγωγής εικόνων υπό συνθήκη που χρησιμοποιεί νεότερα αρχιτεκτονικά χαρακτηριστικά και μηχανισμούς σταθεροποίησης της εκπαίδευσης, που υιοθετούνται ευρέως από νευρωνικά δίκτυα τα τελευταία χρόνια. Επίσης χρησιμοποιεί έναν μηχανισμό προσοχής που ενέπνευσε σε μεγάλο βαθμό τους δικούς μας καινοτόμους μηχανισμούς για την εκμάθηση χαρακτηριστικών σε ακολουθίες εικόνων.

1.2.1 StackGAN

Προηγούμενες εργασίες για τη δημιουργία εικόνων με βάση το κείμενο [42, 43] προσπάθησαν να παράξουν εικόνες πλήρους ανάλυσης απευθείας, με αποτέλσεμα τα δείγματα να υπολείπονται σε πειστικότητα και λεπτομέρεια. Το StackGAN, που προτάθηκε από τους Zhang et al. [63] πέτυχε τα πρώτα σημαντικά βελτιωμένα αποτελέσματα στην παραγωγή εικόνας από κείμενο, δημιουργώντας σταδιακά την τελική εικόνα με πολλαπλά στάδια ανταγωνιστικής εκπαίδευσης. Αρχικά, δημιουργείται μια εικόνα χαμηλής ανάλυσης 64x64 (Stage-I GAN), αποτυπώνοντας τα ευρύτερα χαρακτηριστικά της εικόνας με βάση το κείμενο και στη συνέχεια ενας δεύτερος γεννήτορας (Stage-II GAN) βασιζεται στην πρώτη εικόνα για να δημιουργήσει αποτέλεσμα υψηλής ανάλυσης 256x256. Διευκρινιστές εκπαιδεύονται και στα δύο στάδια. Η ίδια εργασία πρότεινε έναν νέο τρόπο επαύξησης των δεδομένων εκπαίδευσης, που ονομάζεται Επαύξηση Συνθήκης.

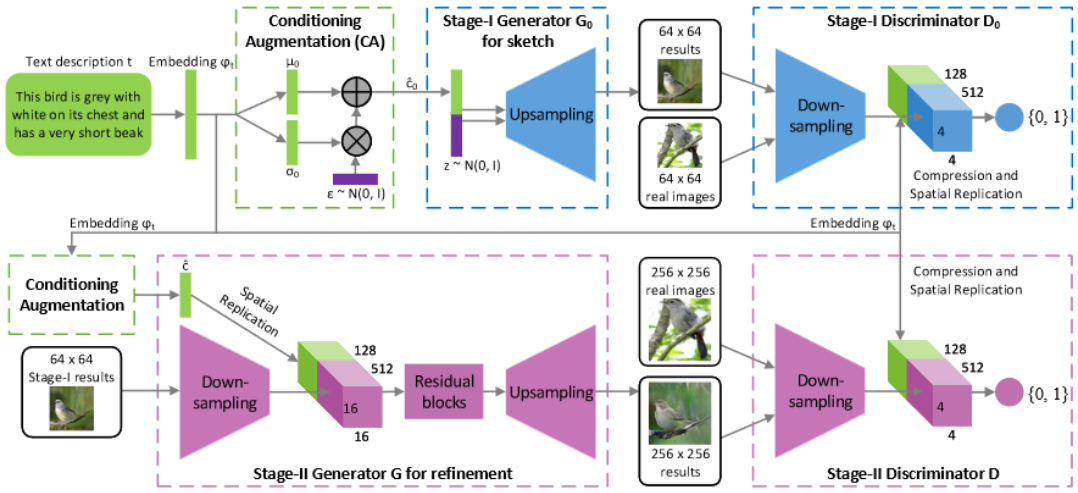


Figure 1.2.1: Η αρχιτεκτονική του StackGAN.

Επαύξηση Συνθήκης

Τα σύνολα δεδομένων που χρησιμοποιούνται για τη δημιουργία εικόνας από κείμενο έχουν από τη φύση τους προβληματικά χαρακτηριστικά που επηρεάζουν αρνητικά την εκπαίδευση. Οι περιγραφές αναπόφευκτα παράγουν

ένα πολύ αραιό σύνολο σε ένα χώρο χαρακτηριστικών υψηλών διαστάσεων, καθιστώντας την προσέγγιση μιας κατανομής πιθανοτήτων στο εν λόγω σύνολο δύσκολη. Για να γίνει ο χώρος ορισμού πιο συνεχής και να μάθει το δίκτυο να είναι αδρανές σε μικρές διακυμάνσεις θέσης στον χώρο αναπαράστασης, προτείνεται ένας νέος τρόπος αύξησης των δεδομένων:

Αντί να εκπαιδευτει το GAN σε μια αναπαράσταση του κειμένου ϕ_t, λαμβάνεται ένα τυχαίο δειγμα $\hat{c}$ από μία πολυδιάστατη γκαουσιανή κατανομή $\mathcal{N}(\mu(\varphi_t, \Sigma(\varphi_t)))$ με μέσο όρο $\mu(\varphi_t)$ και ο διαγώνιο πίνακα συνδιακύμανσης $\Sigma(\varphi_t)$, αμφότερα συναρτήσεις των αναπαραστάσεων του κειμένου. Το διάνυσμα $\hat{c}$ χρησιμεύει ως μεταβλητή συνθήκης, ενώ οι ίδιες οι συναρτήσεις υλοποιούνται ως νευρωνικά δίκτυα με μεταβλητές παραμέτρους, εκπαιδευμένα παράλληλα με το υπόλοιπο StackGAN. Για να εξασφαλιστεί η ομαλότητα της αναπαράστασης, η απόκλιση Kullback-Leibler μεταξύ της προσεγγιζόμενης Gaussian κατανομής και της κανονικής προστίθεται στη συνάρτηση απώλειας του γεννήτορα ως όρος ομαλοποίησης:

$$D_{KL}(\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))\|\mathcal{N}(0, I))$$

Αυτός ο όρος βοηθά στην αποφυγή υπερπροσαρμογής διαμέσου εκμάθησης μιας "συμπτυγμένης" σημειακής κατανομής ή μιας κατανομής που αποκλίνει πάρα πολύ από την κανονική.

1.2.2 StoryGAN

Οι Li et al. εισήγαγαν το πρόβλημα της οπτικοποίσης ιστορίας [29] ως φυσικό διάμεσο μεταξύ των εργασιών της δημιουργίας εικόνας από κείμενο και βίντεο από κείμενο. Ο σκοπός είναι να δημιουργήθεί μια ακολουθία εικόνων που εξαρτώνται από ένα σύνολο προτάσεων που σχηματίζουν μία συνεκτική ιστορία. Η βασική διεργασία ξεπερνά την απλή διαδοχική εφαρμογή ενός μοντέλου κειμένου-εικόνας, αφού οι παραγόμενες εικόνες χρειάζεται να διατηρούν μια αίσθηση οπτικής και εννοιολογικής συνέπειας και προόδου. Ένας γεννήτορας που δεν γνωρίζει το πλαίσιο στο οποίο ανήκει μια εικόνα θα αποτύχει, δίνοντας ένα μη συνεκτικό αποτέλεσμα.

Για το σκοπό αυτό εισήχθη το StoryGAN, ένα γεννητικό ανταγωνιστικό μοντέλο που μπορεί να παράγει αλληλουχίες εικόνας απο διαδοχές προτάσεων. Το δίκτυο χρησιμοποιεί μια δομή αναδρομικών νευρωνικών δικτύων (RNN) που εμποτίζει τις αναπαραστάσεις των προτάσεων με πληροφορίες από τη συνολική ιστορία, καθοδηγώντας τη δημιουργία μιας εικόνας από έναν υπό συνθήκη γεννήτορα εικόνας παρόμοιο στη δομή με το StackGAN και άλλες αρχιτεκτονικές σύνθεσης εικόνας από κείμενο [42]. Ο γεννήτορας G εκπαιδεύεται ανταγωνιστικά με δύο διευκρινιστές. Ο διευκρινιστής εικόνας D_{im} έχει σκοπό να αξιολογήσει πόσο γνήσια φαίνεται η εικόνα σε σύγκριση με τα πραγματικά δεδομένα και πόσο ανταποκρίνεται στην πρόταση, ενώ ο διευκρινιστής ιστορίας D_{st} εκπαιδεύεται για να διασφαλίζει τη συνέπεια μεταξύ των εικόνων, δεδομένου του νοηματικού πλαισίου.

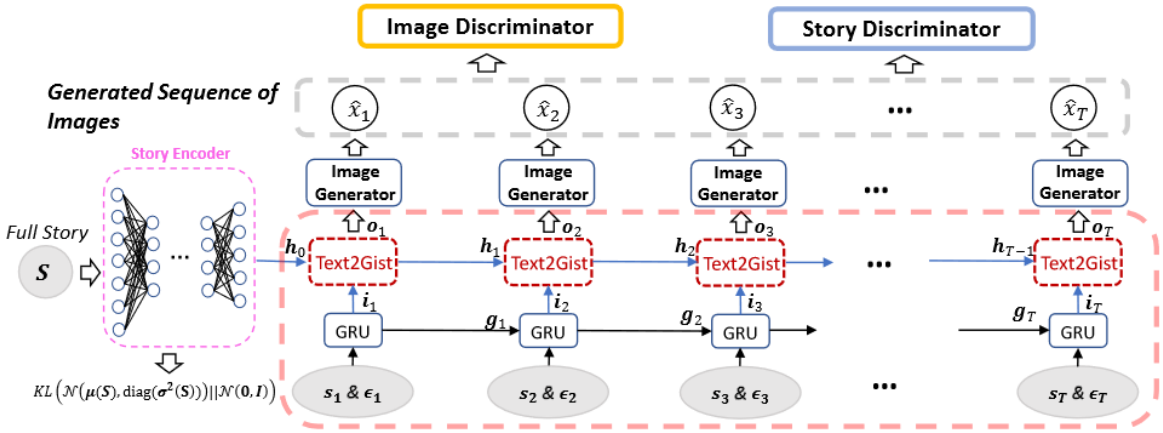


Figure 1.2.2: Η δομή του StoryGAN.

Κωδικοποιητής ιστορίας

Προκειμένου να αντιμετωπιστεί η ασυνέχεια του πολυτόπου δεδομένων, οι Zhang et al προτείνουν το μηχανισμό Επάυξησης Συνθήκης. Το [63] χρησιμοποιείται για την κωδικοποίηση της ιστορίας. Ολόκληρη η ιστορία απεικονίζεται σε ένα χαμηλής διαστασης διάνυσμα h_0 με δειγματοληψία μιας κατανομής Gauss $h_0 \sim \mathcal{N}(\mu(S), \Sigma(S))$ όπου το μ και το Σ είναι συναρτήσεις της ιστορίας S. Αυτό το διάνυσμα παρέχεται ως η αρχική κρυμμένη κατάσταση στον κωδικοποιητή νοήματος RNN που περιγράφεται παρακάτω.

Ο ίδιος όρος κανονικοποίησης προστίθεται στην απώλεια γεννήτριας:

$$\mathcal{L}_{KL} = D_{KL}(\mathcal{N}(\mu(S), \Sigma(S)) \| \mathcal{N}(0, I))$$

Κωδικοποιητής νοήματος

Για την παραγωγή του διανύσματος συνθήκης για κάθε παραγόμενη εικόνα χρησιμοποιείται μια στοιβαγμένη δομή RNN. Το κατώτερο επίπεδο του RNN χρησιμοποιεί τυπικές μονάδες GRU [9], ενώ το ανώτερο χρησιμοποιεί μια παραλλαγή μονάδων GRU που προτείνεται από τους Li et al. ονομαζόμενες Text2Gist. Αυτό το δεύτερο στρώμα είναι και αυτό του οποίου η κρυφή κατάσταση αρχικοποιείται το διάνυσμα h_0. Για κάθε βήμα t στην ακολουθία το επίπεδο GRU λαμβάνει ισομετρικό θόρυβος ϵ_t μαζί με την πρόταση s_t και η έξοδος τροφοδοτείται στο επίπεδο Text2Gist που τη συνδυάζει με πληροφορίες που προέρχονται από το πλαίσιο της ιστορίας. Η τελική έξοδος o_t είναι το διάνυσμα που ελέγχει τη δημιουργία εικόνας. Αν g_t, h_t είναι οι κρυφές καταστάσεις του GRU και του Text2Gist κελιά αντίστοιχα, το στοιβαγμένο RNN είναι δομημένο ως:

$$\begin{aligned} i_t, g_t &= GRU(s_t, \epsilon_t, g_{t-1}) \\ o_t, h_t &= Text2Gist(i_t, h_{t-1}) \end{aligned}$$

Ο ορισμός των προτεινόμενων κελιών Text2Gist έχει ως εξής:

$$\begin{aligned} z_t &= \sigma_z(W_z i_t + U_z h_{t-1} + b_z) \\ r_t &= \sigma_r(W_r i_t + U_r h_{t-1} + b_r) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \sigma_h(W_h i_t + U_h(r_t \odot h_{t-1} + b_h)) \\ o_t &= Filter(i_t) * h_t \end{aligned}$$

Το $Filter(.)$ είναι μια συνάρτηση που αντιστοιχίζει το διάνυσμα i_t σε ένα πολυκαναλικό φίλτρο που χρησιμοποιείται για 1x1 συνέλιξη με την κατάσταση h_t, ωστε να συνδυάσει τοπικές και περιφερειακές πληροφορίες πιο αποτελεσματικά στη διαδικασία δημιουργίας ενός διανύσματος συνθήκης.

Διευκρινιστής Εικόνας

Ο διευκρινιστής εικόνας D_{im} του StoryGAN λειτουργεί πολύ παρόμοια με αυτόν που προτείνεται στην εργασία του StackGAN. Η μόνη διαφορά είναι οτι μαζί με την προτεινόμενη εικόνα και το αντίστοιχο διάνυσμα αναπαράστασης του κειμένου, λαμβάνει επιπλέον και ολοκληρο το πλαίσιο της ιστορίας καθώς είναι απαραίτητο για την παραγωγή της τελικής εικόνας.

Διευκρινιστής Ιστορίας

Ο Διευκρινιστής Ιστορίας αναπαριστά τόσο την ιστορία όσο και την παραγόμενη ακολουθία εικόνων σε έναν κοινό χώρο προκειμένου να υπολογιστεί μια βαθμολογία ομοιότητας μεταξύ τους.

Ένας κωδικοποιητής εικόνας παράγει μια σειρά από διανύσματα χαρακτηριστικών $E_{img}(\mathbf{X}) = [E_{img}(x_1), \ldots, E_{img}(x_T)]$ από μια εικόνα εισόδου $\mathbf{X}$ που συνενώνονται σε ένα ενιαίο διάνυσμα ενώ είναι ένας κωδικοποιητής κειμένου κάνει το ίδιο για όλες τις προτάσεις στην ιστορία $\mathbf{S}$, δημιουργώντας μια σειρά χαρακτηριστικών κειμένου $E_{txt}(\mathbf{X}) = [E_{txt}(s_1), \ldots, E_{txt}(s_T)]$, που επίσης συνενώνονται. Τα τελικά μεγάλα διανύσματα πολλαπλασιάζονται στοιχείο προς στοιχείο και τροφοδοτούνται σε ένα γραμμικό μετασχηματισμό, ισοδύναμος με ένα πλήρως συνδεδεμένο στρώμα με σιγμοειδή ενεργοποίση:

$$D_{st}(\mathbf{X}, \mathbf{S}) = \sigma(w^T(E_{img}(\mathbf{X}) \odot E_{txt}(\mathbf{S})) + \beta)$$

όπου $D_{st}(\mathbf{X}, \mathbf{S})$ είναι η τελική βαθμολογία ομοιότητας που κανονικοποιείται σε $[0, 1]$.

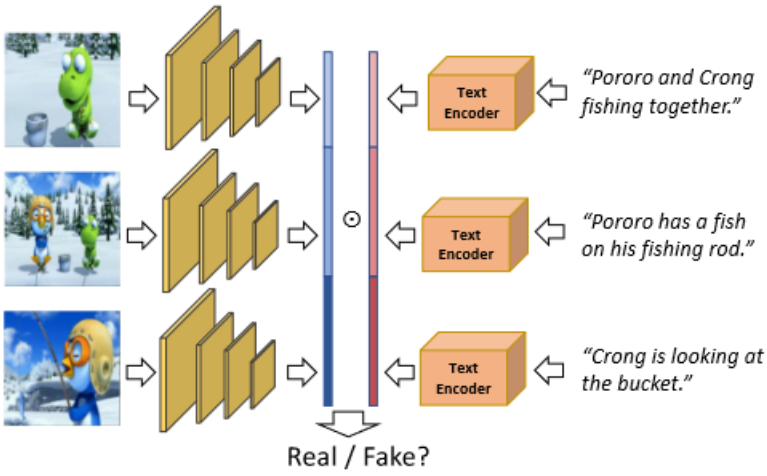


Figure 1.2.3: Δομή του Story Discriminator

1.2.3 SAGAN

Τα πρόσφατα υπο συνθήκη GAN για εικόνες [37, 34] φαίνεται να έχουν μεγαλύτερη επιτυχία σε εικόνες όπου η υφή και το χρώμα είναι τα πιο σημαντικά χαρακτηριστικά, αλλά δυσκολεύονται με τη δομή των αντικειμένων και άλλες μακρινές εξαρτήσεις. Το πρόβλημα μπορεί να αποδοθεί στον έντονα τοπικό χαρακτήρα των συνελικτικών φίλτρων, που είναι ο κύριος τύπος στρώματος που χρησιμοποιείται για τη δημιουργία εικόνων υψηλής ανάλυσης, αφού συνήθως προτιμώνται λόγω της υπολογιστικής τους αποδοτικότητας.

Το Self-Attention GAN [65] είναι ένα μοντέλο που προτείνεται από τους Zhang et al. για να αντιμετωπιστεί αυτό το αποτέλεσμα, με κίνητρο το πρόσφατο κύμα δημοφιλίας των μοντέλων προσοχής στην επεξεργασία φυσικής γλώσσας [5, 54] καθώς και την παραγωγή εικόνας [62, 39]. Παράλληλα με τον προτεινόμενο μηχανισμό αυτο-προσοχής που εισάγονται τόσο στο γεννήτορα όσο και στο διευκρινιστή, οι συγγραφείς υποστηρίζουν τη χρήση συγχρονων τεχνικών σταθεροποίσης για την εκπαίδευση , όπως Φασματική Κανονικοποίηση βαρών [35] και τον Κανόνα Ενημέρωσης Δύο Χρονικών Κλιμάκων [18].

Μοντέλο

Το μοντέλο ακολουθεί την τυπική δομή ενός γεννήτορα που αυξάνει σταδιακα την διασταση των χαρακτηριστικών και ενός διευκρινστή που τη μειώνει, όπως περιγράφεται και στις παραπάνω αρχιτεκτονικές. Στο γεννήτορα, οι ετικέτες δεδομένων διανυσματοπούνται, και αυξάνονται σε ανάλυση για την παραγωγή της εικόνας εξόδου. Ο διευκρινστής λαμβάνει μια προτεινόμενη εικόνα και το αντίστοιχο διάνυσμα ετικέτας, και παράγει ένα βαθμωτό αποτέλεσμα σαν έξοδο, την πιθανότητα το ζεύγος εικόνας-κειμένου να προέρχεται από το σύνολο δεδομένων. Η αύξηση και μείωση της διάστασης εκτελούνται από Υπολειπτικά Μπλοκ [17] που μετασχηματίζουν τα ενδιάμεσα χαρακτηριστικά της εικόνας με συνελικτικά στρώματα. Το πρωτότυπο αρχιτεκτονικό τμήμα του μοντέλου είναι η μονάδα αυτο-προσοχής που εισάγεται μία φορά μεταξύ των προαναφερθέντων μπλοκ ανασχηματισμού σε κάθε δίκτυο.

Η ενότητα αυτο-προσοχής, εμπνευσμένη από το μη τοπικό μοντέλο που εισήχθη από τους Wang et al. [56], λειτουργεί με γραμμική αντιστοίχιση χαρακτηριστικών εικόνας (κανάλια) στα διανύσματα κλειδιού, τιμής και ερωτήματος (χρησιμοποιώντας την ορολογία του Transformer [54]).

Σταθεροποίηση εκπαίδευσης

Το SAGAN περιορίζει τη σταθερά Lipschitz του διευκρινστή [3] μέσω Φασματικής Κανονικοποίησης [35], μιας τεχνικής που αποδεδειγμένα είναι αποτελεσματική στη σταθεροποίηση της εκπαίδευσης, ενώ παραμένει υπολογιστικά αποτελεσματική. Εμπνευσμένο από τους Odena et al. [38] στην έρευνά τους σχετικά με τη σημασία της καλής προσαρμογής του γεννήτορα για σταθερή εκπαίδευση, η φασματική κανονικοποίηση χρησιμοποιείται περαιτέρω στο γεννήτορα για βελτίωση της σύγκλισης του μοντέλου. Η χρήση της φασματικής Κανονικοποίησης

και για τους δύο αντιπάλους μειώνει επίσης την ανάγκη για πολλαπλές ενημερώσεις του διευκρινιστή ανά επανάληψη, το οποίο είναι μια συνήθης τεχνική για τη βελτίωση της ποιότητας του παραγόμενου δείγματος στα GANs.

Χρησιμοποιούνται επίσης ξεχωριστοί ρυθμοί εκμάθησης για το γεννήτορα και το διευκρινστή (Κανόνας Ενημέρωσης Δύο Χρονικών Κλιμάκων / Two Time-scale Update Rule - TTUR), μια τεχνική που προτείνουν οι Heusel et al. [18] για την καταπολέμηση της ανισσοροπίας μάθησης που εμφανίζεται συχνά στην εκπαίδευση GAN μεταξύ των δύο δικτύων. Είναι σύνηθες να βλέπουμε αποτελέσματα χαμηλής ποιότητας από το γεννήτορα να είναι επαρκή για να ξεγελάσουν νωρίς στην εκπαίδευση το διευκρινιστή, που σημαίνει ότι το μοντέλο δεν συγκλίνει. Το TTUR επιτρέπει σε έναν καλά σχεδιασμένο διαχωριστή να κινείται προς ένα βέλτιστο σημείο γρηγορότερα, οδηγώντας το γεννήτορα να παράγει καλύτερα δείγματα. Αυτή η προσέγγιση προτιμάται απο τη χρήση διαφορετικού αριθμού βημάτων εκπαίδευσης ανά εποχή για κάθε δίκτυο, καθώς είναι πιο αποδοτική από άποψη χρόνου εκτέλεσης.

1.3 Οπτικοποίηση Ιστορίας με Προσοχή

Σε αυτή την εργασία προτείνουμε ένα ενημερωμένο πλαίσιο για το πρόβλημα της Οπτικοποίησης Ιστορίας με βάση την εμφάνιση τεχνικών προσοχής για την επεξεργασία ακολουθιών και καινοτομίες στο συναφές πρόβλημα της δημιουργίας εικόνων. Αρχικά, συνιστούμε τη χρήση ενός Transformer Encoder [54] ως αντικατάσταση της δομής RNN που προτείνεται από τους Li et al. για το StoryGAN [29], για να κωδικοποιήσει το πλαίσιο της ιστορίας στο διάνυσμα συνθήκης για κάθε παραγόμενη εικόνα.

Ως δεύτερη συνεισφορά, συνιστούμε τη χρήση ενός δικτύου παρόμοιο με το SAGAN για ανταγωνιστική μάθηση, και πειραματιζόμαστε με τη χρήση πρόσθετων μηχανισμών προσοχής για την ενίσχυση της ακολουθιακής συνέπειας και προόδου μεταξύ των χαρακτηριστικών των εικόνων που δημιουργούνται. Εκτός από την προσοχή, που προτείνεται στο SAGAN [65] (όπου οι τοποθεσίες της εικόνας συντίθενται με παρακολούθηση άλλων τοποθεσιών στην ίδια εικόνα), διερευνούμε επίσης την αποτελεσματικότητα των μηχανισμών προσοχής μεταξύ των εικόνων της ακολουθίας, παρόμοια με τον Transformer Decoder. Αναφέρουμε επίσης λεπτομερώς τα αποτελέσματα των πειραματισμών μας με τα διάφορα τμήματα και τις παραμέτρους αυτού του πλαισίου.

1.3.1 Γεννήτορας

Η είσοδος στο γεννήτορα G που φαίνεται στο σχήμα 1.3.1 είναι μια ακολουθία συμβόλων s_t, πιθανώς κωδικοποιήμένα από έναν κατάλληλο κωδικοποιητή (όπως ο Universal Sentence Encoder [7], στην περίπτωση προτάσεων φυσικής γλώσσας) σε διανυσματικές αναπαραστάσεις φ_t, $t \in [1, T]$ όπου το T είναι το μήκος όλων των ιστοριών στο σύνολο δεδομένων, και υπερπαράμετρος του μοντέλου.

Το δίκτυο δομείται με τα εξής χαρακτηριστικά:

- Διανυσματική Επάυξηση για όλα τα διανύσματα εισόδου.
- Χρήση ενός κωδικοποιητή Transformer Encoder για την ενσωμάτωση σημασιολογικών πληροφοριών από την ιστορία σε κάθε διάνυσμα.
- Υπερδειγματοληψία και μετασχηματισμός των χαρακτηριστικών της εικόνας με τη χρήση υπολειπτικών μπλοκ.
- Πιθανή παρεμβολή των εξής μηχανισμών προσοχής:
    1. Ενδο-προσοχή, δηλαδή σε κάθε εικόνα ο μηχανισμός προσοχής όπως προτείνεται για το SAGAN.
    2. Δια-προσοχή: Όπου τα κανάλια των εικόνων της ακολουθίας συμπεριφέρονται ως κεφαλές στην Πολυκεφαλική Προσοχή (Multi-head Attention) του Transformer, και τα κλειδιά, οι τιμές και οι ερωτήσεις όλα προερχονται από τις εικόνες.
    3. Προσοχή Κωδικοποιητή-Γεννήτορα: Όμοια με την άνω, αλλά οι ερωτήσεις προέρχονται από τα διανύσματα συνθήκης, όμοια με την προσοχή Κωδικοποιητή-Αποκωδικοποιητή στον Transformer.

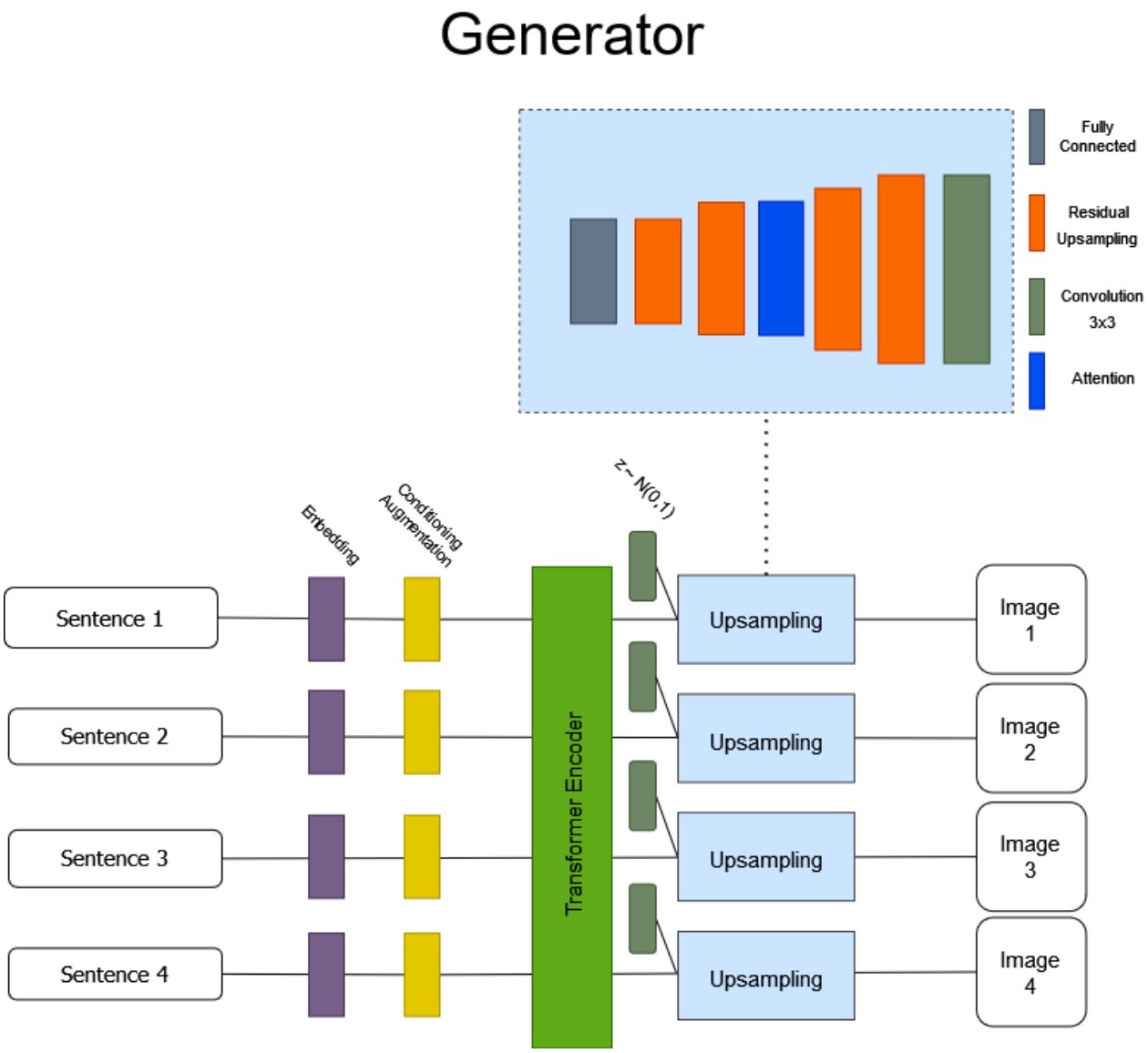


Figure 1.3.1: Γεννήτορας

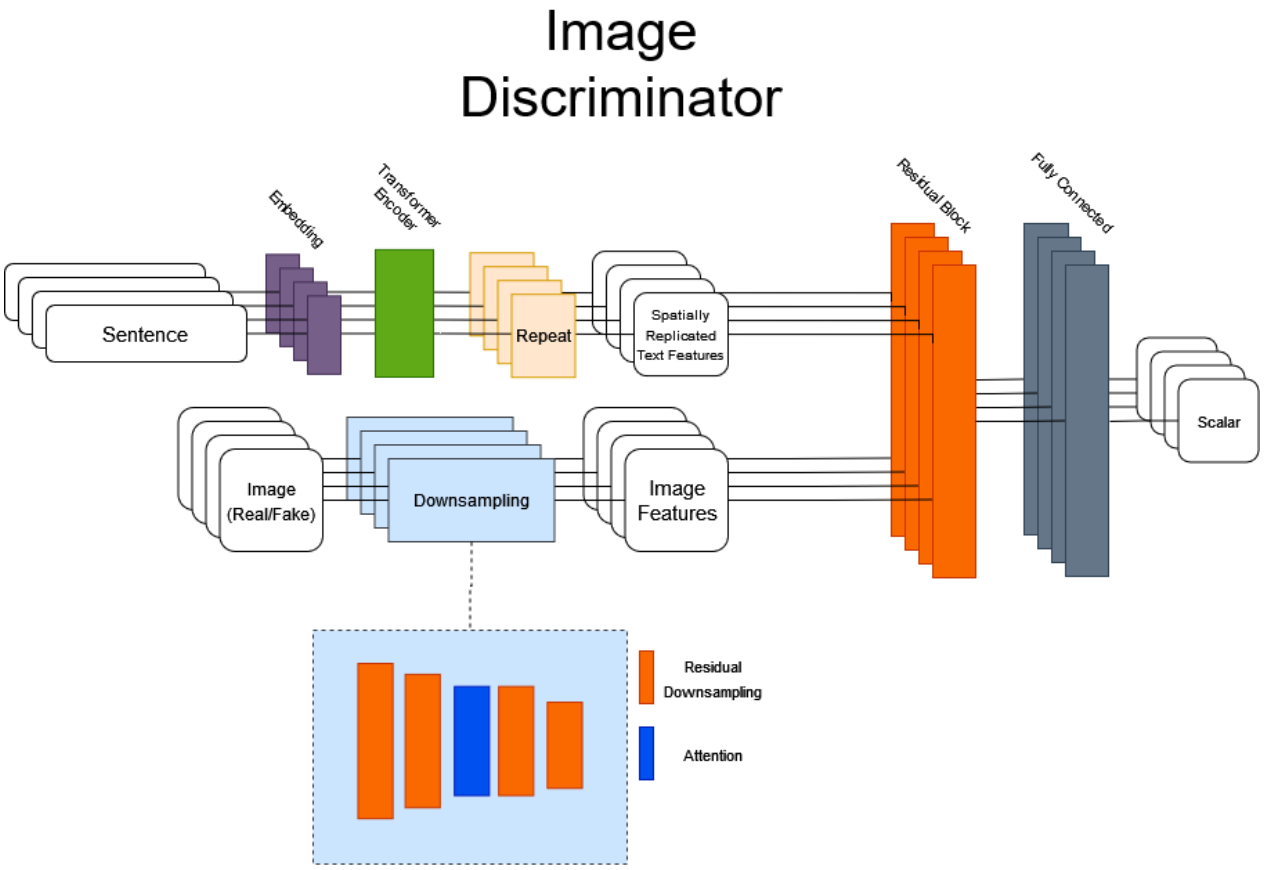


Figure 1.3.2: Διευκρινιστής Εικόνας

1.3.2 Διευκρινιστής Εικόνας

Ο σκοπός του διευκρινιστή εικόνας D_{im} (Εικόνα 1.3.2) είναι να διακρίνει μεταξύ εικόνων από το σύνολο δεδομένων και τεχνητών εικόνων. Για το σκοπό αυτό, χρησιμοποιεί τα χαρακτηριστικά του κειμένου φ_t της αντίστοιχης πρότασης της ιστορίας, το πλαίσιο (δηλ. τις άλλες προτάσεις της ιστορίας) και την εικόνα I_t προς αξιολόγηση. Το πλαίσιο είναι σημαντικό για τον διευκρινιστή, επειδή κάθε εικόνα που αντιστοιχεί σε μία πρόταση στην ιστορία εξαρτάται από τις υπόλοιπες για να σχηματίσει πολλές από τις λεπτομέρειες. Αναλογιστείτε τα ακόλουθα παραδείγματα:

1. "Προσθέστε έναν κόκκινο μεταλλικό κύβο. Στη συνέχεια προσθέστε έναν κίτρινο κύλινδρο."
2. "Μια σιλουέτα φαινόταν έξω από το παράθυρο. Ήταν μια μαύρη γάτα."

Στην πρώτη περίπτωση, η δεύτερη εικόνα εξαρτάται από το νοηματικό πλαίσιο στα αριστερά, ενώ στη δεύτερη περίπτωση, η πρώτη εικόνα πρέπει να έχει επίγνωση του πλαισίου προς τα δεξιά.

Ο διευκρινιστής εικόνας προορίζεται να ταξινομήσει κάθε εικόνα ξεχωριστά, όχι ως μέρος της ακολουθίας στην οποία ανήκει. Ωστόσο, όλες οι παραγόμενες εικόνες μιας ιστορίας αξιολογούνται παράλληλα, για να επωφεληθούμε από την παράλληλη φύση του Transformer.

Το δίκτυο δομείται με τα εξής χαρακτηριστικά:

- Χρήση ενός κωδικοποιητή Transformer Encoder για την ενσωμάτωση σημασιολογικών πληροφοριών από την ιστορία σε κάθε διάνυσμα.
- Υποδειγματοληψία και μετασχηματισμός των χαρακτηριστικών της εικόνας με τη χρήση υπολειπτικών μπλοκ.
- Πιθανή παρεμβολή ενδο-προσοχής, δηλαδή σε κάθε εικόνα ο μηχανισμός προσοχής όπως προτείνεται για το SAGAN.
- Συνελικτικό φιλτραρισμα των χαρακτηριστικών εικόνας και κειμένου ταυτόχρονα για απο κοινού μάθηση των εξαρτήσεων, προς ταξινόμησης της τριάδας εισόδου (πρόταση, ιστορία, εικόνα).

1.3.3 Διευκρινιστής Ιστορίας

Ο διευκρινιστής ιστορίας D_{st} λειτουργεί πολύ παρόμοια με αυτόν στο StoryGAN. Σκοπός του είναι να επιβάλει συνοχή και ουσιαστική εξέλιξη κατά μήκος της ακολουθίας εικόνων $(I_1, \ldots, I_T)$ με την εκμάθηση ενός κοινού χώρου χαρακτηριστικών για προτάσεις και εικόνες. Τα χαρακτηριστικά της εικόνας υποβάλλονται σε μείωση δειγματοληψίας χρησιμοποιώντας το ίδιο είδος υπολειπτικού μπλοκ με το διευκρινιστή εικόνας, για να τα προβάλλει σε ένα χώρο που προορίζεται για κοινή χρήση με τα χαρακτηριστικά κειμένου. Όλα τα χαρακτηριστικά εικόνας για την ίδια ιστορία συνενώνονται σε ένα ενιαίο διάνυσμα.

Από την πλευρά του κειμένου, ένα πλήρως συνδεδεμένο επίπεδο αντιστοιχίζει όλες τις αναπραστάσεις προτάσεων $(\varphi_1, \ldots, \varphi_T)$ σε διανύσματα σε αυτόν τον κοινό χώρο, και επίσης συνενώνονται σε ένα μεγάλο διάνυσμα χαρακτηριστικών κειμένου. Στη συνέχεια, τα δύο διανύσματα πολλαπλασιάζονται ανά στοιχείο και το αποτέλεσμα περνά μέσα από ένα πλήρως συνδεδεμένο στρώμα για να εξάγουμε τη βαθμολογία ομοιότητας:

$$D_{st}\left((I_1, \ldots, I_T), (\varphi_1, \ldots, \varphi_T)\right) = \sigma\left(\left(W^{st}(\text{Image}(I_1, \ldots, I_T) \odot \text{Text}(\varphi_1, \ldots, \varphi_T)\right) + b\right)$$

1.4 Πειράματα

1.4.1 Πείραμα: Κανόνας ενημέρωσης τριών χρονικών κλιμάκων

Εμπνευσμένοι από το [18], προσπαθούμε να βρούμε μια βελτιστη αναλογία ρυθμών μάθησης για τα τρία δίκτυα διατηρώντας παράλληλα αναλογία ενημέρωσης 1/1/1 με σκοπό την πιο αποδοτική εκπαίδευση. Η αρχιτεκτονική που χρησιμοποιούμε για αυτές τις δοκιμές είναι αυτή που φαίνεται στα σχήματα της προηγούμενης ενότητας, εξαιρώντας τους μηχανισμούς προσοχής. Για τα ακόλουθα πειράματα χρησιμοποιούμε το Adam optimizer [25] με $\beta_1 = 0,5$ και $\beta_2 = 0,999$. Μετά από 20 εποχές, οι ρυθμοί μάθησης μειώνονται στο μισό, ως συνηθίζεται για την προσέγγιση ελαχίστου.

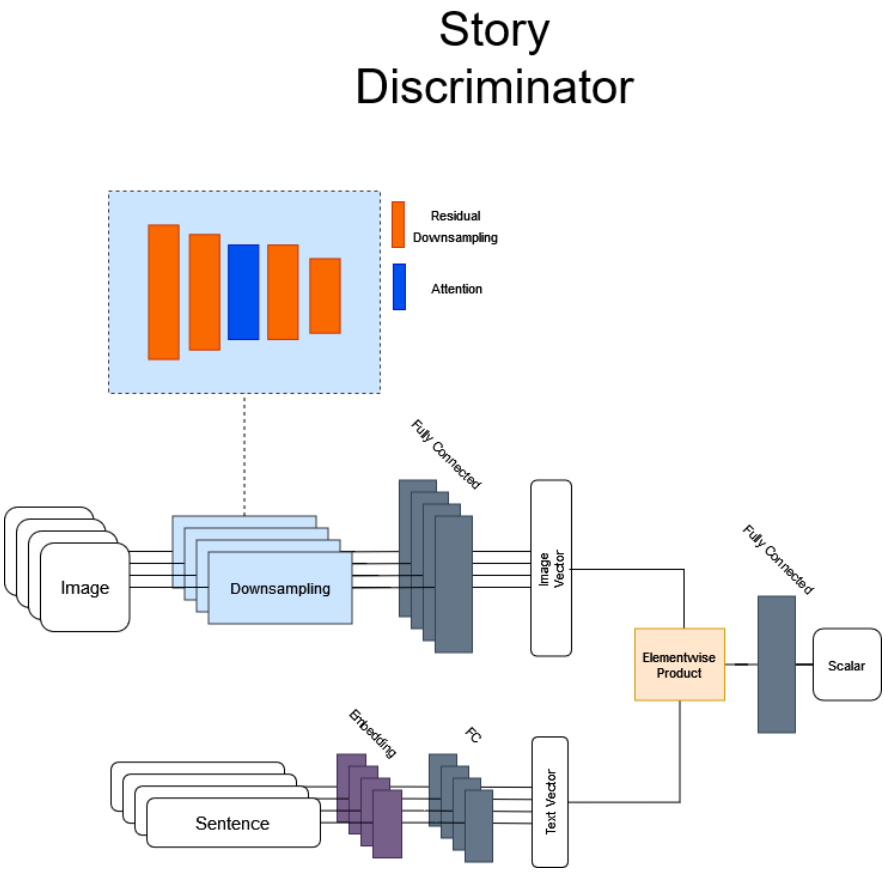


Figure 1.3.3: Διευκρινιστής Ιστορίας

Παρατηρούμε ότι όταν ο γεννήτορας μαθαίνει γρηγορότερα από τους διευκρινιστές, ολόκληρο το μοντέλο υποφέρει από κατάρρευση συστήματος:

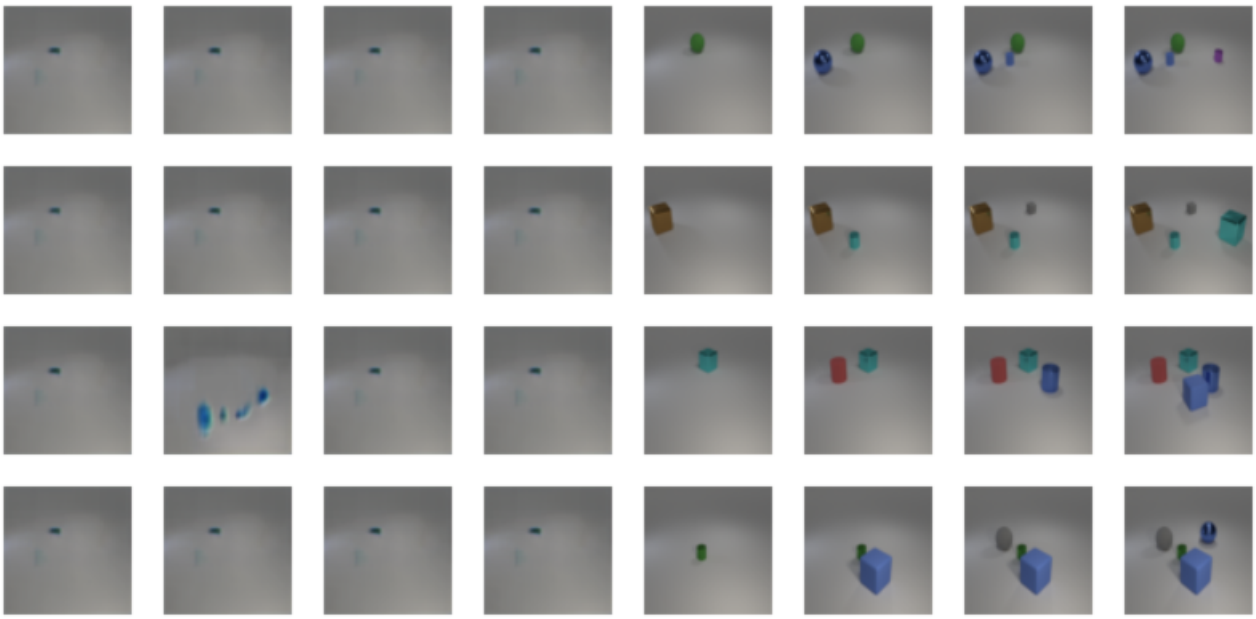


Figure 1.4.1: (αριστερά) Παραχθείσα ακολουθία (δεξιά) Πραγματική ακολουθία

Ο γεννήτορας ξεγελάει εύκολα και τους δύο διευρινιστές από νωρίς (ακόμη και όταν δημιουργεί σχεδόν την ίδια ακολουθία) οδηγώντας την εκπαίδευση σε αδιέξοδο, δεδομένου ότι οι διακρίσεις δεν μπορούν να παράγουν σημαντικές παραγώγους που να καθοδηγούν τη δημιουργία εικόνων.

Όταν διατηρείται χαμηλός ο ρυθμός του γεννήτορα, η αύξηση του ρυθμού του διευκρινιστή εικόνας αποδεικνύεται ότι οδηγεί το γεννήτορα στη δημιουργία εικόνων που αντιστοιχούν πιο κοντά στις πληροφορίες εισόδου.

Ο γεννήτορας μαθαίνει πιο γρήγορα τη σωστή αντιστοίχιση για το χρώμα και σχήμα μεταξύ εικόνας και περιγραφής.

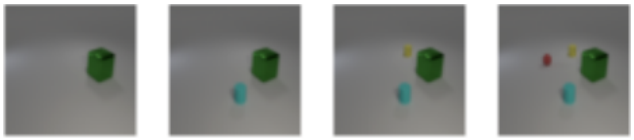
Αυξάνοντας το ρυθμό εκμάθησης του διευκρινιστή ιστορίας, παρατηρούμε αμέσως μεγαλύτερη συνέπεια σε όλες τις εικόνες. Τα αντικείμενα διατηρούν το πλήθος και τη θέση τους σε όλη την ιστορία τις περισσότερες φορές. Οι χαμηλότεροι ρυθμοί μάθησης φαίνεται επίσης να επηρεάζουν αντιστοίχιση κειμένου-εικόνας, με το γεννήτορα να δημιουργεί εικόνες με λάθος χρώμα, σχήμα και μέγεθος πολύ πιο συχνά.

Επομένως, υποστηρίζουμε ότι είναι ωφέλιμο για τους δύο διευκρινιστές να μαθαίνουν περίπου 4 φορές πιο γρήγορα από το γεννήτορα. Θεωρούμε ότι τα $lr_G = 0,0001$, $lr_{D_{im}} = 0,0004$, $lr_{D_{st}} = 0,0004$ είναι βέλτιστα, καθώς υψηλότεροι ρυθμοί μάθησης αποδείχθηκαν υπερβολικά γρήγοροι. Χρησιμοποιούμε αυτήν τη διαμόρφωση για μεταγενέστερα πειράματα, εκτός εάν διευκρινίζεται διαφορετικά.

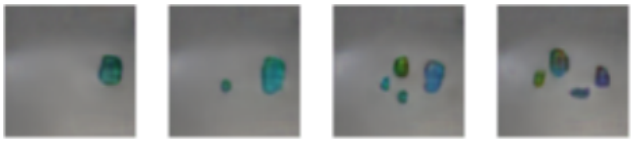
1.4.2 Πείραμα: Αμερόληπτος Κωδικοποιητής

Υποστηρίζουμε ότι η χρήση ενός Transformer Encoder είναι βέλτιστη για την κωδικοποίηση του νοήματος στα διανύσματα συνθήκης μιας ακολουθίας, αλλά το να εντοπιστούν βέλτιστες αντιστοιχίσεις για τα χαρακτηριστικά ενός συνόλου δεδομένων σε τέτοια περίπλοκη διεργασία μπορεί να αποδειχθεί δύσκολο. Τα διανύσματα ενημερωμένα από το πλαίσιο είναι απαραίτητα και για το γεννήτορα και τον διευκρινιστή εικόνας. Θεωρούμε ότι ο διευκρινιστής ιστορίας μπορεί να μάθει επαρκείς αντιστοιχίσεις για αναπαραστάσεις κειμένου και εικόνων από κοινού χωρίς καμία άμεση ανάγκη άλλης επεξεργασίας, καθώς εξετάζει ολόκληρες αλληλουχίες παράλληλα.

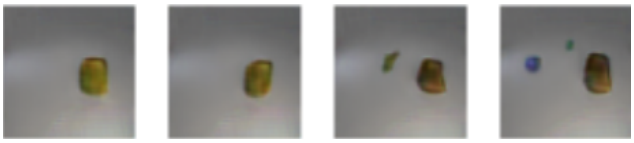
Διερευνούμε τη χρήση ενός "Αμερόληπτου" κωδικοποιητή, του οποίου οι παράμετροι ενημερώνονται από κοινού από το γεννήτορα και τον διευκρινιστή εικόνας. Υποθέτουμε ότι ένας τέτοιος κωδικοποιητής θα μάθαινε μια αναπαράσταση ακολουθιών που απλώς κωδικοποιεί το απαραίτητο πλαίσιο χωρίς να δίνει πλεονέκτημα σε οποιοδήποτε



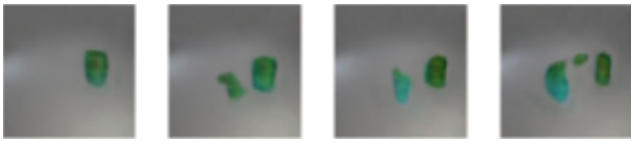
(a) Πραγματική ακολουθία



(b) LR: 0,0001



(c) LR: 0,0002



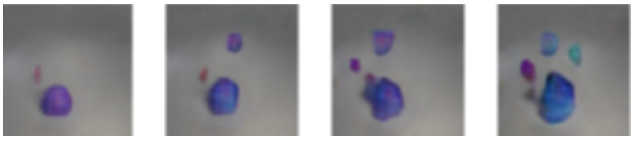
(d) LR: 0,0004

Figure 1.4.2: Αύξηση του ρυθμού εκμάθησης του Image Discriminator.

(a) Πραγματική ακολουθία

(b) LR: 0,0001

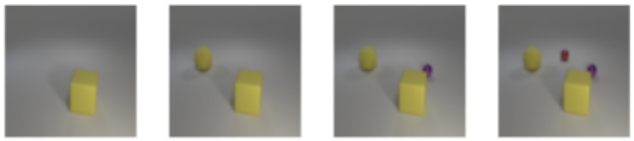
(c) LR: 0,0002



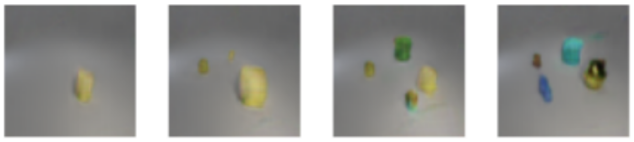
(d) LR: 0,0004

Figure 1.4.3: Αύξηση του ρυθμού εκμάθησης του Story Discriminator.

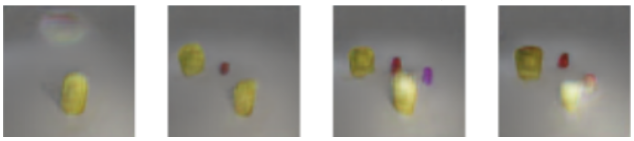
αντίπαλο. Όπως φαίνεται στην εικόνα, η θεωρία μας αποδειχθηκε σωστή. Προσπαθήσαμε επίσης να εκπαιδεύσουμε τον κωδικοποιητή να λαμβάνει επίσης παραγώγους από το διευκρινιστή ιστορίας, αλλά διαπιστώσαμε ότι αυτή η προσθήκη συγχέει τον ο κωδικοποιητή, σε σημείο να μαθαίνει εντελώς παράλογες αναπαραστάσεις του χώρου νοήματος.



(a) Πραγματική ακολουθία



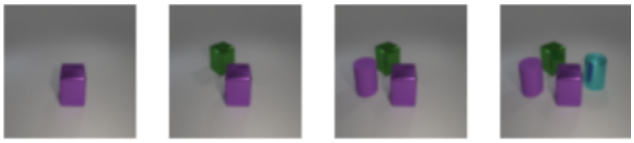
(b) Ξεχωριστοί κωδικοποιητές



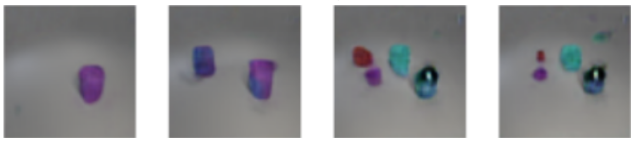
(c) Ουδέτερος Κωδικοποιητής

(d) Ουδέτερος κωδικοποιητής (με εκπαιδευση από το διευκρινιστή ιστορίας)

Figure 1.4.4: Η χρήση ενός αμερόληπτου κωδικοποιητή μετασχηματιστή φαίνεται να δίνει τα καλύτερα αποτελέσματα, παρόλο που Οι διαβαθμίσεις από το Story Discriminator βλάπτουν την απόδοση.

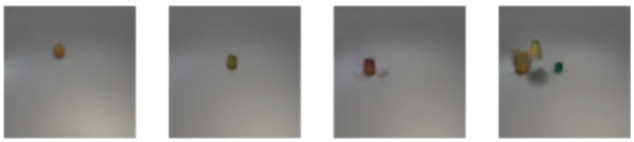


(a) Πραγματική ακολουθία



(b) Ξεχωριστοί κωδικοποιητές

(c) Ουδέτερος Κωδικοποιητής



(d) Ουδέτερος κωδικοποιητής (με εκπαιδευση από το διευκρινιστή ιστορίας)

Figure 1.4.5: Ένα άλλο παράδειγμα Αμερόληπτου Κωδικοποιητή που δείχνει τα ίδια συμπεράσματα.

Οι παράμετροι του Transformer είναι σε όλες τις περιπτώσεις: $d_{model} = 512$, κεφαλές προσοχής $N_{heads} = 8$ και $N_{layers} = 6$, ίδιο με το αρχικό σχέδιο Transformer που προτάθηκε από τους Vashwani et al. [54] Για όλα τα παρουσιαζόμενα αποτελέσματα εκπαιδεύουμε το μοντέλο για 120 εποχές. Για τα υπόλοιπα πειράματά μας χρησιμοποιούμε έναν Αμερόληπτο Κωδικοποιητή τόσο για γεννήτρια όσο και για το διευκρινιστή Εικόνας.

1.4.3 Πείραμα: Προθέρμανση Ρυθμού Εκμάθησης

Μέχρι τώρα εκπαιδεύαμε τον Αμερόληπτο Κωδικοποιητή χρησιμοποιώντας τον Adam optimizer [25] με ξεχωριστούς ρυθμους εκμάθησης για τις παραγώγους που επιστρέφονται τόσο από το γεννήτορα όσο και από το διευκρινιστή εικόνας.

Η αρχική εργασία για τον Transformer [54] προτείνει ένα συγκεκριμένο σχέδιο προγραμματισμού ρυθμού εκμάθησης που θα χρησιμοποιηθεί μαζί με τον βελτιστοποιητή Adam για εκπαίδευση Transformer αρχιτεκτονικων. Σύμφωνα με το σχήμα, ο ρυθμός εκμάθησης θα πρέπει πρώτα να αυξηθεί γραμμικά για έναν αριθμό βημάτων προθέρμανσης και στη συνέχεια να μειώνεται αναλογικά με την αντίστροφη τετραγωνική ρίζα του αριθμού των συνολικών βημάτων:

$$lrate = d_{model}^{-0,5} \cdot \min(step\_num^{-0,5}, step\_num \cdot warmup\_steps^{-1,5})$$

Ένα βήμα θεωρείται ότι είναι μια ενιαία παρτίδα δεδομένων που διέρχεται μέσω του δικτύου. Αλλαγή του ρυθμού μάθησης για $warmup_steps = 4000$ και $warmup_steps = 8000$ μπορείτε να δείτε παρακάτω:

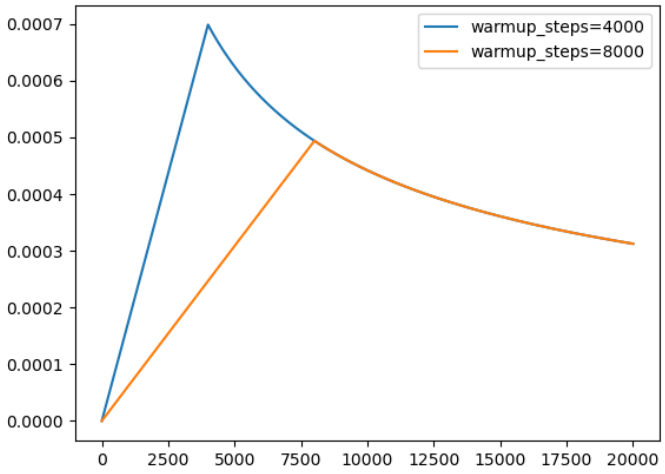


Figure 1.4.6: Ρυθμός εκμάθησης σε σχέση με τον αριθμό των βημάτων, για τη διάσταση μοντέλου 512.

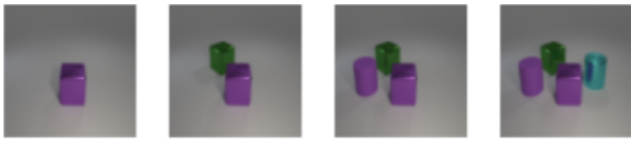
Συγκρίνουμε τα αποτελέσματα της εκπαίδευσης με αυτή τη στρατηγική βελτιστοποίησης σε σύγκριση με αυτή που περιγράφεται στα παραπάνω πειράματα.

Παρατηρούμε ότι η μέθοδος αποτυγχάνει να εκπαιδεύσει τον κωδικοποιητή περιβάλλοντος, με αποτέλεσμα ως επί το πλείστον παράλογες αναπαραστάσεις που δεν δίνουν ουσιαστικά αποτελέσματα. Υποθέτουμε ότι αυτό συμβαίνει επειδή το προτεινόμενο σύστημα βελτιστοποίησης λαμβάνει υπόψη μόνο το d_{model} και τον αριθμό των βημάτων προθέρμανσης, αναγκάζοντας έτσι το ρυθμό εκμάθησης να παραμείνει γενικά πολύ υψηλότερος από αυτό που έχουν οι βελτιστοποιητές Adam στην προηγούμενη διαταξη. Αυτό προκαλεί και την αποτυχία σύγκλισης του δικτύου.

1.4.4 Πείραμα: Υπερπαράμετροι Transformer

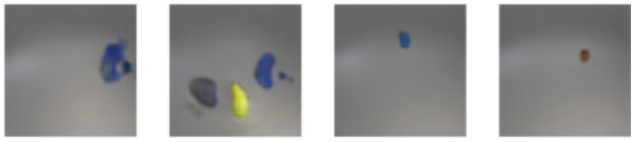
Πειραματιζόμαστε με διαφορετικές τιμές για τις παραμέτρους του Transformer Encoder.

Τα αποτελέσματά μας δείχνουν ότι ο αρχικός μετασχηματιστής με $d_{model} = 512, N_{heads} = 8, N_{layers} = 6$ είναι πράγματι βέλτιστος για το έργο μας. Η μείωση του αριθμού των κεφαλών αποδείχθηκε ότι είναι άμεσα

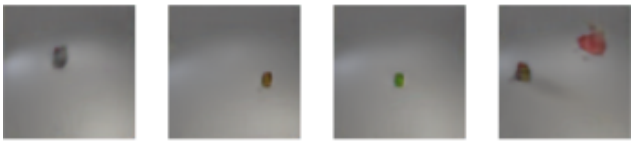


(a) Πραγματική ακολουθία

(b) Ο ρυθμός μάθησης μειώνεται κατά το ήμισυ κάθε 20 εποχές.



(c) Προτεινόμενη βελτιστοποίηση με 4000 βήματα προθέρμανσης)



(d) Προτεινόμενη βελτιστοποίηση με 8000 βήματα προθέρμανσης)

Figure 1.4.7: Σύγκριση μεθόδων προγραμματισμού του ρυθμού μάθησης.

επιζήμια για την απόδοση, ενώ οι βαθύτεροι μετασχηματιστές αποδείχθηκαν να έχουν υπερβολικά ευρύ πεδίο αναπαραστάσεων για να εκπαιδευτούν με επιτυχία σε 120 εποχές.

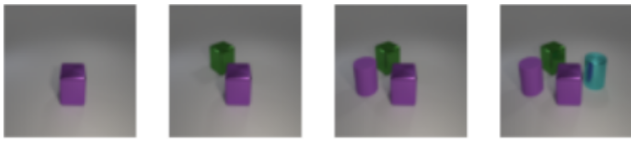
1.4.5 Πείραμα: Μηχανισμοί Προσοχής

Προσπαθούμε να χρησιμοποιήσουμε τους προτεινόμενους μηχανισμούς προσοχής όπως περιγράφονται στην προηγούμενη ενότητα σε συνδυασμό με το υπόλοιπο μοντέλο μας. Προσπαθήσαμε να χρησιμοποιήσουμε κάθε μηχανισμό προσοχής διαδοχικά στο γεννήτορα, ξεκινώντας από την Ενδο-προσοχη και μετά την προσθήκη Δια-προσοχής και Προσοχής Κωδικοιποιητή-Γεννήτορα. Προσθέτουμε επίσης Ενδο-προσοχη στους δύο διευκρινιστές.

Πειραματιστήκαμε επίσης με την τοποθέτηση των εν λόγω μηχανισμών σε διαφορετικά επίπεδα της διαδικασίας αλλαγής μεγέθους της εικόνας (αύξηση ανάλυσης - upsampling/μείωση ανάλυσης - downsampling), δοκιμάζοντας εάν οι μηχανισμοί θα ήταν επιτυχείς όταν εφαρμόζονταν σε χαρακτηριστικά υψηλότερης ή χαμηλότερης διάστασης αντίστοιχα. Παρά τις καλύτερες προσπάθειές μας, όλες οι παραλλαγές οδήγησαν αναπόφευκτα σε κατάρρευση συστήματος:

1.5 Συμπέρασματα και Μελλοντικές κατευθύνσεις

Τα αποτελέσματα του πειραματισμού μας με την προτεινόμενη αρχιτεκτονική, παρότι δεν είναι άμεσα εντυπωσιακά, έχουν αποδείξει τα πλεονεκτήματα της προσέγγισής μας. Ένας Transformer Encoder είναι ένα έγκυρο και αποτελεσματικό σύστημα για τη λήψη πληροφοριών νοήματος απο την ιστορία χωρίς την ανάγκη δομής RNN ή ειδικές τροποποιήσεις σε αναδρομικές μονάδες. Πολλαπλές παραλλαγές του μοντέλου μας έδειξαν την ικανότητα παραγωγής ακολουθιών εικόνων που εμφανίζουν οπτικά χαρακτηριστικά αντίστοιχα με αυτά στις εισαγόμενες προτάσεις. Διατήθηκε επίσης η συνέπεια σε όλα σχεδόν τα οπτικά χαρακτηριστικά εκτός από το σχήμα και το υλικό. Τα προαναφερθέντα αποτελούν απόδειξη της ισχύος των μηχανισμών προσοχής ακόμη και σε υψηλών

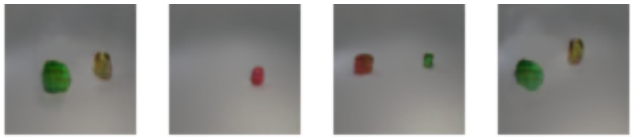


(a) Πραγματική ακολουθία

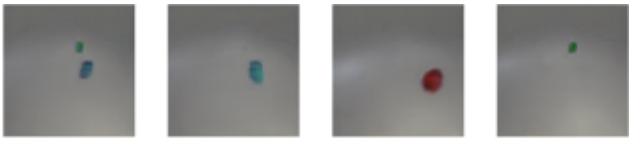
(b) Dimension: 256, Heads: 4, Layers: 6

(c) Dimension: 512, Heads: 4, Layers: 6

(d) Dimension: 512, Heads: 8, Layers: 6



(e) Dimension: 512, Heads: 8, Layers: 8



(f) Dimension: 768, Heads: 12, Layers: 6

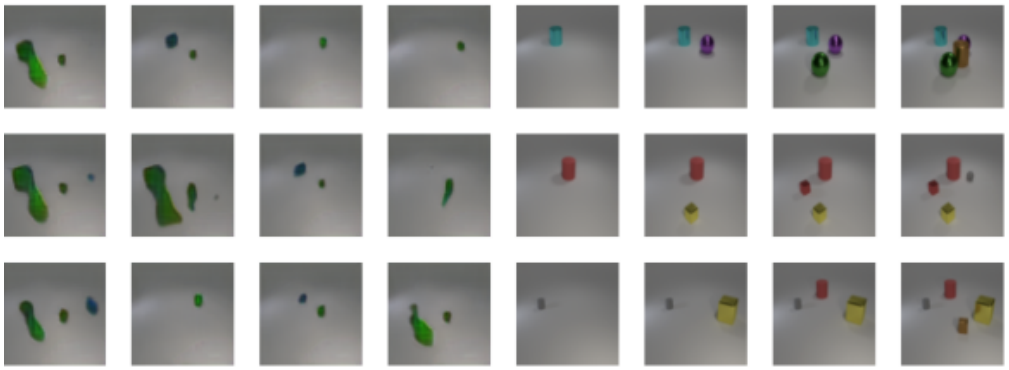
Figure 1.4.8: Σύγκριση παραμέτρων για τον Transformer Encoder.

διαστάσεων διανυσματικές ακολουθίες, όπως αυτές που αποτελούνται από εικόνες.

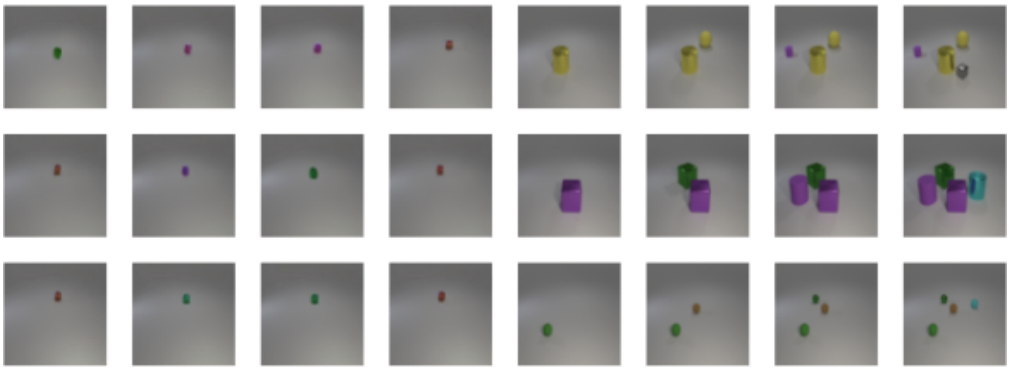
Οι προτεινόμενοι μηχανισμοί προσοχής δεν απέφεραν σημαντικά αποτελέσματα για την ώρα, ωστόσο πιστεύουμε ότι αυτό είναι λόγω περιορισμένων πειραμάτων από πλευράς μας και όχι εγγενές ελάττωμα της προσέγγισης. Θεωρούμε ότι η προσοχή τύπου Transformer Decoder - μαζί με περαιτέρω ρύθμιση υπερπαραμέτρων - είναι το κλειδί για να αντιμετωπιστούν τα σφάλματα ποιότητας που εισάγονται από το υπόλοιπο μοντέλο.

Ωστόσο, παρατηρήσαμε ένα σημαντικό αποτέλεσμα με την εισαγωγή ενός αμερόληπτου Transformer Encoder. Από όσο γνωρίζουμε, υπάρχουν λίγες έως καμία, ανταγωνιστικές αρχιτεκτονικές που χρησιμοποιούν μια προσέγγιση "διακλάδωσης", όπου ένα κοινό τμήμα του δικτύου διακλαδώνεται σε έναν γεννήτορα και έναν διευκρινιστή που εκπαιδεύονται από κοινού, λαμβάνοντας παραγώγους και από τους δύο κατά την οπισθοδιάδοση. Κατά τη γνώμη μας αυτή είναι μια γενικότερα εφαρμόσιμη διατύπωση για τα Γεννητικά Ανταγωνιστικά Δίκτυα, όπου ένα δίκτυο "διαιτητής" θα μπορούσε να λάβει σχόλια και από τους δύο αντιπάλους για να προβάλλει εκ νέου τα δεδομένα εισόδου σε μια αναπαράσταση που δημιουργεί ίσους όρους ανταγωνισμού χωρίς να ευνοεί κανέναν.

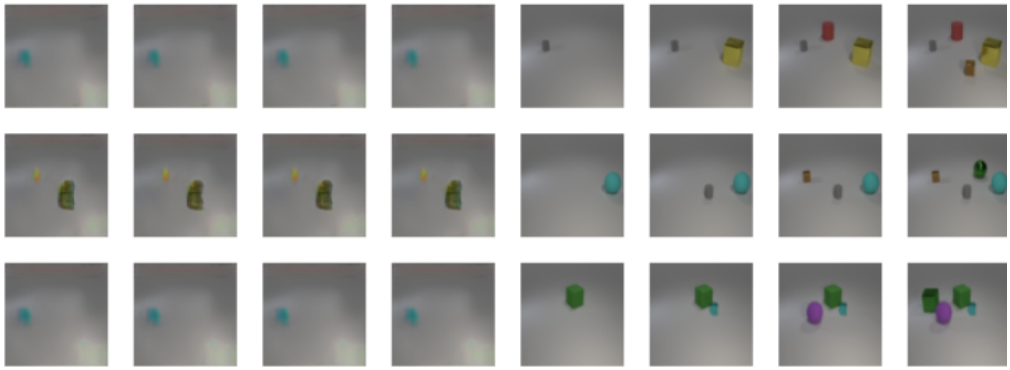
Δεν επιχειρήσαμε να εκπαιδεύσουμε το δίκτυο σε ένα πιο περίπλοκο σύνολο δεδομένων, όπως το σύνολο δεδομένων κινουμένων σχεδίων Pororo-SV [29], το οποίο δεν είναι ευρέως διαθέσιμο. Θα υποστηρίζαμε όμως να εκπαιδευτεί η παρούσα, ή καλύτερα, μια βελτιωμένη έκδοση αυτής της αρχιτεκτονικής σε ένα τέτοιο σύνολο δεδομένων για να παρατητηθεί μια πιο ουσιαστική εφαρμογή των ίδιων ιδεών σε ένα πιο δύσκολο πρόβλημα



(a) Ενδο-προσοχή σε χαρακτηριστικά μεγέθους 32x32



(b) Ενδο-προσοχή σε χαρακτηριστικά μεγέθους 16x16, με δια-προσοχη στο Generator



(c) Ενδο-προσοχή σε χαρακτηριστικά μεγέθους 32x32, με δια-προσοχη και προσοχη Κωδικοιποιητή-Γεννήτορα στο γεννήτορα

Figure 1.4.9: Όλες οι παραλλαγές των μηχανισμών προσοχής οδηγούν σε κατάρρευση συστήματος.

μοντελοποίησης. Ελπίζουμε ότι θα προκύψει μια τέτοια βελτιωμένη έκδοση, διορθώνοντας τις ελλείψεις που προκάλεσαν χαμηλή ποιότητα στις επιμέρους παραγόμενες εικόνες μας, καθώς και κατάρρευση συστήματος στο πείραμα των πολλαπλών μηχανισμών προσοχής.

Καθώς η εργασία σχετίζεται στενά με τα πολύ πιο εκτενώς μελετημένα θέματα της εικόνας από κείμενο και της μετατροπής ακολουθίας-σε-ακολουθία, για να προτείνει κανείς αρχιτεκτονικές αλλαγές ειδικά για το πρόβλημα SV πέρα από αυτό που ήδη καλύπτεται σε αυτή τη διατριβή, θα πρέπει κυρίως να προβλέψει βελτιώσεις σε οποιαδήποτε από αυτές τις κατευθύνσεις - ενώ η ροή της έμπνευσης θα έπρεπε πιθανότατα να είναι αντίθετη σε αυτή την περίπτωση. Όσον αφορά τις τρέχουσες τάσεις, ως μελλοντική εργασία θα μπορούσαμε να επιβάλλουμε μεγαλύτερη συνέπεια αντιστοίχισης κειμένου-εικόνας μέσω ενός εξωτερικού δικτύου όπως το DAMSM που προτείνεται στο AttnGAN [62].

Ωστόσο, θα θέλαμε να προτείνουμε δύο σχετικές ιδέες μελλοντικής έρευνας, προς όφελος ιδιαίτερα της έρευνας

για την Οπτικοποίηση Ιστορίας. Πρώτα, δεδομένου ότι τα σύνολα δεδομένων που δημιουργήθηκαν για αυτήν την εργασία είναι λίγα, θα θέλαμε να προτείνουμε μια ερευνητική ιδέα για τη δημιουργία ενός μοντέλου σχεδιασμένου για τη δημιουργία συνόλων δεδομένων: Ξεκινώντας από τα πολλά δημοφιλή σχολιασμένα σύνολα δεδομένων βίντεο [53, 44, 60] για εργασίες όπως περιγραφή φυσικής γλώσσας για βίντεο, θα μπορούσε κανείς να σχεδιάσει ένα δίκτυο παρόμοιο σε λειτουργία με μια αρχιτεκτονική σύνοψης βίντεο [55, 13] που θα ήταν εκπαιδευμένο να λαμβάνει τόσο ένα βίντεο, όσο και την αντίστοιχη περιγραφή κειμένου πολλών προτάσεων και να επιλέγει μια ακολουθία καρέ από το βίντεο για κάθε πρόταση. Η εργασία θα μπορούσε να θεωρηθεί ως σύνοψη βίντεο υπό συνθήκη, με κάποιες ιδιαιτερότητες: Για παράδειγμα, εάν υπάρχουν προτάσεις T στην περιγραφή, όχι μόνο θα πρέπει να υπάρχουν T καρε εξόδου από το βίντεο, αλλά αυτά τα καρέ θα πρέπει να είναι μια υποακολουθία των καρέ του βίντεο, που σημαίνει ότι κανένα επιλεγμένο καρέ δεν πρέπει να εμφανίζεται στην έξοδο πριν από ένα προηγούμενο του στο βίντεο. Για το σκοπό αυτό, ολόκληρη η ακολουθία κειμένου θα πρέπει να ληφθεί υπ' όψη για να παραχθεί η έξοδος και μια βαθμολογία αντιστοίχισης υπό συνθήκη θα πρέπει να λαμβάνεται υπόψη για κάθε ζεύγος καρέ-πρότασης, δεδομένων των καρέ που επιλέχθηκαν για άλλες προτάσεις.

Υπάρχει επίσης έλλειψη κατάλληλων μετρικων για την αξιολόγηση των προτεινόμενων μοντέλων οπτικοποίησης ιστορίας. Συνεπώς είναι αξιόλογη η προοπτική έρευνας για μία μετρική σχετική με την Frechet Inception Distance [18] για ακολουθίες εικόνων. Αυτή η μετρική θα πρέπει να παράγει μια τελική απόσταση για δύο σετ ακολουθιών εικόνων, λαμβάνοντας υπόψη τόσο την ομοιότητα χαρακτηριστικών όσο και την αλληλουχιακή συνέπεια.

Τέλος, ενθαρρύνουμε την προσπάθεια παραγωγής εικόνων υψηλότερης ανάλυσης διατηρώντας παράλληλα την ποιότητα, με την προσθήκη περισσότερων υπολοιπτικών μπλοκ στα δίκτυα. Δεδομένου ότι τα υπολοιπτικά μπλοκ [17] είναι καλύτερα στην εκμάθηση σύνθετων αναπαραστάσεων χαρακτηριστικών εικόνων και αντιστέκονται στην εξαφάνιση παραγώγων στα βαθιά νευρωνικά δικτυα, θεωρούμε ότι είναι πλεονέκτημα αυτής της προσέγγισης πως μια ακολουθία εικόνων υψηλότερης ανάλυσης θα μπορούσε να παραχθεί από μια καλή υλοποίηση του πλαισίου μας, απλώς αυξάνοντας τον αριθμό των υπερδειγματοληψιών απο τις οποίες θα περάσουν τα χαρακτηριστικά.

Συμπερασματικά, πιστεύουμε ότι αυτή η προσέγγιση που βασίζεται στην προσοχή για την οπτικοποίηση ιστορίας είναι η ορθή διαδρομή προς νέα μοντέλα state-of-the-art και πιθανώς μια μεγάλη βοήθεια στο συναφές έργο της δημιουργίας βίντεο από κείμενο. Παρά το γεγονός ότι αυτή η εργασία δεν είναι η καταληκτική προσπάθεια επίλυσης του προβλήματος αναφορικά με τις τελικές ακολουθίες, ελπίζουμε να έχουμε ανοίξει το δρόμο προς μια βέλτιστη αρχιτεκτονική SV, που θα εμφανιστεί σε μελλοντική έρευνα.

Chapter 2

Introduction

Contents

3.1 Original Formulation 26
3.2 Conditional GANs 26
3.3 DCGAN 26

Artificial intelligence has progressed significantly since the emergence of Alan Turing's computation theory [10] and the first Turing-complete artificial neurons [32]. What was once an attempt to unify mathematics under a singular theoretical framework has in recent decades become a pursuit to understand, model and surpass human capability in every area of activity. Machine learning, and in recent years the success of deep learning, have enabled us to create systems that achieve impressive human or beyond-human level results in tasks our brains have evolved to perform.

One of the most important fields of artificial intelligence is computer vision. Computer vision is the study of systems meant to classify and model image features in tasks related to human perception and ability. Over the years, advancements in this field have led to many important state-of-the-art innovations such as highly accurate image classification [11] and object detection [61], as well as significant advancements in healthcare AI, assisting medical diagnosis and treatment [66, 49, 2].

2.1 Motivation

2.1.1 Background

While this classification side of statistical learning saw the emergence of many successful models over the last decades, the same cannot be said for the generally more difficult generative approach. Modeling the distributions of complex data, such as natural language or realistic images, to the point of creating a system capable of generating new samples that seem credible to a human observer presents a number of challenges, especially dependent on the type of data. As hardware performance keeps improving and networks become deeper and more complex, capturing an often significantly multimodal distribution on these high-dimensional data spaces has become a research area of growing interest.

Most progress before recent years was slow, with networks such as Restricted Boltzmann Machines [48] [14, §20.2] proving unstable in training, impractical in use and producing inadequate results. The emergence of Variational Autoencoders (VAEs) [24] and Generative Adversarial Networks [15] has caused great advancements in tasks such as unconditional [22, 23] and conditional [57, 58] image generation, as well as video generation [45].

Figure 2.1.1: Images generated by StyleGANv2 [23]

In the conditional case, a great amount of effort has gone into designing models to generate visual data conditioned on text. A number of papers showcasing impressive results have been branching out into different variants, displaying a long line of architectural exploration. Reed et al. [42, 43] presented an early architecture that influenced many later designs, using an novel RNN-CNN encoder for text and generating full resolution images. The gradual scaling of features as seen in StackGAN [63] presented by Zhang et al. also spawned multiple successful descendants of its own [64, 62, 67]. There have even been some forays into generating video from text [28], although the task of video generation is still in its infancy and significantly more challenging, so notable results have yet to be achieved.

2.1.2 Story Visualization

The two aforementioned topics inspire a natural midpoint in the novel task of Story Visualization (SV), described by Li et al. [29] as the generation of an image sequence based on a short story made up of natural

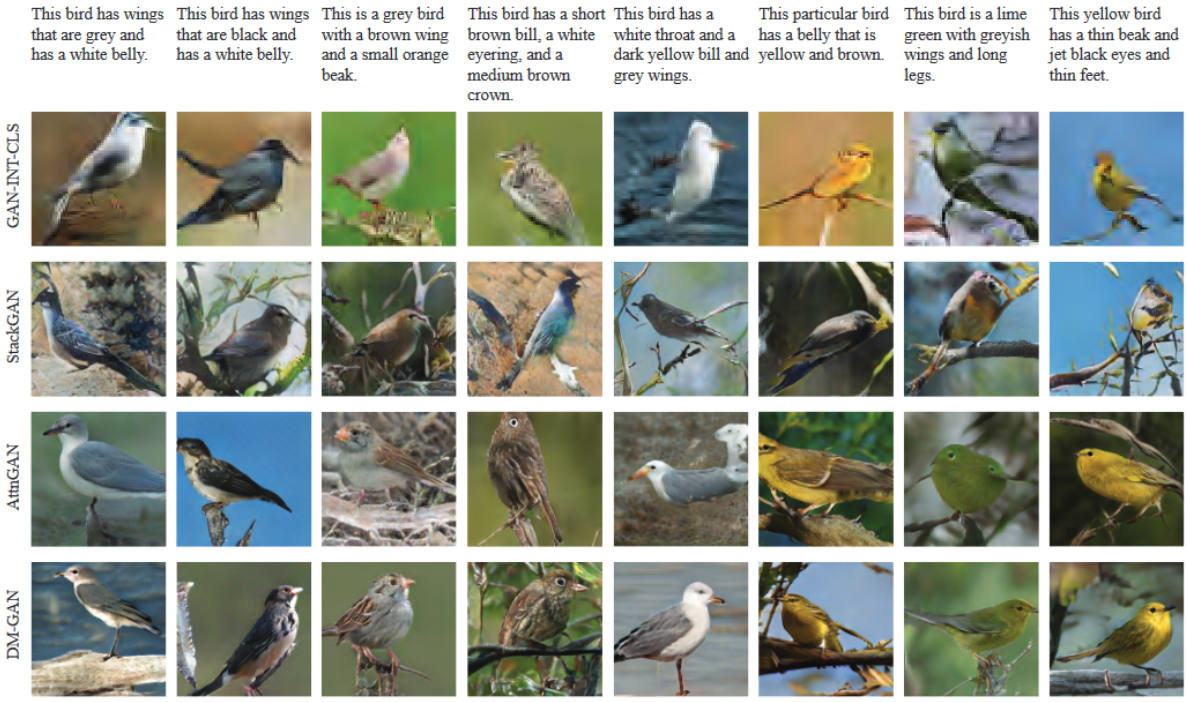


Figure 2.1.2: Comparison of Text to Image methods
"DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image" [67]

language sentences or other semantic information. The task borrows from Text-to-Image in its pursuit of language-image correspondence, as well as Text-to-Video in its aim for consistency across frames. Currently there are few improvements on this challenging topic [27] as well as a scarcity of viable datasets and evaluation methods.

2.1.3 Sequence-to-Sequence

Another way to view the same SV task is as a sequence-to-sequence transduction problem, similar to machine translation. Sequence-to-sequence models have been an area of study primarily pursued by Natural Language Processing (NLP) research, with the focus shifting gradually from Recurrent Neural Networks [51, 8] towards attention based models [5]. This trend culminated in the Transformer [54], a seminal framework that performs NLP tasks using attention mechanisms exclusively. Since its appearance, the Transformer has been favored for its simple approach, efficient training scheme and impressive results. Many prevalent models have been built atop the original Transformer for machine translation [59] and language modelling [12, 31, 41], including GPT-3 [6], a natural language model capable of performing a variety of tasks at a near-human or beyond-human level. Transformers and other attention mechanisms have also been used in computer vision tasks where they are capable of better learning complex dependencies that common convolution-based methods fail to capture [39, 65].

It is the combination of these recent advances that motivated our approach to the task of Story Visualization, in the hopes of contributing towards a model that can capture the nuances of image sequence generation and language-to-vision temporal correspondence.

2.2 Contribution

The main objective of this thesis is to research various improvements on the original StoryGAN and experiment with different implementations of our architectural proposals.

To that end we:

- Examine the effects of using a Transformer encoder in place of the original RNN.
- Apply more recent architectural approaches to the image generating GAN.
- Explore the effects of attention mechanisms in the model, both as presented in the SAGAN architecture and by proposing two novel attention mechanisms for image sequences.

2.3 Thesis Structure

This thesis consists of seven chapters, the first being this introduction. Chapters 3 and 4 aim to familiarize the reader with the theoretical background necessary to follow our experiments. Chapter 3 covers basic Generative Adversarial Network theory, while Chapter 4 examines the three GAN architectures most influential to our approach: StackGAN, StoryGAN and SAGAN. Chapter 5 describes the original Transformer framework, including a general explanation of attention mechanisms. Chapter 6 describes all the proposed elements of our model in detail, presenting an initial setup for a Story Visualization network. In Chapter 7 we relate five experiments evaluating the effect of different parameter variations and component ablations in the proposed architecture. Finally, Chapter 7 concludes this work, summarizing our findings and proposing some future directions in the research for an improved Story Visualization model.

Chapter 3

Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a training framework in which a generator network attempts to capture a desired data distribution and generate new instances while competing against a classifier called the discriminator. The discriminator's job is to learn to differentiate between samples produced by the generator from samples taken from the real data distribution. The two networks are trained in parallel until the discriminator isn't able to distinguish generated samples from the genuine ones.

Contents

4.1 StackGAN 30
4.1.1 Conditioning Augmentation 30
4.1.2 Stage-I GAN 30
4.1.3 Stage-II GAN 31
4.2 StoryGAN 31
4.2.1 Story Encoder 32
4.2.2 Context Encoder 32
4.2.3 Image Discriminator 33
4.2.4 Story Discriminator 33
4.2.5 Training 33
4.3 SAGAN 34
4.3.1 Model 34
4.3.2 Stabilization 35

3.1 Original Formulation

The original GAN framework proposed by Goodfellow et al. [15] consists of two Multilayer Perceptrons representing the objective functions $G(z;\theta_g)$ and $D(x;\theta_d)$. G is a function mapping random noise z following a prior distribution with $z \sim p_z$ to the data space. D is a function mapping a sample x_i from the same data space to a scalar value $p_i = D(x_i)$ representing the probability the sample comes from the real data distribution. θ_g and θ_d are the parameters of the generator and discriminator networks respectively.

The training of the two networks is analogous to the game theory concept of a minmax game with objective:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log(D(x))] + \mathbb{E}_{z\sim p_z(x)}[1 - log(D(G(z)))]$$

Meaning that the generator is trying to minimize the discriminator's ability to recognize its own produced samples as non-genuine. A Nash equilibrium is reached when the discriminator outputs $D(x) = D(G(z)) = 0.5$, meaning the generator is producing samples that are indistinguishable from the real ones.

According to Goodfellow et al. it is favorable in practice to use an alternative formulation of the problem where instead of minimizing $1 - D(G(z))$ the generator aims to maximize $D(G(z))$ since the original function may not provide sufficiently large gradients at the start of training, when the discriminator can distinguish between real and generated samples with high confidence.

Training such a system of networks can be done with gradient descent by backpropagating the classification error of the discriminator $J^{(D)}$ to the parameters of the generator, essentially using the discriminator network as a differentiable loss function to provide gradients for θ_g.

3.2 Conditional GANs

A conditional GAN [33] is similar to the original formulation, the only difference being in the input of the generator and discriminator network. Along with random noise z following a specified prior p_z, the generator network may receive additional information y, usually dubbed 'conditioning variable' or 'conditioning vector'. The conditioning vector is meant to guide the generator towards producing samples from different subareas of the target distribution, such as in the case of labeled image generation or sequence-to-sequence transduction. The additional information is usually provided to the generator at the input stage (concatenated with the random noise) and to the discriminator along with the sample to be evaluated.

The minmax game between generator and discriminator now has an alternative objective, defined as

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log(D(x|y))] + \mathbb{E}_{z\sim p_z(x)}[1 - log(D(G(z|y)))]$$

where y is the conditioning vector for each sample.

3.3 DCGAN

Figure 3.3.1 illustrates the generator of this seminal contribution of Radford et al. [40] to the task of image generation.

The generator makes use of transpose convolutions to upsample the random input towards a full-resolution image, while the discriminator follows a typical convolutional image classifier architecture (e.g. AlexNet [26], VGG-16 [47]) to output the probability the image is generated or real. Both networks are implemented using transpose and strided convolutions to change the resolution of images directly in intermediate layers, instead of relying on techniques such as pooling layers and nearest neighbor upsampling. This approach is meant to allow the network to learn more complex rules to directly rescale the image into the new feature space, yet it has its shortcomings: in the generator case, using transpose convolutions (deconvolutions) causes the well-known issue of the "checkerboard effect" on produced images (figure 3.3.2). Thus, in subsequent years, the decoupling of rescaling and learning of features has become prevalent, usually by means of a technique such as nearest-neighbor or bilinear interpolation, followed by a separate convolution filter.

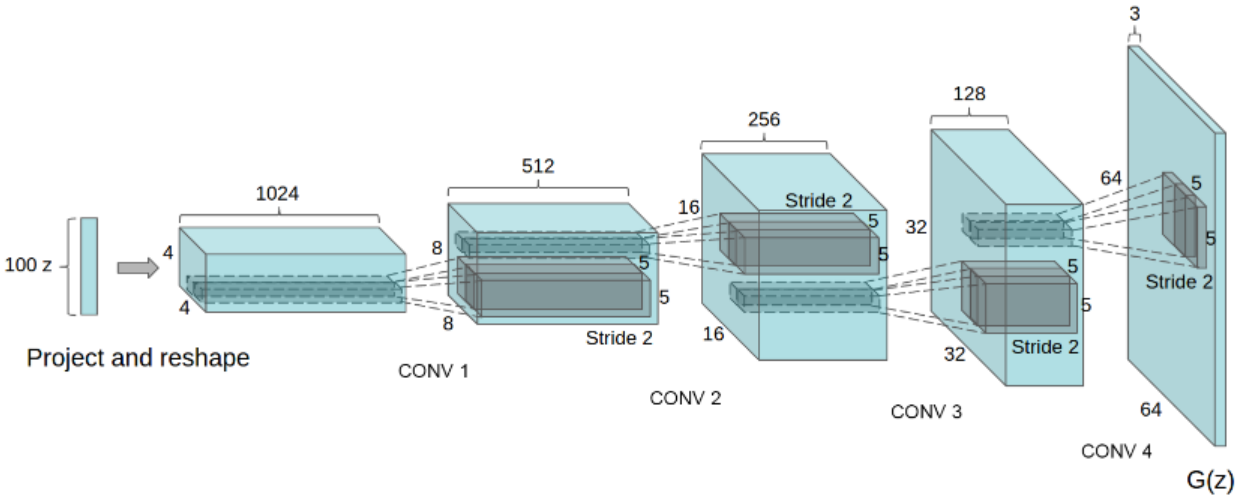


Figure 3.3.1: The DCGAN generator.

Figure 3.3.2: Comparison of transpose convolution to separate upsampling and filtering. "Deconvolution and Checkerboard Artifacts" [36]

Chapter 4

Relevant GAN Architectures

In this chapter we examine three GAN architectures that build the foundation of our contribution. These models have been designed for different conditional image generation tasks, yet in summation provide the fundamental blocks for understanding Chapter 6, where we introduce our ideas for an attention-based, modern Story Visualization GAN.

StackGAN is an important milestone for Text-to-Image generation that has impacted many works on the subject in terms of both structure and training approach. Thus, we consider it a natural predecessor to StoryGAN, the original Story Visualization model that has greatly borrowed from it, while tackling a novel but related task. We then move on to detailing StoryGAN's approach to Story Visualization as well as its architectural choices. Finally, we introduce the Self-Attention GAN as a recent conditional image generation model that employs newer architectural and training stabilization devices, widely adopted by state-of-the-art networks in the field. It also employs an attention mechansim that largely inspired our own novel mechanisms for learning features accross image sequences.

Contents

5.1 Architecture **37**
5.2 Attention **37**
5.3 Position-wise Feed-Forward Networks **39**
5.4 Positional Encoding **39**
5.5 Training **40**

4.1 StackGAN

Earlier works on image generation conditioned on text [42, 43] attempted to generate a full resolution image at once, producing samples lacking in fidelity and detail. StackGAN, proposed by Zhang et al. [63] achieved the first significantly improved results in the Text-to-Image task by gradually creating the final image with multiple stages of adversarial training. First, a low-resolution 64x64 image is generated (Stage-I GAN), capturing the broader characteristics of the image based on the text and then a second generator (Stage-II GAN) is conditioned on the first image to produce a high resolution 256x256 result. Discriminator networks train in both stages, again downsampled versions of $x_i \sim p_{data}$ in the Stage-I case. The same paper proposed a novel way of augmenting the training data, called Conditioning Augmentation (CA).

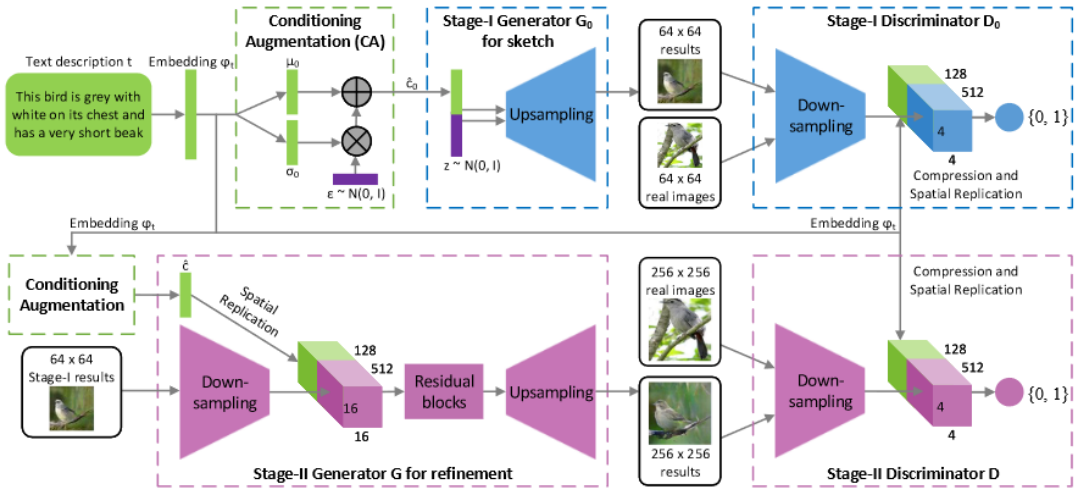


Figure 4.1.1: The architecture of StackGAN.

4.1.1 Conditioning Augmentation

The datasets used for most text-conditioned generation tasks by nature have problematic characteristics that negatively impact training. Text descriptions inevitably produce a very sparse set in a high-dimensional feature space, thus making learning a probability distribution on said set difficult. In order to make the definition space more continuous and make the network learn to be imprevious to small positional variations in embedding space, a new way of augmenting the data is proposed:

Instead of conditioning the GAN on an embedding of the text ϕ_t, a random vector $\hat{c}$ is sampled from a gaussian distribution $\mathcal{N}(\mu(\varphi_t, \Sigma(\varphi_t)))$ with the mean $\mu(\varphi_t)$ and diagonal covariance matrix $\Sigma(\varphi_t)$ being functions of the text embeddings. The vector $\hat{c}$ serves as the conditioning variable, whereas the functions themselves are implemented as neural networks with learnable parameters, trained alongside the rest of StackGAN.

Training the parameters of this stochastic process becomes possible using the reparametrization trick [24], where a sample from a gaussian distribution with arbitrary mean μ and covariance matrix diagonal σ can be produced as: $\hat{c} = \mu + z * \sigma$, where $z \sim \mathcal{N}(0, 1)$. In addition, to ensure the smoothness of the manifold, the Kullback-Leibler (KL) divergence between the learned Gaussian distribution and the standard one is added to the loss function of the generator as a regularization term:

$$D_{KL}(\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t)) \| \mathcal{N}(0, I))$$

This term helps avoid overfitting by ways of learning a "collapsed" point distribution or one that deviates too much from the standard Gaussian.

4.1.2 Stage-I GAN

The first stage of StackGAN aims to create a low resolution image capturing the basic features of the text description φ_t. After producing a conditioning vector $\hat{c_0} \sim \mathcal{N}(\mu_0, \Sigma(\varphi_t))$, using the CA network, the generator

G_0 and discriminator D_0 are trained in a typical GAN scheme by modifying their parameters through gradient descent on the following loss functions:

$$\mathcal{L}_{D_0} = \mathbb{E}_{(I,t)\sim p_{data}}[logD_0(I, \varphi_t)] + \mathbb{E}_{z\sim p_z, t\sim p_{data}}[log(1 - D_0(G_0(z, \hat{c}), \varphi_t))]$$
$$\mathcal{L}_{G_0} = \mathbb{E}_{z\sim p_z, t\sim p_{data}}[log(1 - D_0(G_0(z, \hat{c}), \varphi_t))] + \lambda D_{KL}(\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))\|\mathcal{N}(0, I))$$

where $\mathcal{L}_D$ is maximized while $\mathcal{L}_G$ is minimized. I_0,t are a real image and text description coming from the data distribution while z is the random noise sampled by a prior p_z.

The conditioning vector c_0 is concatenated with random noise z and then passed through a series of alternating nearest-neighbor upsampling and convolutional layers, to produce the 64x64 resolution image I_0. The discriminator mirrors this approach by downsampling the image towards a N_d dimensional array of 4x4 feature grids. At this point the conditioning vector is spatially replicated to form a $4x4xN_d$ tensor and concatenated along the last dimension with the image features. The output feature tensor is passed through a 1x1 convolutional layer to jointly learn from text and image features. A final fully connected layer outputs the scalar representing the discriminator's confidence in the veracity of the sample.

4.1.3 Stage-II GAN

In the second stage no additional noise input is used since the authors saw no benefit in providing the network with extra randomness beyond what is inherent in the Stage-I image. Conditioning vectors $\hat{c}$ based on the text embedding are once again produced by a different Conditioning Augmentation network, allowing the network to focus on different representations of the text while adding detail and correcting the previous stage's output. In this second stage the loss functions defining the optimization problem are:

$$\mathcal{L}_D = \mathbb{E}_{(I,t)\sim p_{data}}[logD(I, \varphi_t)] + \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{data}}[log(1 - D(G(s_0, \hat{c}), \varphi_t))]$$
$$\mathcal{L}_G = \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{data}}[log(1 - D(G(s_0, \hat{c}), \varphi_t))] + \lambda D_{KL}(\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))\|\mathcal{N}(0, I))$$

which are the same as the Stage-I expressions, except for the random noise $z \sim p_z$ being replaced by the image $s_0 = G_0(z, \hat{c_0})$.

The second stage generator follows an encoder-decoder architecture. As in the previous stage's generator, the conditioning vector $\hat{c}$ is spatially replicated and concatenated in the channel dimension with a downsampled version of the Stage-I image. The joint feature map is then passed through a series of residual blocks [17] designed to learn multiple modes of the underlying text-image distribution. The result is finally upsampled in the same way as in the Stage-I generator. The discriminator is similar to the Stage-I discriminator, requiring only additional downsampling layers to reduce the larger-sized image.

The whole model is trained with the matching-aware rule proposed by Reed et al. [42] where the discriminator is trained to classify both pairings of fake images with corresponding text descriptions and real images with mismatched descriptions as non-genuine.

4.2 StoryGAN

Li et al. introduced the task of Story Visualization [29] as a natural midpoint between the tasks of Text-to-Image and Text-to-Video generation. The purpose is to generate a sequence of images conditioned on a set of sentences that form a coherent story. The task goes beyond simple sequential application of a Text-to-Image model, since the produced images need to maintain a sense of visual and conceptual consistency and progression. A generator that is unaware of the context an image belongs to will fail at producing an adequate result.

To this end they introduce StoryGAN, a generative adversarial model that can generate images based on sequential conditioning. The network uses an RNN structure that imbues the sentence embeddings with context information, guiding the generation of an image by a conditional image generator similar in structure to StackGAN and other text-to-image architectures [42]. The generator G is trained adversarially against two discriminators. The image discriminator D_{im} is tasked with evaluating how genuine the image seems compared to the real data and how well it corresponds to the sentence, while the story discriminator D_{st} is trained to ensure consistency across images given the entire story context.

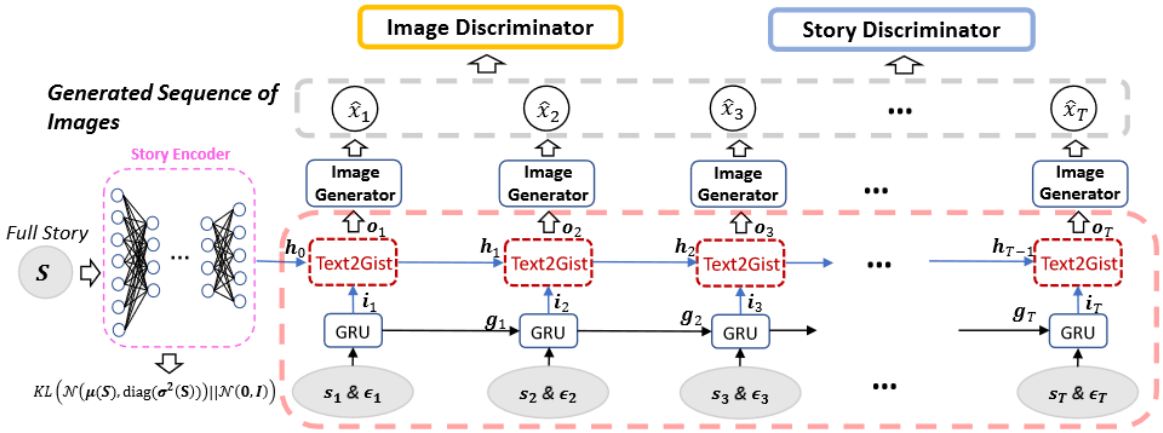


Figure 4.2.1: The StoryGAN framework.

4.2.1 Story Encoder

In order to counter the discontinuity of the original data manifold, the Conditioning Augmentation mechanism proposed by Zhang et al. [63] is used to encode the story. The entire story is mapped to a low dimensional embedding h_0 by sampling a Gaussian distribution $h_0 \sim \mathcal{N}(\mu(S), \Sigma(S))$ where μ and Σ are the functions of the story S, implemented as an Multilayer Perceptron trained alongside the rest of the network. The vector h_0 is once again produced using the reparametrization trick described in section 4.1.1. This sampled vector is provided as the initial hidden state in the context encoder RNN outlined below.

The same regularization term is added to the generator loss

$$\mathcal{L}_{KL} = D_{KL}(\mathcal{N}(\mu(S), \Sigma(S)) \| \mathcal{N}(0, I))$$

to ensure the conditional manifold remains smooth and the approximated distribution doesn't collapse to a single point.

4.2.2 Context Encoder

To produce the conditioning vector for each generated image, a stacked RNN structure is utilized. The lower layer of the RNN uses standard GRU cells [9], while the upper uses a variant of GRU cells proposed by Li et al. called Text2Gist cells. This second layer is the one whose hidden state is initialized with the story encoding h_0. For each step t in the sequence the GRU layer receives isometric Gaussian noise ϵ_t concatenated with the sentence s_t and the output is fed to the Text2Gist layer that combines it with information derived from the story context. The final output o_t is the vector conditioning the image generation. If g_t, h_t are the hidden states of GRU and Text2Gist cells repectively, the stacked RNN is structured as:

$$i_t, g_t = GRU(s_t, \epsilon_t, g_{t-1})$$
$$o_t, h_t = Text2Gist(i_t, h_{t-1})$$

The formulation of the proposed Text2Gist cells is as follows:

$$z_t = \sigma_z(W_z i_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma_r(W_r i_t + U_r h_{t-1} + b_r)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \sigma_h(W_h i_t + U_h(r_t \odot h_{t-1} + b_h))$$
$$o_t = Filter(i_t) * h_t$$

where z_t and r_t are update and reset vectors that work essentially in the same way as in the original GRU cell. $Filter(.)$ is a function that maps the vector i_t to a multichannel filter used for 1x1 convolution with state h_t, that is meant to blend local and contextual information more effectively in the process of creating a conditioning vector.

4.2.3 Image Discriminator

The image discriminator D_{im} of StoryGAN works very similarly to the one proposed in the StackGAN paper, feeding a single image - either generated $\hat{x}_t$ or real x_t - through a series of downsampling blocks and concatenating it with the spatially replicated conditioning vector s_t, also including the entire story context in the form of the story embedding h_0 described in section 4.2.2. The context is necessary as two images with the same text description can vary greatly when informed by the rest of the story. The resulting array of channels passes through a convolutional layer and finally to a fully connected layer with sigmoid activation, mapping it to a single scalar $D_{im}(s_t, h_0, \hat{x}_t)$ or $D_{im}(s_t, h_0, x_t)$ representing the discriminator's verdict.

4.2.4 Story Discriminator

The Story Discriminator maps both the story and the generated sequence into a common space in order to calculate a similarity score between them.

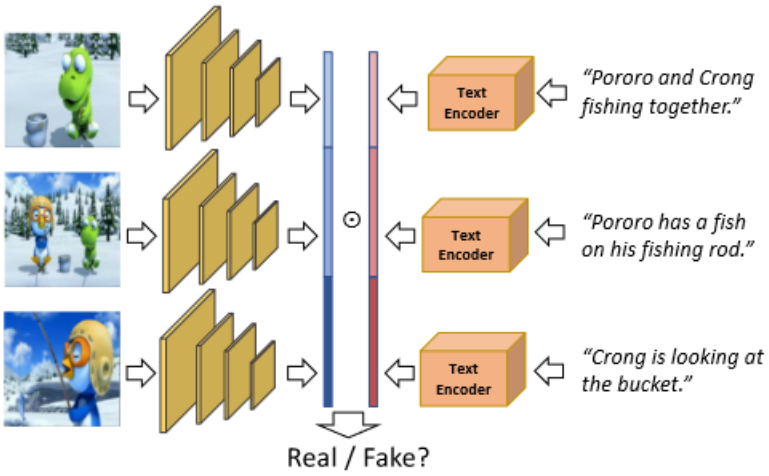


Figure 4.2.2: Structure of the Story Discriminator

An image encoder produces a series of feature vectors $E_{img}(\mathbf{X}) = [E_{img}(x_1), \ldots, E_{img}(x_T)]$ from an input image $\mathbf{X}$ that are concatenated into a single storyboard vector while a text encoder does the same for all the sentences in the conditioning story $\mathbf{S}$, producing a series of text feature vectors $E_{txt}(\mathbf{X}) = [E_{txt}(s_1), \ldots, E_{txt}(s_T)]$, that are also concatenated. The final big vectors are multiplied elementwise and fed through a learned linear transform, equivalent to a fully connected layer with bias and sigmoid activation:

$$D_{st}(\mathbf{X}, \mathbf{S}) = \sigma(w^T(E_{img}(\mathbf{X}) \odot E_{txt}(\mathbf{S})) + b)$$

where $D_{st}(\mathbf{X}, \mathbf{S})$ is the final consistency score normalized in $[0, 1]$.

4.2.5 Training

The objective function for StoryGAN is:

$$\min_{\theta} \max_{\psi_I \psi_S} \alpha \mathcal{L}_{im} + \beta \mathcal{L}_{st} + \mathcal{L}_{KL}$$

where θ, ψ_I, ψ_S are the parameters of the generator, image discriminator and story discriminator respectively, and:

$$\mathcal{L}_{im} = \sum_{t=1}^{T} (\mathbb{E}_(x_t, s_t)[log D_{im}(x_t, s_t, h_0; \psi_I)] + \mathbb{E}_{(\epsilon_t, s_t)}[log(1 - D_{im}(G(\epsilon_t, s_t; \theta), s_t, h_0; \psi_I))])$$

$$\mathcal{L}_{st} = \mathbb{E}_{(\mathbf{X}, \mathbf{S})}[log D_{st}(\mathbf{X}, \mathbf{S}; \psi_S)] + \mathbb{E}_{\epsilon, \mathbf{S}}[log(1 - D_{st}([G(\epsilon_t, s_t; \theta)]_{t=1}^{T}), \mathbf{S}; \psi_S))]$$

and constants α and β chosen before training to balance the losses of the two discriminators.

4.3 SAGAN

Recent image generation cGANs [37, 34] seem to do better in images when texture and color are the most important features but struggle with the structure of objects and other long-range dependencies. The issue can be attributed to the highly local nature of convolutional filters, the main layer type used for generating high resolution images, usually preferred due to their computational efficiency.

The Self-Attention GAN [65] is a model proposed by Zhang et al. to counter this effect, motivated by the recent popularity surge of attention models in natural language processing tasks [5, 54] as well as image generation [62, 39]. Alongside the proposed self-attention module introduced in both the generator and discriminator, the authors endorse the use of modern stabilization techniques for training, such as Spectral Normalization of weights [35] and the two-timescale update rule [18].

4.3.1 Model

The model follows the typical structure of an upsampling generator and a downsampling discriminator as described in the architectures above. In the generator, the dataset labels are embedded, reshaped and upsampled to produce the output image. The discriminator downsamples a proposed image, integrates the encoded image features with the conditioning embedding and produces a scalar output. Both upsampling and downsampling are performed by residual blocks [17] transforming the intermediate image features with convolutional layers. The unique architectural feature of the model is the self-attention module that is introduced once among the rescaling blocks in each network.

The self attention module, inspired by the non-local model introduced by Wang et al. [56], works by linearly mapping image features (channels) to the key, value and query vectors (using Transformer terminology, see section 5.2).

Given an image feature tensor $\boldsymbol{x} \in \mathbb{R}^{C \times N}$, where N are the total locations on the image plane and C is the feature (channel) dimension, attention scores are calculated for the weighted averaging of the value vectors:

$$\beta_{j,i} = \underset{i}{softmax}(s_{ij}), \text{ where } s_{ij} = \boldsymbol{f_q(x_i)}^T \boldsymbol{f_k(x_j)}$$

where f_q, f_k are learned linear transforms of x (implemented as 1×1 convolutions) for the queries and keys respectively, and said averages are mapped back into the original feature space.

Thus, the outputs $\boldsymbol{o} = (\boldsymbol{o_1}, ..., \boldsymbol{o_N}) \in \mathbb{R}^{C \times N}$ are calculated as:

$$\boldsymbol{o_j} = \boldsymbol{f_o}\left(\sum_{i=1}^{N} \beta_{j,i} \boldsymbol{f_v(x)}\right)$$

where $\boldsymbol{f_v}, \boldsymbol{f_o}$ again are learned linear transforms for the value vectors and the output remapping to the original feature space.

With $\boldsymbol{f_q}, \boldsymbol{f_k}, \boldsymbol{f_v}$ the authors elect to reduce the channel number from C to $\bar{C} = \frac{C}{k}$, where $k = 1, 2, 4, 8$ for efficiency, and empirically claim no reduction in performance. The final output is an interpolation between the attention module output and the input features:

$$\boldsymbol{y_i} = \gamma \boldsymbol{o_i} + \boldsymbol{x_i}$$

Using the learned parameter γ intialized to 0 the network can initially focus more on local information provided by the convolutional layers and gradually incorporate more distant information provided by the attention module.

The GAN is trained using the hinge formulation of the discriminator loss [30, 35]:

$$\mathcal{L}_D = -\mathbb{E}_{(x,y)\sim p_{data}}\left[\min(0, -1 + D(x, y))\right] - \mathbb{E}_{z\sim p_z, y\sim p_{data}}[\min(0, -1 - D(G(z), y))]$$
$$\mathcal{L}_G = -\mathbb{E}_{z\sim p_z, y\sim p_{data}} D(G(z), y)$$

4.3.2 Stabilization

Spectral Normalization

Lipschitz continuity is a strong form of function continuity, requring a function to be limited in its rate of change. Intuitively, a constant limit on the slope of the line between any two points of the function's graph must hold true for a function to be Lipschitz continuous. In other words, a Lipschitz constant of a real-valued function f is a positive real K such that:

$$|f(x_1) - f(x_2)| \leq K\,|x_1 - x_2|$$

Often the smallest such K is reffered to as "the" Lipschitz constant of f.

SAGAN constrains the Lipschitz constant of the discriminator [3] via spectral normalization [35], a technique that is proven to be effective in stabilizing training, while remaining computationally efficient. Inspired by Odena et al. [38] in their research on the importance of well-conditioned generators for stable training, spectral normalization is further used in the generator to improve convergence. It does so by maintaining consistent parameter norms and helping to avoid unusual gradients. Use of spectral normalization for both adversaries also reduces the need for multiple discriminator updates per iteration, which is a common technique to improve the generated sample quality in GANs.

Two Time-Scale Update Rule

SAGAN uses separate learning rates for generator and discriminator, a technique proposed by Heusel et al. [18] to combat the imbalanced learning that often occurs in GAN training between the two networks. It is common to see low-quality results from the generator being enough to fool the discriminator early in training, meaning the model has failed to converge. The TTUR allows a well designed discriminator to move towards an optimal point faster, driving the generator to produce better samples. This approach is preffered to using a different amount of training steps per epoch for each network, as it is more efficient in terms of execution time.

Chapter 5

The Transformer

Recurrent Neural Networks used to be the most prevalent neural framework for NLP with the advent of GRUs [9] and LSTMs[19] and their applications in tasks like machine translation and language modelling. Their use, no matter how effective, is by nature inefficient due to sequential calculation: the production of each symbol s_t requires a hidden state vector h_{t-1} created in the previous step. This characteristic of RNNs prevents any parallel processing of the sequence which hurts training times significantly. To counter the problem, Vashwani et al. introduced the Transformer [54], an encoder-decoder scheme which processes the entire sequence in parallel utilizing only attention mechanisms. The new network was highly efficient in training since it avoided the sequential nature of RNNs and achieved state-of-the art results in machine translation and language modelling tasks, as well as spawning highly prevalent successors [12, 31, 41, 6].

5.1 Architecture

The architecture of the Transformer is illustrated in figure 5.1.1. The model consists of 2 networks, an encoder and a decoder. The encoder's input is a sequence symbol representations $(x_1, \ldots, x_n)$ and it outputs a continuous representation of the symbols $z = (z_1, \ldots, z_n)$, in the same vein as the "annotations" in the attention encoder of Bahdanau et al. [5]. It consists of 6 layers, each one being itself a stack of two sub-layers: a multi-head self-attention layer followed by a feedforward network. Each sub-layer has a residual connection [17] around it and its outputs are normalized via Layer Normalization [4]:

$$\text{output}(x) = \text{LayerNorm}(x + \text{Sublayer}(x))$$

where Sublayer is either attention or linear mapping.

The decoder, as in most encoder-decoder schemes, works autoregressively [16], receiving as input its own previous outputs. In training it's also possible to employ the technique of "teacher forcing", feeding in the expected output from the real data distribution. These outputs are offset by one position and masked to prevent the decoder from attending to subsequent positions. The decoder structure is similar to that of the encoder with the addition of an extra sub-layer in each of the 6 layers: after self-attention, the feature sequence is fed to an encoder-decoder attention sub-layer, utilizing the encoder outputs (called "memory") as keys and values in the attention mechanism described in section 5.2.

5.2 Attention

The paper describes a general definition of attention as an operation on three sets of vectors: queries, keys and values. Keys and values can be viewed as pairings in a dictionary or table to be looked up by query-vectors of the same dimension d_{model} as the keys. This mechanism produces one output for each query vector. Each of these outputs is a vector in value-space calculated as a weighted average of all the original value vectors, with the weights being the dot product similarities of the query vector with each key, scaled by the $\sqrt{d_{model}}$ and

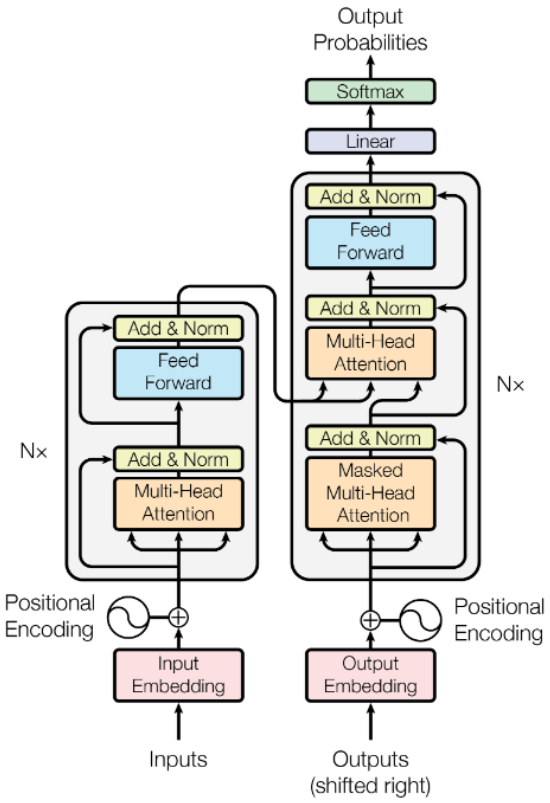


Figure 5.1.1: The transformer architecture

passed through a softmax normalization [14, §6.2.2.3]. This particular approach is called Scaled Dot-Product Attention, and other definitions of attention do exist. (e.g. additive attention [5]).

The whole process can be represented in matrix formulation as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_{model}}}\right)V$$

where Q, K, V are the matrix-packed queries, keys and values respectively.

Instead of a single attention mechanism in each sublayer, the Transformer employs a tactic called Multi-Head Attention (figure 5.2.1), where the sets of vectors are linearly projected in h different subspaces (heads) $\mathbb{R}^{d_k}$ and the attention function is calculated in each subspace. The final output for each symbol is the concatenation of the outputs of each attention head in that position of the sequence, an approach which allows the network to jointly attend to information from different representations of the same symbols.

$$\begin{aligned}\text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{aligned}$$

where all $Wi^(.)$ are parameters of the attention sublayer. The original Transformer employs $h = 8$ heads, $d_{model} = 512$ and $d_k = d_{model}/h = 64$.

This attention mechanism is used in two ways:

- Self-attention, where K, V and Q are all linear transformations of the outputs of the previous layer
- Encoder-Decoder attention in the layers of the decoder, where the queries come from the previous layer, and the keys and values come from the outputs of the encoder

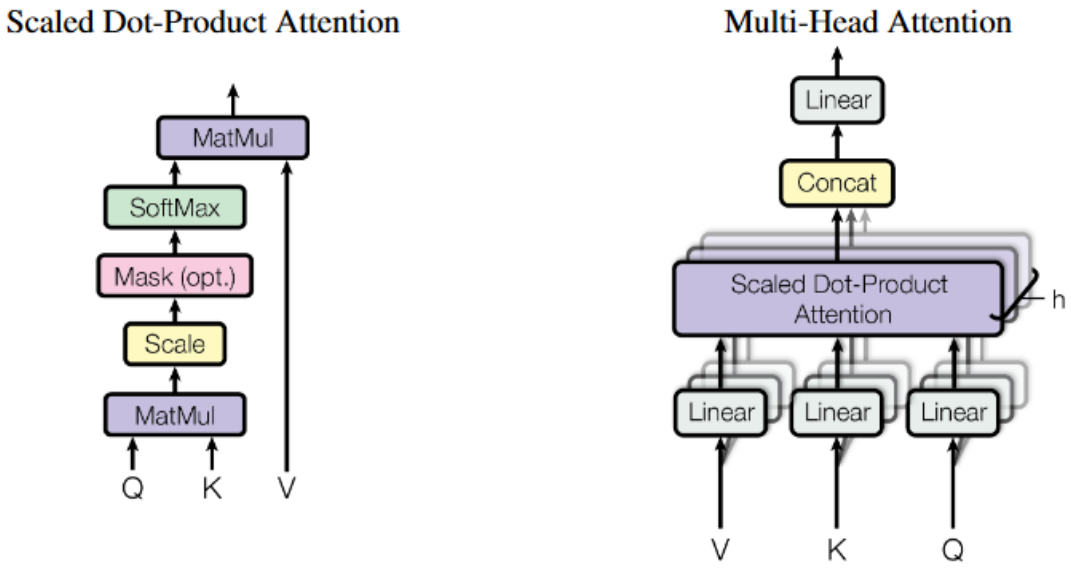


Figure 5.2.1: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention.

In the decoder, both attention layers mask positions following the current symbol being produced by setting them to $-\infty$ in the input of the softmax. This way the decoder can't attend to future information about the sequence.

5.3 Position-wise Feed-Forward Networks

Each layer of the encoder and decoder contains a fully connected feedforward net with ReLU activation:

$$\text{FFN}(x) = \max(0, xW_q + b_1)W_2 + b_2$$

The input and output of the network both have dimension d_{model} and the inner layer has dimension $d_{ff} = 2048$.

5.4 Positional Encoding

In order to use the information of each symbol's position in the sequence in the absence of recurrence or other locality-sensitive elements (e.g. convolutional layers) the embeddings need to have this information encoded into them. This is achieved by adding sine and cosine positional encodings of the same dimension to the inputs of both encoder and decoder, of the form:

$$PE_{pos,2i} = \sin(pos/10000^{2i/d_{model}})$$
$$PE_{pos,2i+1} = \cos(pos/10000^{2i/d_{model}})$$

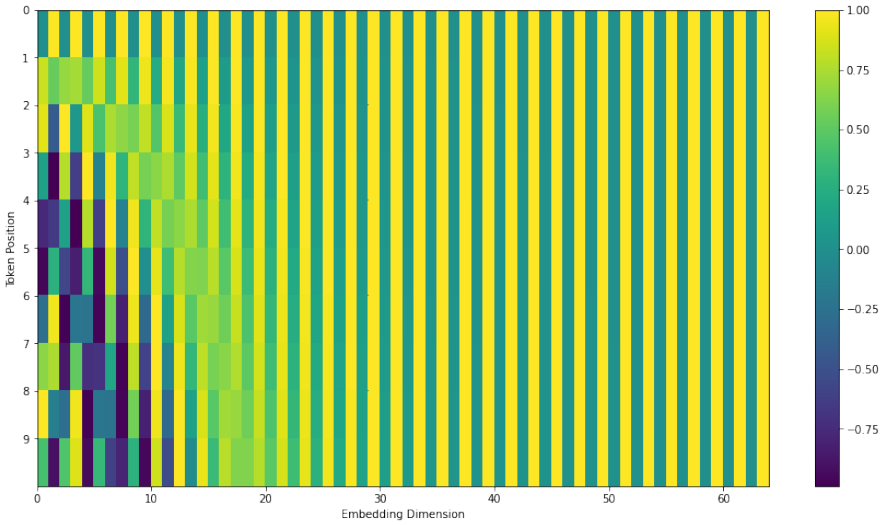


Figure 5.4.1: Positional Encodings as defined in the original transformer.
(source: "The Illustrated Transformer", J. Alammar [1])

5.5 Training

The original transformer is trained with the Adam optimizer [25], and 'warmup' learning rate scheduling. Specifically the learning rate is adjusted over the steps of the optimizer using the following rule:

$$lrate = d_{model}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

Meaning the learning rate is increased linearly for the first $warmup\_steps$ steps and then decreased following the inverse square root of the step number.

During training dropout [50] is applied for regularization to the outputs of each sub-layer and label-smoothing [46] is used to improve the model accuracy and BLEU score.

Chapter 6

Attention-based Story Visualization

In this thesis we propose an updated framework for the task of Story Visualization based on the emergence of attention-based techniques for sequence processing and modern innovations in the related task of image generation. First, we recommend the use of a Transformer Encoder [54] as a replacement for the RNN structure proposed by Li et al. for StoryGAN [29], to encode the story context into the conditioning vector for each produced image.

The authors of StoryGAN put considerable focus into creating an RNN module that captures the context of the story for each produced conditioning vector, even going as far as creating a new RNN cell. We argue that a transformer-based encoder can learn to imbue the conditioning vectors with the story context even when such context is more complex, or the sequence fairly long (the datasets used in the original paper have stories of length $T = 4$ or $T = 5$). Attention mechanisms are capable of learning long-range dependencies across symbol positions while also benefiting performance-wise from the parallel processing of the entire sequence.

As a second contribution, we recommend the use of a SAGAN-like network for adversarial learning, and experiment with the use of additional attention mechanisms to reinforce sequential consistency and progression across the features of generated images. In addition to the intra-image attention proposed in SAGAN [65] (where image locations are synthesized by attending to other locations within the same image), we also explore the effectiveness of inter-image attention mechanisms for the sequence, similar to those of a Transformer Decoder. We also detail the results of our experimentation with the various components and parameters of this framework.

Figures 6.1.1, 6.2.1, 6.3.1 showcase one proposed variant of our framework for story length $T = 4$, using all components as described in the rest of the chapter.

The structure presented is an example of the different architectural choices that can be made as we explore the effects of adding and removing different elements or changing the network parameters. The model consists of three networks, one generator and two discriminators. Similar to StoryGAN, one of the discriminators is tasked with judging individual images on their plausibility, while the second enforces consistency across images in the generated sequence.

6.1 Generator

The input to the generator network G seen in Figure 6.1.1 is a sequence of symbols s_t, possibly embedded by an appropriate encoder (such as the Universal Sentence Encoder [7], in the case of natural language sentences) into vector representations φ_t, $t \in [1, T]$ where T is the length of all stories in the dataset, and a hyperparameter of the model.

6.1.1 Conditioning Augmentation

We recommend using a conditioning augmentation module, the same as StackGAN [63], as described in section 4.1.1. The module achieves two purposes: First, as described, it promotes continuity in the data

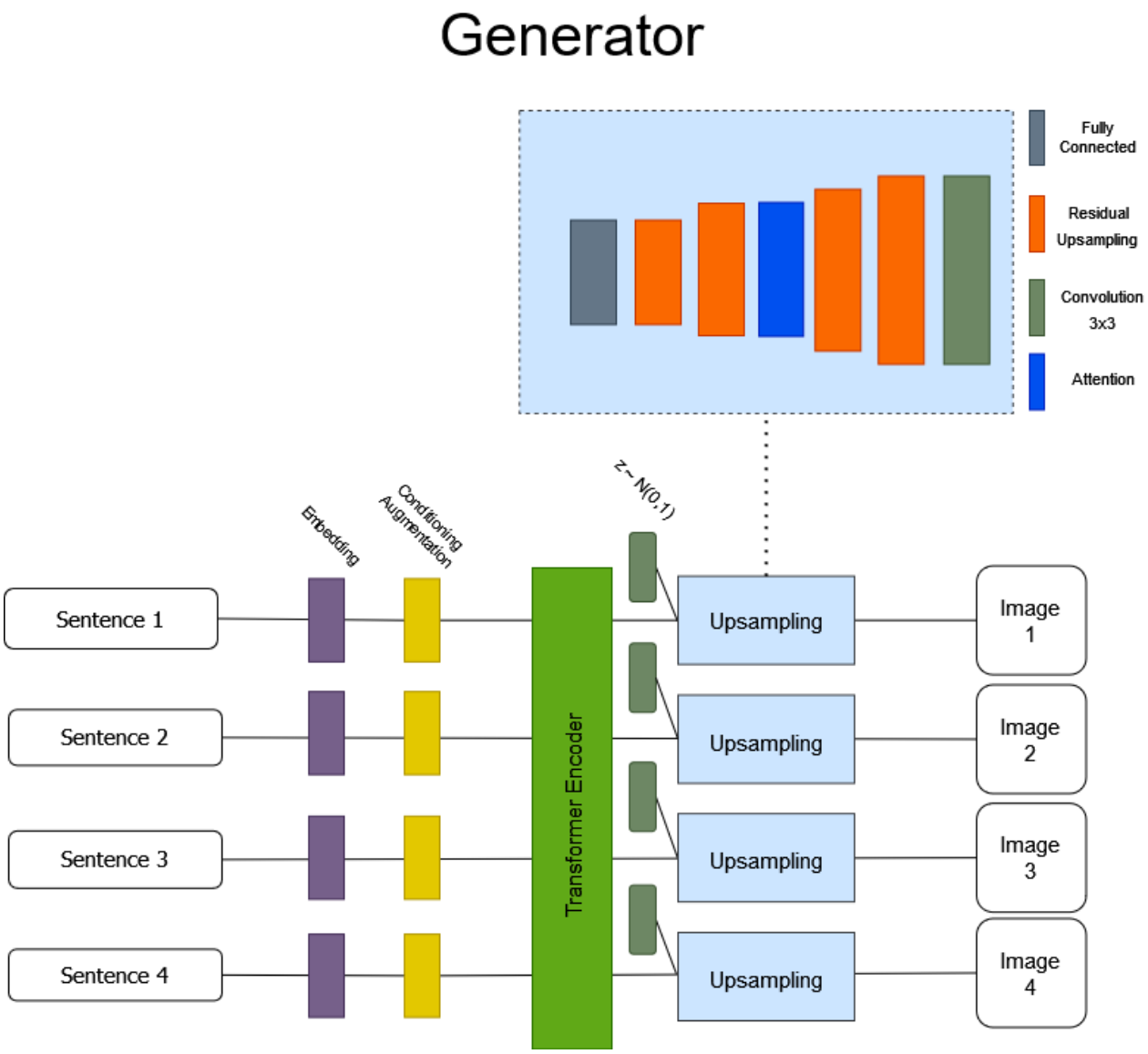


Figure 6.1.1: The generator network. In the embedding stage, the generator utilizes a transformer encoder. The attention block may contain any or all attention mechanisms described in this section, although their position relative to the upsampling blockscould vary.

manifold by sampling conditioning vectors $\hat{c}$ from $\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))$ where μ and the diagonal of Σ are learned linear transformations of φ_t. The regularization term described in Section 4.1.1 is also used in the loss of the generator.

Second, the same network can be used to map the dimension of φ_t to appropriate size since there's no option to do so beforehand in certain cases. One such case is the CLEVR-SV dataset presented later where the "sentences" are presumed to be already embedded to a vector of fixed size.

6.1.2 Transformer Encoder

The input to this encoder can either be the text embeddings φ_t or the output of the Conditioning Augmentation network $\hat{c}_t$.

The inputs are first added to positional encodings as described in the original transformer paper [54], in order for the position of the symbols to influence transduction.

A transformer encoder is then used to produce context-aware conditioning vectors $\bar{c}_t$ from the position-encoded inputs. We presume that it is capable of encoding in this new sequence of vectors, all the relevant information for the rest of the generator network to produce the image in position t, $t \in [1, T]$, relying completely on attention. The entire image sequence can then be generated in parallel as well, greatly improving training efficiency.

6.1.3 Upsampling

The context-informed conditioning vectors $\bar{c}_t$ are concatenated with gaussian noise $z_t \sim p_z$ where p_z is the random input prior (in our case $z \sim \mathcal{N}(0, 1)$). This input is fed through a fully connected layer mapping each one to dimension $C \cdot H \cdot W$ where H, W are the height and width of the initial image channels to be upsampled and C their number. This output is rearranged in a tensor $I_t \in \mathbb{R}^{C \times H \times W}$ and fed through a set of residual upsampling blocks, similar to SAGAN [65].

The purpose of a residual block [17] is to learn a mapping $F(x) = H(x) - x$ where $H(x)$ is the actual desired mapping in the underlying distribution. The final output of such a block is produced utilizing a "skip connection" such that $\hat{H}(x) = F(x) + x$. The authors of ResNet speculate that learning the residual is easier than learning the original transformation and prove successful in countering the accuracy degradation observed with increasing network depth.

In each upsampling block, the input image features $\boldsymbol{I_t}$ are normalized via Batch Normalization [20] and passed through a ReLU activation. Then, both spatial dimensions are doubled via nearest neighbor upsampling, and a convolutional filter transforms image features while halving the channel dimension, to approximately preserve computational complexity as the image planes get larger. The tensor is again normalized and passed through ReLU activation as well as a final convolutional filter.

In order to add the input to the output we perform a minimal transform on the skip connection, using nearest-neighbor upsampling to match the spatial output dimension and passing it through a learned 1×1 convolutional filter (equivalent to a linear transformation) to match the output channels.

After upsampling the features to the desired dimension $H \times W$ a final 3×3 convolution layer is used to produce a 3-channel image, followed by hyperbolic tangent activation to remap pixel values into the range $[-1, 1]$.

6.1.4 Spectral Normalization

Inspired by SAGAN, we make use of spectral normalization for our model. Spectral normalization is a method that fixes the spectral norm (the maximum singular value) of the weight matrix of a model, by normalizing the weights (meaning it does not directly change the outputs). It was proposed by Miyato et al. [35] as a stabilization technique for the training of GANs, by imposing a constraint on the Lipschitz constant of the discriminator. This is the only hyperparameter of the process, which has been in practice proven to be of little importance, since a value of 1 gives good performance in most tasks [65]. We use spectral normalization in the

convolutional layers of residual blocks in both the generator and the discriminator, based on the conclusions by Odena et al. [38] that well-conditioned generators can also affect convergence and produce better results.

6.1.5 Attention

Intra-image Self-Attention

The first of three image attention mechanisms we experiment with in this framework is the one used in SAGAN by Zhang et al. [65], described in Section 4.3.1. Operating on each image individually, an attention layer is employed that assists the generator and discriminator in modeling spatial long-range dependencies within the image.

In this formulation of attention, given an image tensor $\boldsymbol{I_t} \in \mathbb{R}^{C \times H \times W}$, "image features" refers to vectors of dimension C calculated for each location of the image plane. These features are transformed into two spaces operating as the key and value spaces as described in section 5.2 for the transformer.

The output of the self-attention layer is linearly interpolated with the input via a learnable parameter γ, which is expected to change throughout training to allow reliance on local features early on and long-range dependencies later.

Inter-image Self-Attention

As part of this framework we propose a novel attention mechanism for the generator, inspired by the Transformer Decoder [54]. Viewing the Story Visualization task as a Sequence-to-Sequence transduction we examine the effects of attention in the image generation / decoding part of the process to enforce better sequential consistency and sharing of context information between the images of the storyboard.

In a similar vein to the Transformer's multi-head attention, image feature tensors are naturally separated into distinct representations of the encoded information in the form of channels. Thus when we refer to "image features" for these attention layers, we now refer to $\boldsymbol{I_t} = (\boldsymbol{i}_{t,1}, \ldots, \boldsymbol{i}_{t,C})$ vectors of dimension equal to the number of locations on the plane.

The Inter-image Self-Attention mechanism we experiment with mimics the Transformer Decoder's self-attention, by downsampling each channel $\boldsymbol{i}_{j,t}$ where $j \in [1, C]$, $t \in [1, T]$ of the image into 3 new planes

$$\begin{aligned} \boldsymbol{q}_{j,t} &= \boldsymbol{f}_q(\boldsymbol{i}_{j,t}) \\ \boldsymbol{k}_{j,t} &= \boldsymbol{f}_k(\boldsymbol{i}_{j,t}) \\ \boldsymbol{v}_{j,t} &= \boldsymbol{f}_v(\boldsymbol{i}_{j,t}) \end{aligned}$$

where $j \in [1, C]$, $t \in [1, T]$ and $\boldsymbol{f_q}$, $\boldsymbol{f_k}$, $\boldsymbol{f_v}$ are implemented as convolutional filters to take advantage of spatial locality to improve efficiency. We downsample the features in order to make the layer more efficient, due to the large number of heads (matrix multiplications) required. We also opt for convolutional filters to take advantage of the local nature of image channels and again, maintain efficiency.

We then use the new features as the keys, queries and values and calculate the Scaled Dot-Product Attention function for each head/channel as described in Section 5.2. The outputs of the layer are then calculated via another array of convolutional filters $\boldsymbol{f_o}$, preceded by nearest-neighbor upsampling to return image features to their original dimension.

Encoder-Generator Attention

The next attention mechanism we propose directly links the outputs of the encoder to the generated features instead of relying solely on them as conditioning vectors. We model this after the Encoder-Decoder attention of the Transformer and hypothesize it will help reinforce the story information distribution in each image, since we observed great inconsistency in that area during early experimentation.

From an image feature perspective it works virtually identically to the Inter-Image Attention mechanism, producing the $\boldsymbol{k}_{j,t}$, $\boldsymbol{v}_i = j, t$ vectors for each head / channel. For the $\boldsymbol{q}_{j,t}$ vectors we linearly map the outputs of the Encoder (Section 6.1.2): $q_{j,t} = W^Q_{j,t}\bar{c}_t$ learning weight matrix $W^Q_{j,t}$ as a parameter of the network.

We then calculate Scaled Dot-Product Attention as before.

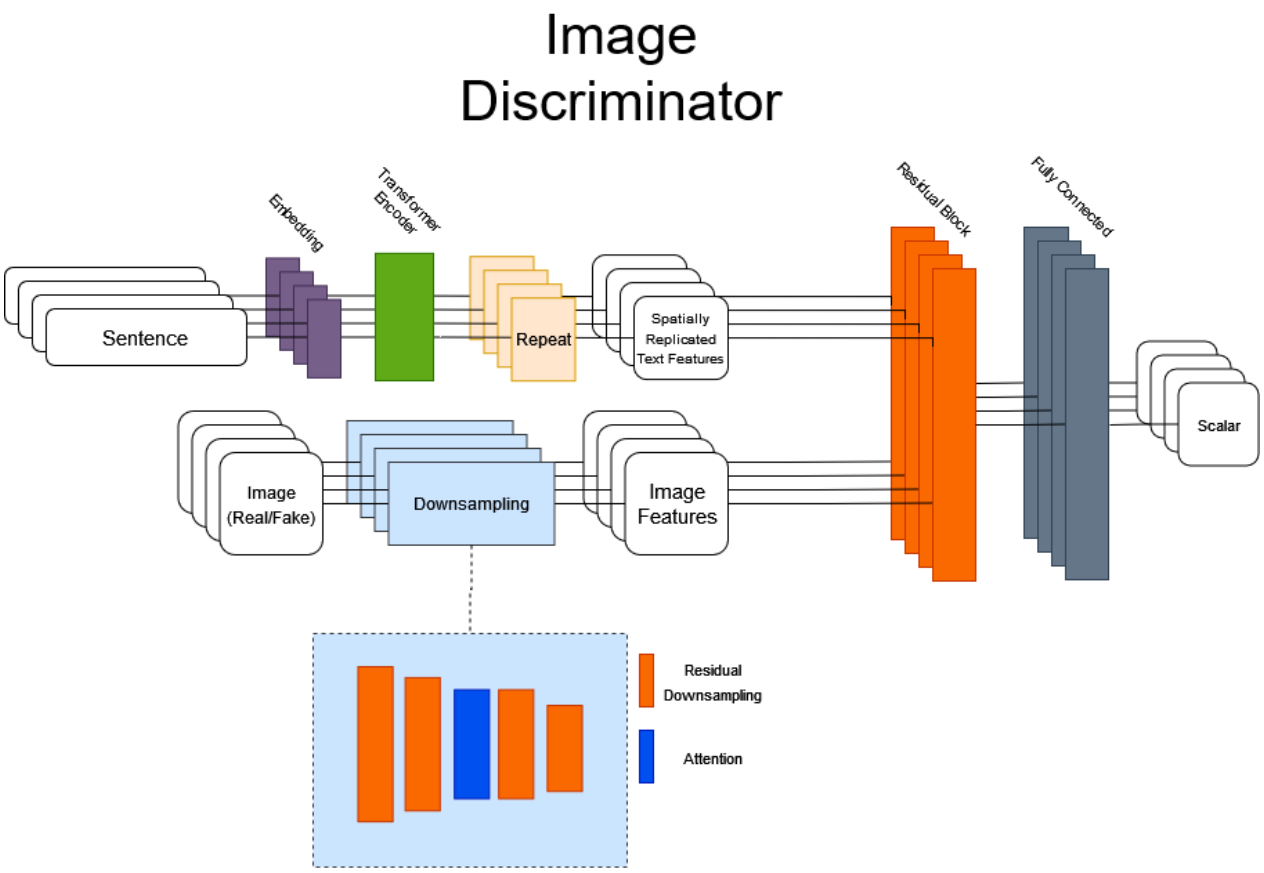


Figure 6.2.1: The Image Discriminator. The transformer encoder might be the same or a different one compared to the Generator, which is explored in Section **??**. Beyond that, D_{im} operates on each image individually. The attention block in this case refers to the Intra-image Attention module.

6.2 Image Discriminator

The purpose of the image discriminator D_{im} (Figure 6.2.1) is to discern between images from the dataset and generated images. To that end it utilizes the text features φ_t of the sentence corresponding to the image in the story, the context (the other sentences in the story) and the image I_t to be evaluated. The context is important for the discriminator, because each image corresponding to one sentence in a story depends on the rest to form many of its details. Consider the following examples:

1. "Add a red metallic cube. Then add a yellow cylinder."
2. "A silhouette was visible outside the window. It was a black cat."

In the first case, the second image depends on context to the left, while in the second case, the first image needs to be aware of context to the right.

The image discriminator is meant to classify each image individually, not as part of the sequence it belongs to. Yet all produced images of a story are evaluated in batch, to take advantage of the parallel Transformer processing.

6.2.1 Transformer Encoder

Another instance of positional encoding followed by a Transformer Encoder is used for the same purpose as described in the generator. The input embeddings for the sentences of an entire story φ_t are encoded through alternating self-attention and linear layers to produce a sequence of context-imbued vectors $\bar{c}$. Each of the

vectors is meant to be used individually for the process of testing image-text correspondence alongside the fidelity of the images.

6.2.2 Downsampling

Each image to be evaluated is passed through a series of residual downsampling blocks. Image features from each layer are first passed through a Leaky ReLU activation, then a spectrally normalized convolutional layer (see 6.1.4), remapping the $C \times H \times W$ tensor to double the channels. After another Leaky ReLU, a spectrally normalized strided convolution layer downsamples the image features. We prefer this option to a pooling layer due to the inferences made by Radford et. al in the original DCGAN work [40] (Section 3.3). The final tensor has dimension $(2C) \times (H/2) \times (W/2)$. For the skip connection, we perform a minimal transform as described in Section 6.1.3.

6.2.3 Dropout

We have found it beneficial to use Dropout [50] in all residual blocks for the discriminators, to prevent overfitting and overt coupling of individual layer units.

6.2.4 Attention

We use the same non-local intra-image attention mechanism described in sections 4.3.1, 6.1.5 in order to assist the network in learning longer-range relationships of image features.

6.2.5 Output

To produce an output scalar, each vector of dimension d_{model} given by the encoder is spatially replicated to create a $d_{model} \times H \times W$ tensor that is then concatenated with the image features along the channel axis. These features are then passed through a residual block to jointly learn from image and text features, a method inspired by the discriminator in StackGAN [64]. A final fully connected layer mapping features to a single scalar leads to a sigmoid activation function, ultimately producing a probability $D_{im}(I_t) \in [0, 1]$.

6.3 Story Discriminator

The story discriminator D_{st} functions very similarly to the one in StoryGAN described in section 4.2.4. Its purpose is to enforce consistency and meaningful progression along the image sequence $(I_1, \ldots, I_T)$ by jointly learning a common feature space for sentences and images. The image features are downsampled using the same kind of residual block as the Image Discriminator (Section 6.2.2), to map them into a space meant to be shared with the text features. All image features for the same story are concatenated into a single storyboard vector.

On the text side, a fully connected layer maps all sentence embeddings $(\varphi_1, \ldots, \varphi_T)$ to vectors in this shared space, also concatenated into one big text feature vector. The two story-wide vectors are then multiplied elementwise and the result is passed through a fully connected layer (equivalent to the StoryGAN formulation of a linear transform with bias) to output the scalar "similarity score":

$$D_{st}\left((I_1, \ldots, I_T), (\varphi_1, \ldots, \varphi_T)\right) = \sigma\left(\left(W^{st}(\text{Image}(I_1, \ldots, I_T) \odot \text{Text}(\varphi_1, \ldots, \varphi_T)\right) + b\right)$$

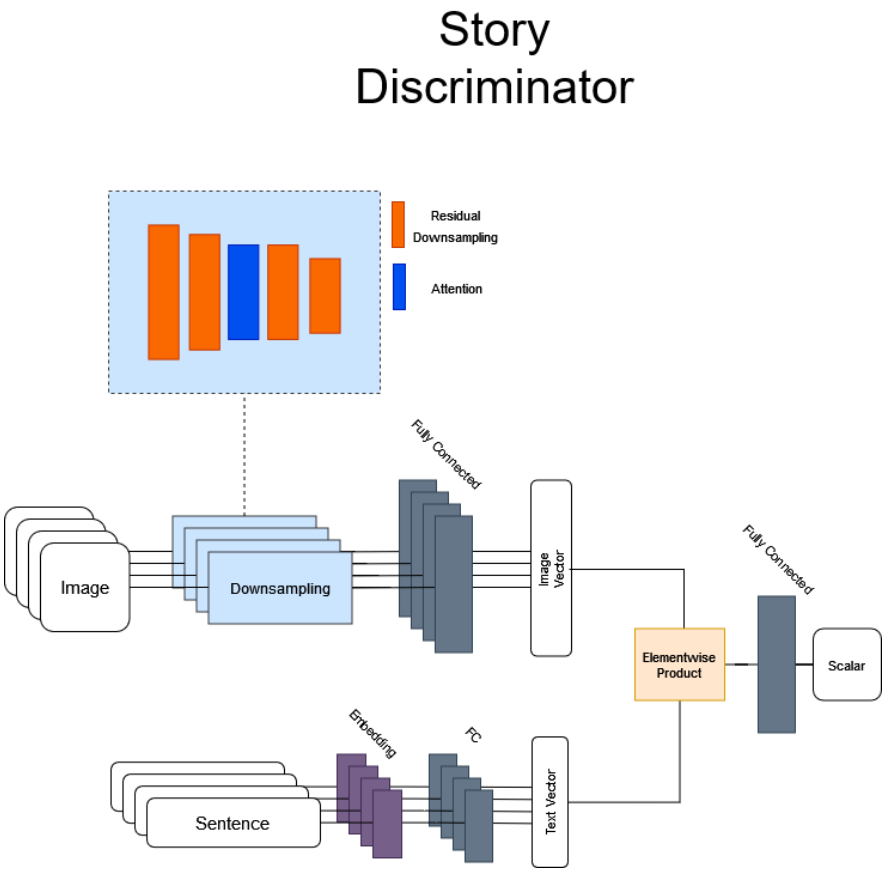


Figure 6.3.1: The Story Discriminator.

6.4 Training

Training our model requires minimizing the following three loss terms:

$$\mathcal{L}_{im} = \sum_{t=1}^{T}(\mathbb{E}_{(i_t,\varphi_t)}[logD_{im}(i_t,\varphi_t,h_0;\psi_I)] + \mathbb{E}_{(z_t,\varphi_t)}[log(1 - D_{im}(G(z_t,\varphi_t;\theta),\varphi_t,h_0;\psi_I))])$$

$$\mathcal{L}_{st} = \mathbb{E}_{(\mathbf{I},\mathbf{S})}[logD_{st}(\mathbf{I},\mathbf{S};\psi_S)] + \mathbb{E}_{\epsilon,\mathbf{S}}[log(1 - D_{st}([G(z_t,\varphi_t;\theta)]_{t=1}^{T}),\mathbf{S};\psi_S))]$$

$$\mathcal{L}_G = \mathbb{E}_{(z_t,\varphi_t)}[log(D_{im}(G(z_t,\varphi_t;\theta),\varphi_t,h_0;\psi_I))] + \mathbb{E}_{\epsilon,\mathbf{S}}[log(D_{st}([G(z_t,\varphi_t;\theta)]_{t=1}^{T}),\mathbf{S};\psi_S))] + D_{KL}(\mathcal{N}(\mu(\mathbb{S}),\Sigma((\mathbb{S}))\|\mathcal{N}(0,1))$$

where $z_t \sim p_z$, $\mathbf{I} = (I_1,\ldots,I_T)$, $\mathbf{S} = (\varphi_1,\ldots,\varphi_T)$

We use the alternative formulation following [15] for the generator to provide sufficient gradients. We also use the "matching aware" discriminator criterion as described by [42].

We also employ one-sided label smoothing by setting positive labels to 0.9 instead of 1.0, which has been recommended by Salimans et al. [46] to avoid the pitfalls of regular label smoothing [52].

For all experiments, the Adam optimizer [25] is used for gradient descent and we experiment with different learning rate values and scheduling schemes for the three networks, in consideration of the two time-scale update rule [18].

We maintain balanced updates for all three networks and pursue convergence via the aforementioned techniques to maintain training efficiency.

Chapter 7

Experiments

In this section we experiment with different versions of the framework described above to determine their effect on generated samples. We use the simpler object dataset proposed by StoryGAN as it can offer a clearer picture of the variations between different generated sequences based on the same input. We concern ourselves with the relative learning rate of the three networks, afterwards we evaluate the effects of different transformer setups, and finally we evaluate the effect of the image attention mechanisms proposed in Section 6.1.5. Since the architecture is - by nature of the task - fairly complicated, for the purposes of this thesis we elect to mostly focus on the sequential consistency and progression of generated images and the improvement different applications of attention can offer.

7.1 Dataset

There are very few existing datasets that can be utilized for the task of story visualization and none specifically tailored to it. Li et al. train StoryGAN on two datasets [29] modified for the task: An artificially generated dataset (CLEVR-SV) containing images of 3D objects and matching vector descriptions rendered with Blender, and a dataset of cartoon video clips (Pororo-SV) and matching text descriptions, normally used for video question answering.

We elect to use the CLEVR-SV dataset for our experiments as the simplicity of the images makes evaluating the result of different parameter or architecture variations more easily discernible. It also helps benchmark our architecture against the reported results of StoryGAN.

In addition, CLEVR-SV is open source while the Pororo-SV dataset is harder to obtain.

The CLEVR dataset [21] was originally designed for visual question answering. It is a framework for programmatically generating images containing 3D rendered objects constrained within specific parameters. Any given object is characterized by its shape (cube, sphere or cylinder), size (small or large), material (rubber or metal) and one of 8 colors. Four objects are added one at a time, creating a four image "story".

The input sentences φ_t are vector representations of the objects present in each image, consisting of their attributes and two real numbers indicating their position. These vectors could be considered as perfectly encoded dataset-specific embeddings of any sentences describing the sequence, e.g. "A small red cube made of metal is on the floor. A big yellow rubber cylinder is added to its left." On a dataset where descriptions are actual text, a pre-trained encoder could be used such as the Universal Sentence Encoder [7] or even a custom embedding process tailored to the data.

For training we generate 10,000 image sequences and corresponding descriptions and an additional 3,000 used for testing.

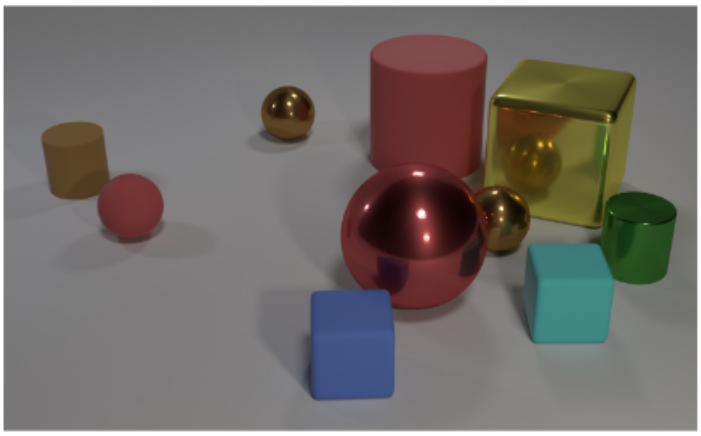


Figure 7.1.1: Example image from the CLEVR dataset

7.2 Resources

All our experiments were run on the national supercomputer system ARIS, maintained by the Greek Research and Technology Network (GRNET). For training we distributed the load among two NVIDIA K40 GPUs. Since the original StoryGAN was trained for 120 epochs to produce satisfactory results we set the goal of finding an architecture that can do better when trained for the same number of iterations. Training for the full 120 epochs on any of the described architectures takes approximately one and a half days.

7.3 Experiment: Three Time-scale Update Rule

Inspired by the Two Time-scale Update Rule [18] we attempt to find an optimal learning rate scheme for the three networks while maintaining a 1/1/1 update ratio for the most efficient training. The architecture we use for these tests is the one shown in the figures of Chapter 6 without any attention in the image scaling layers, and separate transformer encoders for generator and image discriminator (see section 7.4). For the following experiments we use the Adam optimizer [25] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. After 20 epochs, the learning rates are halved based on a typical scheduling scheme.

We observe that when the generator learns faster than the discriminators, the whole model suffers from mode collapse:

The Generator easily fools both discriminators early on, even when generating virtually the same sequence (i.e. being limited to one or a few modes of the target distribution), leading training to a stalemate since the discriminators cannot produce any meaningful gradients to guide image generation.

When maintaining a low learning rate for the generator, increasing the Image Discriminator learning rate proves to lead the Generator into creating images that correspond closer to the conditioning information. (Figure 7.3.2)

The Generator is quicker in learning the correct matching for color and shape between image and description vector, as well as learning to produce more concrete shape features, at least for large objects.

Increasing the learning rate of the Story Discriminator, we immediately observe greater consistency across images. Objects maintain their position throughout the story and the correct number of objects for each image is generated more often than not. Lower learning rates also seem to affect text-image matching, with the generator creating images with wrong color, shape and size a lot more often. (Figure 7.3.3)

We thus maintain that it is beneficial for the two Discriminators to learn about 4 times as fast as the Generator. We find $lr_G = 0.0001$, $lr_{D_{im}} = 0.0004$, $lr_{D_{st}} = 0.0004$ to be optimal, as higher learning rates proved to be too fast for convergence. We use this configuration for later experiments unless specified otherwise.

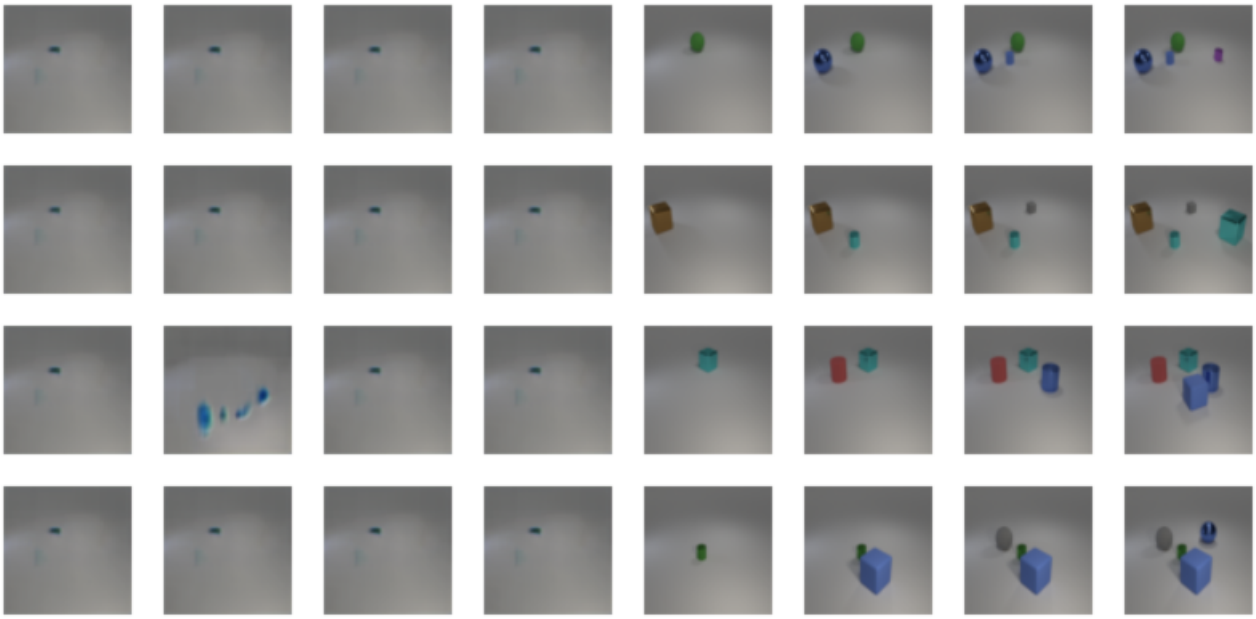
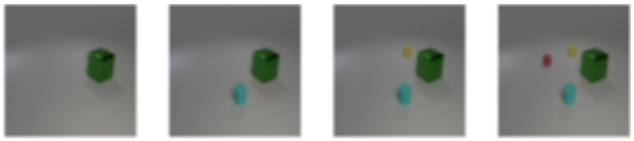
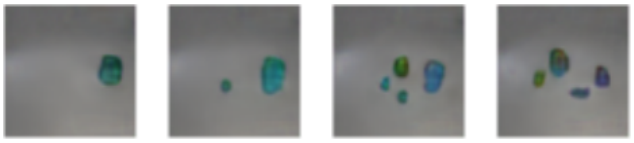


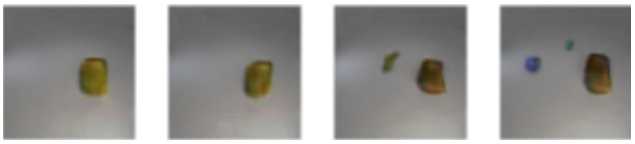
Figure 7.3.1: (left) Generated sequence (right) Ground truth
Training the Generator with faster LR than both Discriminators causes mode collapse.



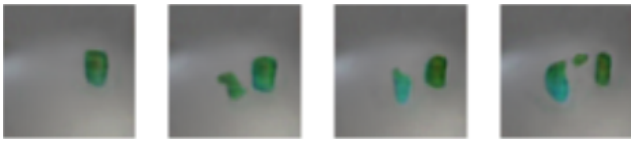
(a) Ground truth



(b) LR: 0.0001



(c) LR: 0.0002



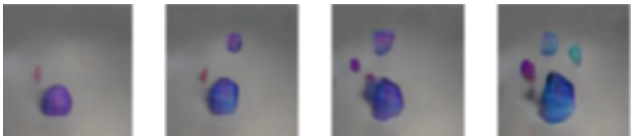
(d) LR: 0.0004

Figure 7.3.2: Increasing the learning rate of the Image Discriminator.

(a) Ground truth

(b) LR: 0.0001

(c) LR: 0.0002



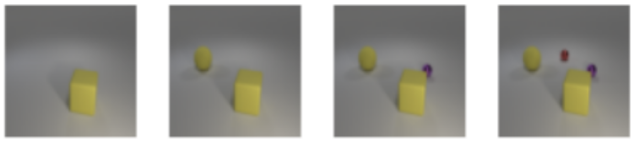
(d) LR: 0.0004

Figure 7.3.3: Increasing the learning rate of the Story Discriminator.

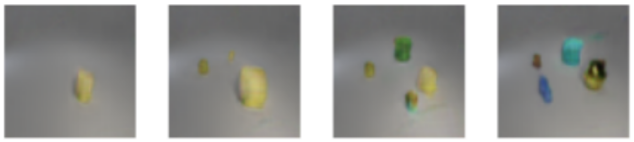
7.4 Experiment: Impartial Transformer Encoder

We maintain that the use of a transformer encoder is prescribed to encode context into the conditioning vectors of a sequence, but learning optimal mappings for the features of a given dataset in such a complicated task can be proven difficult. Context-imbued vectors are necessary both for the Generator and the Image Discriminator. We consider that the Story Discriminator is able to learn sufficient mappings for embeddings and images jointly without any immediate need for other processing, as it considers entire sequences in parallel.

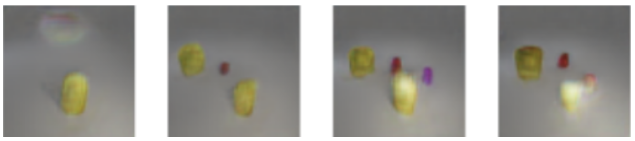
We explore the option of utilizing one "Impartial" transformer encoder, whose parameters are updated jointly by the generator and the image discriminator. We hypothesize such an encoder would learn a task-conducive representation for embedding sequences that simply encodes necessary context without giving an advantage to either adversary. As shown in figure 7.4.1 our intuitions proved to be correct. We also attempted to train the encoder to further receive gradients from the Story Discriminator, but found this addition to be "confusing" the encoder, to the point of learning completely mismatched representations of the context space.



(a) Ground truth



(b) Separate Transformer Encoder

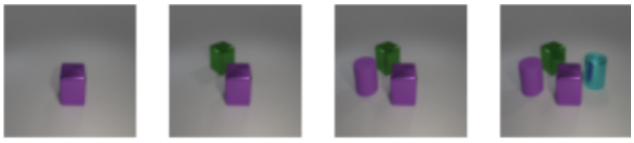


(c) Impartial Encoder (Generator and Image Discriminator gradients)

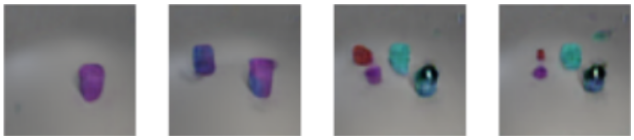
(d) Impartial Encoder (All network gradients)

Figure 7.4.1: Using an Impartial Transformer Encoder seem to give the best results, even though gradients from the Story Discriminator hurt performance.

Transformer parameters are in all cases: $d_{model} = 512$, $N_{heads} = 8$ attention and $N_{layers} = 6$, same as the original Transformer design proposed by Vashwani et al. [54] For all presented results we train the model for 120 epochs. For the rest of our experiments we use an Impartial Transformer Encoder for both Generator and Image Discriminator.

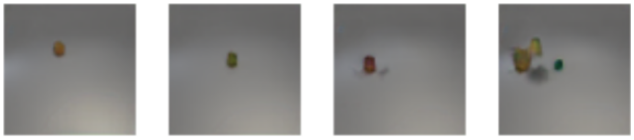


(a) Ground truth



(b) Separate Transformer Encoder

(c) Impartial Encoder (Generator and Image Discriminator gradients)



(d) Impartial Encoder (All network gradients)

Figure 7.4.2: Another Impartial Encoder example illustrating the same conclusions.

7.5 Experiment: Warmup Scheduler

Until now we have been training the Impartial Transformer Encoder using the Adam optimizer [25] with separate learning rates for the gradients flowing back from both Generator and Image Discriminator. We decay the learning rate by halving it every 20 epochs.

The original transformer paper [54] recommends a specific learning rate scheduling scheme to be used along with the Adam optimizer, considered optimal for training transformer-based sequence-to-sequence architectures. According to the scheme, the learning rate should first be increased linearly for a number of warmup steps and then decreased proportionally to the inverse square root of the number of total steps:

$$lrate = d_{model}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

A step is considered to be a single batch of data passing through the network. The learning rate change for $warmup_steps = 4000$ and $warmup_steps = 8000$ can be seen below:

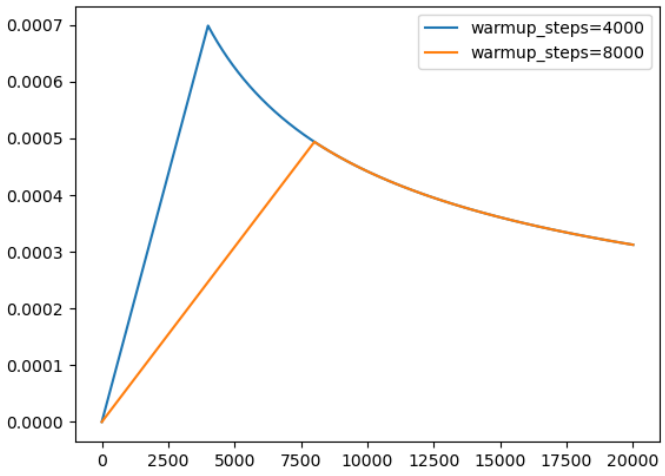
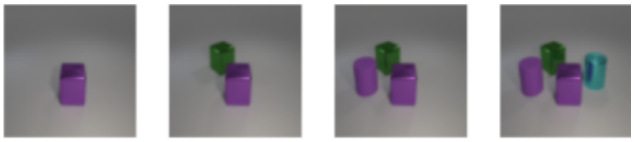


Figure 7.5.1: Learning rate against number of steps, for model dimension 512.

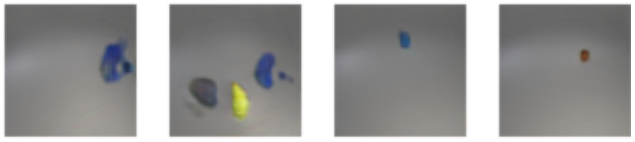
We compare the results of training with this optimization strategy compared to the one described in the above experiments.

We observe the scheduler fails to train the context encoder, resulting in mostly nonsensical representations that do not give meaningful results. We presume this is because the recommended optimizer only takes into account d_{model} and the number of warmup steps, thus forcing the learning rate to generally remain much higher than what the learning rates of the Adam optimizers in regular decay are. This causes the network to fail to converge.

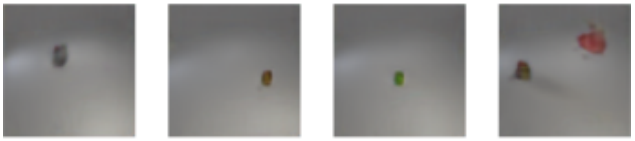


(a) Ground truth

(b) Learning rate reduced by half every 20 epochs.



(c) Recommended optimizer with 4000 warmup steps)



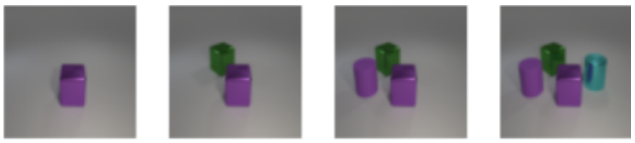
(d) Recommended optimizer with 8000 warmup steps)

Figure 7.5.2: Comparison of regular learning rate decay with the warmup scheduling proposed in the Transformer paper.

7.6 Experiment: Transformer Hyperparameters

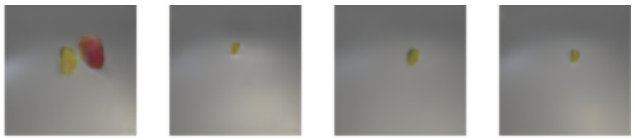
We experiment with different values for the parameters of the Transformer Encoder. Specifically we attempt to train the architecture in the figures of Chapter 6 without any image attention.

Our results show that the original Transformer with $d_{model} = 512, N_{heads} = 8, N_{layers} = 6$ is indeed optimal for our task. Reducing the number of heads proved to be immediately detrimental to performance, while wider or deeper transformers proved to have too much representational capacity to be trained successfully in 120 epochs:



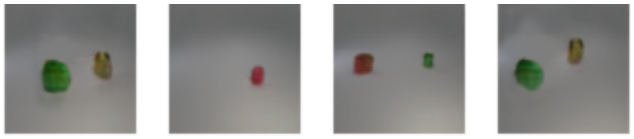
(a) Ground truth

(b) Dimension: 256, Heads: 4, Layers: 6

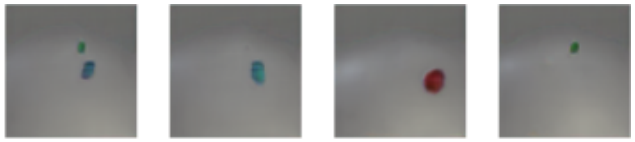


(c) Dimension: 512, Heads: 4, Layers: 6

(d) Dimension: 512, Heads: 8, Layers: 6



(e) Dimension: 512, Heads: 8, Layers: 8



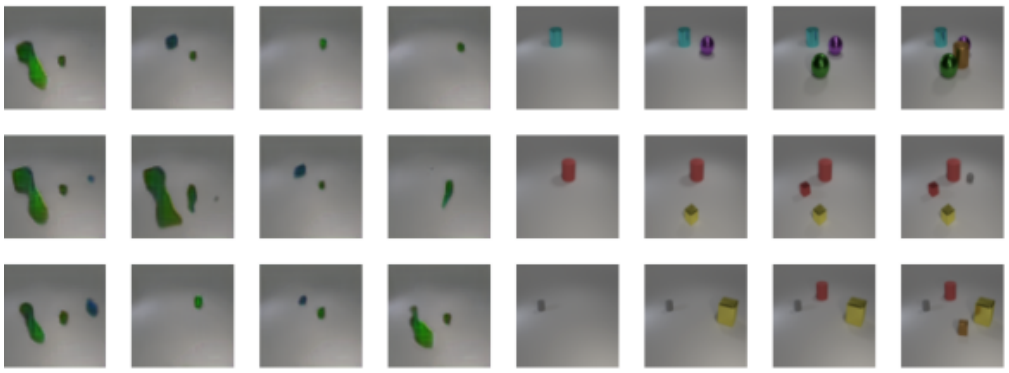
(f) Dimension: 768, Heads: 12, Layers: 6

Figure 7.6.1: Comparison of different hyperparameter settings for the Transformer Encoder.

7.7 Experiment: Attention Mechanisms

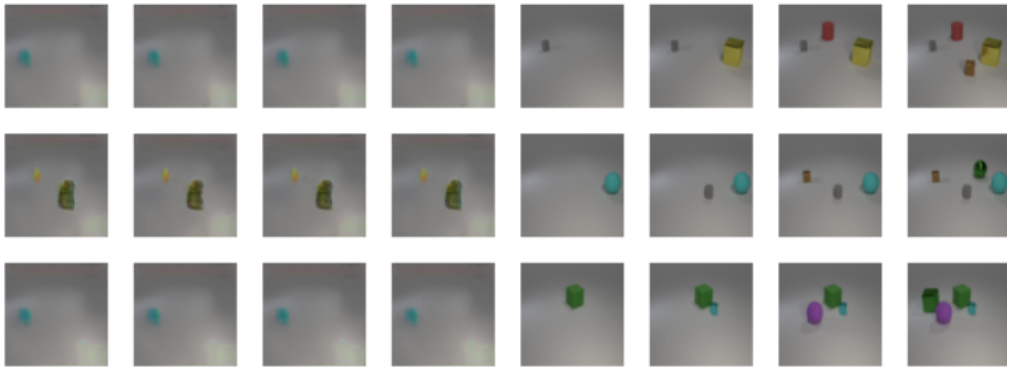
We attempt to use the proposed attention mechanisms as described in Section 6.1.5 in conjunction with the rest of our model. We tried using each attention mechanism successively in the Generator, starting from the Intra-image Attention and adding Inter-image and Encoder-Generator Attention afterwards. We also add Intra-image Attention to the two Discriminators.

We also experimented with the positioning of said mechanisms in different levels of the image scaling (upsampling/downsampling) process, testing whether or not the mechanisms would be successful when applied to higher or lower dimensional features. Despite our best efforts, all variations inevitably resulted in mode collapse:



(a) Intra-image Attention on features of size 32x32

(b) Intra-image Attention on features of size 16x16, with Inter-image attention in the Generator



(c) Intra-image Attention on features of size 32x32, with Inter-image and Encoder-Generator attention in the Generator

Figure 7.7.1: All variations of our attention mechanisms lead to mode collapse.

Chapter 8

Conclusion Future Directions

The results of our experimentation with the proposed architecture, despite not being immediately impressive, have proven the merits of our approach. A Transformer Encoder is a valid and efficient system to capture context information without the need for an RNN structure or special modifications to recurrent cells. Multiple variations of our model showed the ability to produce sequences of images that meaningfully display characteristics proposed in the input sentences. They also maintained consistency on almost all visual features except clear shape and texture. We consider this as evidence of the potency of attention mechanisms even in highly-dimensional sequence-to-sequence tasks, such as when the sequences consist of images.

The newly proposed attention mechanisms did not produce significant results for the time being, yet we believe this to non-exhaustive experimentation on our end and not an integral flaw of the approach. We maintain that Transformer Decoder-like attention - along with further hyperparameter tuning - is the key to eliminating the consistency errors introduced by the rest of the model.

Yet, we did observe a significant result in the introduction of an impartial Transformer Encoder. To our knowledge there are few, if any, adversarial architectures that utilize a "forking" approach where a common part of the network branches of into a Generator and a Discriminator jointly trained by receiving gradients from both of them during backpropagation. It is our opinion that this is a generally applicable formulation in Generative Adversarial Networks, where a "referee" network could take feedback from both adversaries to remap the input data into a representation that creates a level playing field without favoring either one.

We did not attempt to train the network on a more complicated dataset, like the Pororo-SV cartoon dataset [29] due to availability issues. We would endorse that this, or better, an improved version of this architecture be trained on such a dataset to observe a more substantial application of the same ideas on a harder modeling task. We would hope such an improved version would emerge, remedying the shortcomings that caused weak fidelity in the generated individual images, as well as mode collapse in the attention mechanism experiment.

As the task is closely related to the much faster improving and more well-researched topics of Text-to-Image and Sequence-to-Sequence generation, attempting to suggest architectural changes unique to the task beyond what we have already covered in this thesis would mostly have to predict improvements in either of these directions, while the flow of inspiration should most likely be the opposite in this case. As far as current trends go, we would like to also attempt to enforce greater text-image matching consistency by way of an external network such as the DAMSM proposed in AttnGAN [62].

Moreover, we identify two related ideas for future research, for the benefit of Story Visualization research in particular. First, since datasets made for this task are few and lack diversity we would like to propose a research direction towards developing a network designed to create feasible datasets: Starting off of one of many popular annotated video datasets [53, 44, 60] for tasks such as natural language video description, one could design a network similar in function to a video summarization architecture [55, 13] that would be trained to accept both a video and its corresponding multi-sentence text description, and select a sequence of frames from the video for each sentence. The task could be considered a conditional form of video summarization with some task-specific idiosyncracies: For example, if there are T sentences in the description, not only

should there be T output frames from the video, but those frames should be a subsequence of the video frames, meaning no selected frame should appear before the one following it in the output sequence. To that end the entire text sequence should be processed to produce the output and a conditional matching score should be considered for each frame-sentence pairing, given the frames chosen for other sentences.

There is also a lack of appropriate metrics for the evaluation of proposed Story Visualization models. We thus consider the possibility of researching a metric comparable to the Frechet Inception Distance [18] for image sequences. This metric should produce a final distance for two sets of image sequences, taking into account both feature similarity and sequential characteristics.

Finally, we would attempt to produce higher resolution images while maintaining fidelity by adding more rescaling blocks in the networks. Since residual blocks [17] are better at learning complex image feature representations and combat vanishing gradients in deep architectures, we consider it an advantage of this approach that a sequence of higher resolution images could be produced from a good implementation of our framework simply by increasing the number of upsamplings the features will go through.

To conclude, we believe this attention-based approach to Story Visualization to be the correct path towards new state-of-the-art models and possibly a great aid to the parent task of Text-to-Video generation. Despite this thesis not being the definitive attempt to solve the problem we hope to have paved the way to an optimal SV architecture, to be seen in future research.

Chapter 9

Bibliography

[1] Alammar, J. *The Illustrated Transformer*. 2018. URL:

[2] Altay, F. et al. *Preclinical Stage Alzheimer's Disease Detection Using Magnetic Resonance Image Scans*. 2020. arXiv: 2011.14139 [eess.IV].

[3] Arjovsky, M., Chintala, S., and Bottou, L. *Wasserstein GAN*. 2017. arXiv: 1701.07875 [stat.ML].

[4] Ba, J. L., Kiros, J. R., and Hinton, G. E. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML].

[5] Bahdanau, D., Cho, K., and Bengio, Y. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL].

[6] Brown, T. B. et al. "Language Models are Few-Shot Learners". In: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL:

[7] Cer, D. et al. *Universal Sentence Encoder*. 2018. arXiv: 1803.11175 [cs.CL].

[8] Cho, K. et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *CoRR* abs/1406.1078 (2014). arXiv: 1406.1078. URL:

[9] Cho, K. et al. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. 2014. arXiv: 1409.1259 [cs.CL].

[10] Copeland, B. J. "The Church-Turing Thesis". In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta. Summer 2020. Metaphysics Research Lab, Stanford University, 2020.

[11] Dai, Z. et al. "CoAtNet: Marrying Convolution and Attention for All Data Sizes". In: *CoRR* abs/2106.04803 (2021). arXiv: 2106.04803. URL:

[12] Devlin, J. et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL:

[13] Feng, L. et al. "Extractive Video Summarizer with Memory Augmented Neural Networks". In: *Proceedings of the 26th ACM International Conference on Multimedia*. MM '18. Seoul, Republic of Korea: Association for Computing Machinery, 2018, pp. 976–983. ISBN: 9781450356657. DOI: 10.1145/3240508.3240651. URL:

[14] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

[15] Goodfellow, I. et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL:

[16] Graves, A. *Generating Sequences With Recurrent Neural Networks*. 2014. arXiv: 1308.0850 [cs.NE].

[17] He, K. et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[18] Heusel, M. et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2018. arXiv: 1706.08500 [cs.LG].

[19] Hochreiter, S. and Schmidhuber, J. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: URL:

[20] Ioffe, S. and Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. arXiv: 1502.03167 [cs.LG].

[21] Johnson, J. et al. *CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning*. 2016. arXiv: 1612.06890 [cs.CV].

[22] Karras, T., Laine, S., and Aila, T. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *CoRR* abs/1812.04948 (2018). arXiv: 1812.04948. URL:

[23] Karras, T. et al. "Analyzing and Improving the Image Quality of StyleGAN". In: *CoRR* abs/1912.04958 (2019). arXiv: 1912.04958. URL:

[24] Kingma, D. P. and Welling, M. *Auto-Encoding Variational Bayes*. 2014. arXiv: 1312.6114 [stat.ML].

[25] Kingma, D. P. and Ba, J. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[26] Krizhevsky, A. "One weird trick for parallelizing convolutional neural networks". In: *CoRR* abs/1404.5997 (2014). arXiv: 1404.5997. URL:

[27] Li, C., Kong, L., and Zhou, Z. "Improved-StoryGAN for sequential images visualization". In: *Journal of Visual Communication and Image Representation* 73 (2020), p. 102956. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2020.102956. URL:

[28] Li, Y. et al. "Video Generation From Text". In: *CoRR* abs/1710.00421 (2017). arXiv: 1710.00421. URL:

[29] Li, Y. et al. *StoryGAN: A Sequential Conditional GAN for Story Visualization*. 2019. arXiv: 1812.02784 [cs.CV].

[30] Lim, J. H. and Ye, J. C. *Geometric GAN*. 2017. arXiv: 1705.02894 [stat.ML].

[31] Liu, Y. et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL:

[32] McCulloch, W. S. and Pitts, W. "A logical calculus of the ideas immanent in nervous activity". In: 5.4 (Dec. 1943), pp. 115–133. DOI: 10.1007/bf02478259. URL:

[33] Mirza, M. and Osindero, S. "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784 (2014). arXiv: 1411.1784. URL:

[34] Miyato, T. and Koyama, M. *cGANs with Projection Discriminator*. 2018. arXiv: 1802.05637 [cs.LG].

[35] Miyato, T. et al. *Spectral Normalization for Generative Adversarial Networks*. 2018. arXiv: 1802.05957 [cs.LG].

[36] Odena, A., Dumoulin, V., and Olah, C. "Deconvolution and Checkerboard Artifacts". In: *Distill* (2016). DOI: 10.23915/distill.00003. URL:

[37] Odena, A., Olah, C., and Shlens, J. *Conditional Image Synthesis With Auxiliary Classifier GANs*. 2017. arXiv: 1610.09585 [stat.ML].

[38] Odena, A. et al. *Is Generator Conditioning Causally Related to GAN Performance?* 2018. arXiv: 1802.08768 [stat.ML].

[39] Parmar, N. et al. "Image Transformer". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 4055–4064. URL:

[40] Radford, A., Metz, L., and Chintala, S. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. arXiv: 1511.06434 [cs.LG].

[41] Radford, A. et al. "Language Models are Unsupervised Multitask Learners". In: 2019.

[42] Reed, S. E. et al. "Generative Adversarial Text to Image Synthesis". In: *CoRR* abs/1605.05396 (2016). arXiv: 1605.05396. URL:

[43] Reed, S. E. et al. "Learning What and Where to Draw". In: *CoRR* abs/1610.02454 (2016). arXiv: 1610.02454. URL:

[44] Rohrbach, A. et al. *A Dataset for Movie Description*. 2015. arXiv: 1501.02530 [cs.CV].

[45] Saito, M. and Saito, S. "TGANv2: Efficient Training of Large Models for Video Generation with Multiple Subsampling Layers". In: *CoRR* abs/1811.09245 (2018). arXiv: 1811.09245. URL:

[46] Salimans, T. et al. *Improved Techniques for Training GANs*. 2016. arXiv: 1606.03498 [cs.LG].

[47] Simonyan, K. and Zisserman, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[48] Smolensky, P. "Information Processing in Dynamical Systems: Foundations of Harmony Theory". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, pp. 194–281. ISBN: 026268053X.

[49] Srivastava, A. et al. *MSRF-Net: A Multi-Scale Residual Fusion Network for Biomedical Image Segmentation*. 2021. arXiv: 2105.07451 [eess.IV].

[50] Srivastava, N. et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL:

[51] Sutskever, I., Vinyals, O., and Le, Q. V. "Sequence to Sequence Learning with Neural Networks". In: *CoRR* abs/1409.3215 (2014). arXiv: 1409.3215. URL:
[52] Szegedy, C. et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV].
[53] Torabi, A. et al. *Using Descriptive Video Services to Create a Large Data Source for Video Annotation Research*. 2015. arXiv: 1503.01070 [cs.CV].
[54] Vaswani, A. et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].
[55] Wang, J. et al. "Stacked Memory Network for Video Summarization". In: MM '19. Nice, France: Association for Computing Machinery, 2019, pp. 836–844. ISBN: 9781450368896. DOI: 10.1145/3343031.3350992. URL:
[56] Wang, X. et al. *Non-local Neural Networks*. 2018. arXiv: 1711.07971 [cs.CV].
[57] Wu, Y., Rosca, M., and Lillicrap, T. P. "Deep Compressed Sensing". In: *CoRR* abs/1905.06723 (2019). arXiv: 1905.06723. URL:
[58] Wu, Y. et al. "LOGAN: Latent Optimisation for Generative Adversarial Networks". In: *CoRR* abs/1912.00953 (2019). arXiv: 1912.00953. URL:
[59] Xu, H., Durme, B. V., and Murray, K. *BERT, mBERT, or BiBERT? A Study on Contextualized Embeddings for Neural Machine Translation*. 2021. arXiv: 2109.04588 [cs.CL].
[60] Xu, J. et al. "MSR-VTT: A Large Video Description Dataset for Bridging Video and Language". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
[61] Xu, M. et al. *End-to-End Semi-Supervised Object Detection with Soft Teacher*. 2021. arXiv: 2106.09018 [cs.CV].
[62] Xu, T. et al. "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks". In: *CVPR 2018*. 2018. URL:
[63] Zhang, H. et al. *StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks*. 2017. arXiv: 1612.03242 [cs.CV].
[64] Zhang, H. et al. *StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks*. 2018. arXiv: 1710.10916 [cs.CV].
[65] Zhang, H. et al. *Self-Attention Generative Adversarial Networks*. 2019. arXiv: 1805.08318 [stat.ML].
[66] Zhou, C. et al. "One-pass Multi-task Networks with Cross-task Guided Attention for Brain Tumor Segmentation". In: *CoRR* abs/1906.01796 (2019). arXiv: 1906.01796. URL:
[67] Zhu, M. et al. "DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis". In: *CoRR* abs/1904.01310 (2019). arXiv: 1904.01310. URL: