National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

Investigating Optimization Techniques for Multimodal Neural Networks

Diploma Thesis

of

IOANNA P. KAFFEZA

Supervisor: Alexandros Potamianos
Associate Professor, NTUA

Athens, February 2025

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

Investigating Optimization Techniques for Multimodal Neural Networks

Diploma Thesis

of

IOANNA P. KAFFEZA

Supervisor: Alexandros Potamianos
Associate Professor, NTUA

Approved by the examination committee on 25th February 2025.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
Alexandros Potamianos	Stefanos Kollias	Constantinos Tzafestas
Associate Professor, NTUA	Professor Emeritus, NTUA	Associate Professor, NTUA

Athens, February 2025

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

.....

IOANNA P. KAFFEZA

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών, ΕΜΠ
Diploma in Electrical and Computer Engineering, NTUA

25th February 2025

Περίληψη

Η πολυτροπική μάθηση έχει προσελκύσει σημαντικό ενδιαφέρον στην ανάλυση συναισθήματος, ωστόσο, τα πολυτροπικά μοντέλα συχνά εμφανίζουν υποδεέστερη απόδοση σε σύγκριση με τα μονοτροπικά—ένα αντιφατικό φαινόμενο. Οι ανισόρροπες δυναμικές μάθησης, όπου ορισμένες μορφές δεδομένων κυριαρχούν στη διαδικασία εκπαίδευσης, ενώ άλλες παραμένουν αναξιοποίητες, οδηγούν σε μη βέλτιστη απόδοση του μοντέλου.

Η παρούσα διπλωματική διερευνά την επίδραση των τεχνικών βελτιστοποίησης σε πολυτροπικά νευρωνικά δίκτυα, εστιάζοντας στο πώς διαφορετικές στρατηγικές επηρεάζουν τις ανισόρροπες δυναμικές μάθησης στην ανάλυση συναισθήματος.

Αξιολογούμε δύο κατηγορίες τεχνικών βελτιστοποίησης στα σύνολα δεδομένων CMU-MOSI και CMU-MOSEI. Οι τεχνικές OGM-GE, AGM εφαρμόζουν άμεσες προσαρμογές των παραγώγων κατά την ανάστροφη διάδοση, στοχεύοντας σε ισορροπημένη συνεισφορά από κάθε μορφή δεδομένων. Αντίθετα, οι τεχνικές PMR και ReconBoost βασίζονται στην εξισορρόπηση μέσω συνάρτησης πολλαπλών απωλειών. Το PMR εισάγει ένα σχήμα ποινής και ενίσχυσης, ενώ το ReconBoost ενσωματώνει ένα εναλλασσόμενο μαθησιακό πρότυπο. Επιπλέον, αξιολογούμε αρχιτεκτονικές επιλογές, όπως ο optimizer, το batch size και η χρήση συνόλου ανάπτυξης για αμερόληπτους υπολογισμούς.

Παρόλο που οι τεχνικές εξισορρόπησης μέσω παραγώγων και πολλαπλών απωλειών συμβάλλουν στη βελτίωση της ισορροπίας μάθησης, καμία δεν επιλύει πλήρως το πρόβλημα της ανισόρροπης εκπαίδευσης. Καθιερωμένα βασικά μοντέλα, όπως το Late Concatenation και το Uni-Pre Finetuned, διατηρούν την υπεροχή τους όσον αφορά την ακρίβεια ταξινόμησης. Η χρήση ενός συνόλου ανάπτυξης αποδεικνύεται ευεργετική για τη σταθερότητα και την αποφυγή μεροληψίας, ενώ ο Adam αναδεικνύεται ως ο πιο αποτελεσματικός optimizer.

Παρά αυτές τις εξελίξεις, η βελτιστοποίηση πολυτροπικών μοντέλων παραμένει μια ανοιχτή πρόκληση. Οι δυναμικές τεχνικές βελτιστοποίησης ενισχύουν την ισορροπία, αλλά όχι τη συνολική απόδοση, υπογραμμίζοντας την ανάγκη για πιο προσαρμοστικές στη δομή των δεδομένων εισόδου τεχνικές. Τα παρόντα αποτελέσματα συμβάλλουν στην καλύτερη κατανόηση των δυναμικών της πολυτροπικής μάθησης, προσφέροντας πολύτιμες προοπτικές για μελλοντικές βελτιώσεις στην πολυτροπική ανάλυση συναισθήματος.

Λέξεις Κλειδιά

Μηχανική Μάθηση, Πολυτροπικά Νευρωνικά Δίκτυα, Ανάλυση Συναισθήματος, Αλγόριθμος Ανάστροφης Διάδοσης, Ανισόρροπη Εκμάθηση, Τεχνικές Βελτιστοποίησης

Abstract

Multimodal learning has gained significant attention in sentiment analysis, yet multimodal models often have degraded performance compared to their unimodal counterparts—a counterintuitive phenomenon. Imbalanced learning dynamics, where certain modalities dominate the learning process while others remain underutilized, lead to suboptimal model performance.

This thesis investigates the impact of optimization techniques on multimodal neural networks, focusing on how different strategies influence unbalanced learning dynamics in sentiment analysis.

We evaluate two categories of optimization techniques on the CMU-MOSI and CMU-MOSEI datasets for sentiment classification. Methods of OGM-GE and AGM, apply direct gradient adjustments during backpropagation to ensure balanced contributions from each modality. On the other hand, PMR and ReconBoost focuses on a multi-loss approach. PMR introduces a penalty-boosting loss scheme, while ReconBoost incorporates an alternating learning paradigm. Additionally, we assess architectural choices, including optimizer selection, batch size, and the use of a development set for unbiased auxiliary calculations in dynamic adjustments.

While gradient-based and multi-loss approaches help balance learning dynamics, no single method fully resolves modality imbalance in our tasks. Established baselines, such as Late Concatenation and Uni-Pre Finetuned, remain superior in accuracy. The use of a development set enhances stability and reduces bias, while Adam proves to be the most effective optimizer.

Despite these advancements, multimodal optimization remains an open challenge. While dynamic optimization techniques improve modality balance, they do not consistently enhance overall performance, highlighting the need for more adaptive and modality-aware optimization strategies. These findings provide a deeper understanding of multimodal learning dynamics, offering valuable insights for future advancements in multimodal sentiment analysis.

Keywords

Machine Learning, Multimodal Neural Networks, Sentiment Analysis, Backpropagation Algorithm, Imbalanced Learning, Optimization Techniques

στους αγαπημένους μου γονείς,
Παναγιώτη και Σοφία.

Ευχαριστίες

Θα ήθελα, πρωτίστως, να ευχαριστήσω τον καθηγητή, Αλέξενδρο Ποταμιάνο, για την επίβλεψη της διπλωματικής μου εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο Εργαστήριο Επεξεργασίας Λόγου και Γλώσσας. Η επιστημονική του καθοδήγηση υπήρξε καθοριστικός παράγοντας για την επιτυχή ολοκλήρωση της παρούσας εργσίας, καθώς με την εμπειρία και τις πολύτιμες συμβουλές του συνέβαλε καθοριστικά στη διαμόρφωση των βασικών κατευθύνσεων κατά τη διάρκεια της έρευνας. Θα ήθελα να ευχαριστήσω ακόμα τον υποψήφιο διδάκτορα, Ευθύμιο Γεωργίου, για την πολύτιμη καθοδήγησή του, την επιστημονική του κατάρτιση και την συνεχή υποστήριξή του καθ'όλη τη διάρκεια αυτής της διπλωματικής. Ένα μεγάλο ευχαριστώ οφείλω και στην οικογένειά μου για την αμέριστη υποστήριξη και ηθική συμπαράσταση που μου προσέφεραν καθ' όλη τη διάρκεια των σπουδών μου. Οι γονείς μου στάθηκαν δίπλα μου σε κάθε δυσκολία και κάθε επιτυχία. Ιδιαίτερη αναφορά αξίζει στον θείο μου Ανδρέα, ο οποίος υπήρξε πολύτιμος αρωγός στην προετοιμασία μου για την εισαγωγή στο Πολυτεχνείο. Τέλος, ευχαριστώ θερμά τους φίλους μου και όλους τους αγαπημένους μου ανθρώπους για την κατανόηση, την ενθάρρυνση και την στήριξή τους σε κάθε μου βήμα. Η παρουσία τους υπήρξε ανεκτίμητη πηγή δύναμης και αισιοδοξίας, δίνοντας νόημα σε αυτή τη σημαντική διαδρομή της ζωής μου. Η τελευταία χρονιά υπήρξε καθοριστική για εμένα, και τους ευχαριστώ από καρδιάς που στάθηκαν όλοι δίπλα μου.

Αθήνα, Φεβρουάριος 2025

Ιωάννα Καφφέζα

Contents

List of Figures

List of Tables

Chapter 0

Εκτεταμένη Ελληνική Περίληψη

0.1 Εισαγωγή

Το πεδίο της Τεχνητής Νοημοσύνης έχει σημειώσει αξιοσημείωτη πρόοδο τις τελευταίες δεκαετίες. Στον πυρήνα αυτής της προόδου βρίσκεται η Μηχανική Μάθηση, η οποία επιτρέπει στα συστήματα να μαθαίνουν από δεδομένα και να πραγματοποιούν προβλέψεις ή να λαμβάνουν αποφάσεις χωρίς να απαιτείται ρητός προγραμματισμός. Αυτές οι εξελίξεις βασίζονται σε τεχνητά νευρωνικά δίκτυα, σχεδιασμένα να αναγνωρίζουν πρότυπα στα δεδομένα και να γενικεύουν σε νέα, άγνωστα δεδομένα. Καθώς η Μηχανική Μάθηση συνεχίζει να εξελίσσεται, η ικανότητα επεξεργασίας και συνδυασμού ετερογενών πηγών πληροφορίας καθίσταται ολοένα και πιο απαραίτητη, σηματοδοτώντας την στροφή στην Πολυτροπική Μηχανική Μάθηση. Αυτή η προσέγγιση επιτρέπει μια πιο ολοκληρωμένη κατανόηση σύνθετων προβλημάτων, αξιοποιώντας τη συμπληρωματική πληροφορία που παρέχουν οι διαφορετικές τροπικότητες.

Ωστόσο, η ενοποίηση πολυτροπικών δεδομένων εισάγει νέες προκλήσεις, όπως η αποτελεσματική διαχείριση της μοναδικής φύσης κάθε τροπικότητας σε συγχρονισμένο περιβάλλον. Ο διαφορετικός ρυθμός εκμάθησης κάθε τροπικότητας σε ένα πολυτροπικό νευρωνικό δίκτυο δημιουργεί ανισορροπία στην μαθησιακή διαδικασία, με αποτέλεσμα μοντέλα που βασίζονται υπέρμετρα σε κυρίαρχες τροπικότητες και αμελούν τις υπόλοιπες. Η ανάγκη για εξειδικευμένες τεχνικές βελτιστοποίησης που λαμβάνουν υπόψη τη μοναδική δυναμική μάθησης κάθε τροπικότητας και ενισχύουν την ισορροπημένη ενσωμάτωση των τροπικοτήτων κρίνεται πλέον απαραίτητη.

Εμπνευσμένοι από τις προκλήσεις που παρουσιάζει η ανισορροπία στην πολυτροπική μάθηση, η παρούσα διπλωματική ερευνά δυναμικές τεχνικές βελτιστοποίησης πολυτροπικών νευρωνικών δικτύων στο πεδίο της ανάλυσης συναισθημάτων, παρέχοντας μια ολοκληρώμενη αξιολόγηση αλγορίθμων εξισορρόπησης συνεισφοράς στη μάθηση. Η εργασία αυτή συνεισφέρει στους εξής τομείς:

- Διερεύνηση δυναμικών μεθόδων βελτιστοποίησης: Παρουσιάζεται μια συνολική ανάλυση τεσσάρων δυναμικών τεχνικών βελτιστοποίησης—OGM-GE [26] και AGM [27] που βασίζονται σε άμεσες τροποποιήσεις των παραγώγων οπισθοδιάδοσης, καθώς και PMR [20] και ReconBoost [21] που ακολουθούν προσέγγιση πολλαπλών συναρτήσεων απωλειών—σχεδιασμένων ειδικά για την αντιμετώπιση της ανισορροπίας στις πολυτροπικές διαδικασίες εκμάθησης συναισθημάτων.

- Αξιολόγηση σε διαφορετικά σενάρια ανισορροπίας: Πραγματοποιούνται πειράματα σε τρία διαφορετικά σενάρια ανισορροπίας, όπου εξετάζεται η περίπτωση μιας κυρίαρχης και μιας ασθενέστερης τροπικότητας, δύο ασθενέστερων τροπικοτήτων, καθώς και τριών τροπικοτήτων με διαφορετικά επίπεδα συνεισφοράς. Η ανάλυση αυτή παρέχει μια λεπτομερή κατανόηση της συμπεριφοράς των μεθόδων υπό συνθήκες τροπικοτήτων με διαφορετική ισχύ.
- Ανάλυση καθοριστικών παραγόντων: Διερεύνηση παραγόντων που επηρεάζουν την αποτελεσματικότητα των δυναμικών μεθόδων βελτιστοποίησης, όπως η επιλογή του optimizer, καθώς και προτάσεις για τη βελτίωση της εφαρμοσιμότητας των αλγορίθμων, συμπεριλαμβανομένης της χρήσης βοηθητικού συνόλου ανάπτυξης για αμερόληπτους βοηθητικούς υπολογισμούς.

Τα ευρήματα αυτής της εργασίας αποσκοπούν στην παροχή πολύτιμων γνώσεων για τη δυναμική βελτιστοποίηση των πολυτροπικών νευρωνικών δικτύων υπό το πρίσμα της ανάλυσης συναισθημάτων, ενώ παράλληλα συμβάλλουν στην ευρύτερη κατανόηση της ανισορροπίας τροπικοτήτων ως βασικής πρόκλησης στην έρευνα της πολυτροπικής μάθησης.

0.2 Μηχανική Μάθηση: Ο Αλγόριθμος Ανάδρασης

Ο αλγόριθμος ανάδρασης (backpropagation), σε συνδυασμό με αλγόριθμο (gradient descent) και τη συνάρτηση απώλειας, αποτελεί τον πυρήνα της διαδικασίας βελτιστοποίησης των νευρωνικών δικτύων.

Η συνάρτηση κόστους: Η συνάρτηση απώλειας εκφράζει τη διαφορά μεταξύ της προβλεπόμενης εξόδου ενός μοντέλου και της πραγματικής τιμής-στόχου. Η απώλεια ποσοτικοποιεί το σφάλμα και καθοδηγεί τη διαδικασία βελτιστοποίησης των παραμέτρων του μοντέλου ώστε να ελαχιστοποιηθεί το σφάλμα. Η συνάρτηση απώλειας συγκρίνει την έξοδο y_t με την αντίστοιχη τιμή-στόχο $\hat{y}_t$ στη χρονική στιγμή t και ορίζεται ως:

$$L(y, \hat{y}) = \sum_{t=1}^{T} L(y_t, \hat{y}_t) \tag{1}$$

Η απώλεια στη χρονική στιγμή t εκφράζεται ως $L(y_t, \hat{y}_t)$, και το T αντιστοιχεί στο συνολικό αριθμό των χρονικών στιγμών. Η εξίσωση αυτή αναπαριστά το συνολικό άθροισμα των απωλειών σε κάθε χρονική στιγμή. Η επιλογή της συνάρτησης απώλειας παίζει κρίσιμο ρόλο στη διαδικασία μάθησης και θεωρείται εξαρτώμενη από το εκάστοτε πρόβλημα.

Ο αλγόριθμος ανάδρασης: Όπως περιγράφεται στο [28], κατά το βήμα της προώθησης (forward pass), τα δεδομένα εισόδου διέρχονται από κάθε επίπεδο του νευρωνικού δικτύου. Κάθε επίπεδο υπολογίζει το σταθμισμένο άθροισμα των εισόδων του και εφαρμόζει τη συνάρτηση ενεργοποίησης. Στη συνέχεια, μεταβιβάζει το αποτέλεσμα στο επόμενο επίπεδο, καταλήγοντας τελικά στην παραγωγή της εξόδου. Υπολογίζεται στη συνέχεια το σφάλμα, το οποίο βασίζεται στη διαφορά μεταξύ της προβλεπόμενης εξόδου και της πραγματικής τιμής-στόχου. Στο βήμα της οπισθοδιάδοσης, ο αλγόριθμος ξεκινά από το επίπεδο εξόδου και

κινείται προς τα πίσω προς το επίπεδο εισόδου, ενημερώνοντας συστηματικά τα βάρη με σκοπό τη μείωση του σφάλματος, ακολουθώντας τον αλγόριθμο (gradient descent). Σε κάθε επίπεδο, υπολογίζεται η παράγωγος του σφάλματος ως προς τα βάρη ώστε να προσδιοριστεί η συμβολή κάθε βάρους στο συνολικό σφάλμα. Η προσαρμογή των βαρών γίνεται με την εφαρμογή του κανόνα της αλυσίδας, υπολογίζοντας τον τρόπο με τον οποίο το σφάλμα της εξόδου διαδίδεται σε κάθε επίπεδο του δικτύου. Συγκεκριμένα, για κάθε βάρος, η κλίση του σφάλματος υπολογίζεται ως το γινόμενο των μερικών παραγώγων της συνάρτησης απώλειας ως προς την έξοδο, της εξόδου ως προς την ενεργοποίηση, και της ενεργοποίησης ως προς τα βάρη. Μαθηματικά, για ένα συγκεκριμένο βάρος w, η κλίση δίνεται από την εξίσωση:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial w} \tag{2}$$

όπου L είναι η συνάρτηση απώλειας, y είναι η έξοδος του δικτύου, z είναι το σταθμισμένο άθροισμα των εισόδων προς τη συνάρτηση ενεργοποίησης, και w είναι το βάρος. Η εφαρμογή του κανόνα της αλυσίδας διασφαλίζει ότι η επίδραση των βαρών σε κάθε επίπεδο λαμβάνεται σωστά υπόψη στον υπολογισμό των κλίσεων. Στην οπισθοδιάδοση, τα (biases) ενημερώνονται παράλληλα με τα βάρη, επιτρέποντας στο μοντέλο να προσαρμόσει το κατώφλι ενεργοποίησης των νευρώνων και να μετατοπίσει την έξοδο ώστε να ταιριάζει καλύτερα στα δεδομένα. Η παρακάτω εξίσωση περιγράφει τον κανόνα ενημέρωσης των βαρών του αλγορίθμου gradient descent, ο οποίος αποτελεί θεμελιώδες στοιχείο της διαδικασίας οπισθοδιάδοσης. Συγκεκριμένα, προσαρμόζει τα βάρη Δw κλιμακώνοντας την κλίση της συνάρτησης απώλειας $\frac{\partial L}{\partial w}$ με έναν ρυθμό μάθησης η:

$$\Delta w_{ij} = -\eta \frac{\partial L}{\partial w_{ij}} \tag{3}$$

Η επαναλαμβανόμενη αυτή διαδικασία ελαχιστοποιεί το σφάλμα, καθοδηγώντας το δίκτυο προς ένα σύνολο βαρών που μειώνουν τη συνολική απώλεια. Στο πλαίσιο της οπισθοδιάδοσης, ο αλγόριθμος gradient descent λειτουργεί ως ο μηχανισμός βελτιστοποίησης που ενημερώνει τα βάρη σε κάθε επανάληψη, βασιζόμενος στις κλίσεις των σφαλμάτων που διαδίδονται προς τα πίσω στο δίκτυο. Η διαδικασία της προώθησης και της οπισθοδιάδοσης εκτελείται επανειλημμένα για πολλές εποχές, με τα βάρη να ενημερώνονται σταδιακά, μέχρι το σφάλμα να συγκλίνει σε ένα ελάχιστο επίπεδο. Η πιο σημαντική υπερπαράμετρος είναι ο ρυθμός μάθησης η, καθώς ελέγχει τον βαθμό προσαρμογής των παραμέτρων του μοντέλου σε σχέση με την κλίση της απώλειας. Συνεπώς, η επιλογή του ρυθμού μάθησης είναι κρίσιμη για τη σταθερότητα και την αποτελεσματικότητα της εκπαίδευσης του δικτύου.

0.3 Εκπαίδευση Πολυτροπικών Νευρωνικών Δικτύων

Οι θεμελιώδεις αρχές της οπισθοδιάδοσης, της βελτιστοποίησης και της γενίκευσης σε νέα δεδομένα παραμένουν συνεπείς με αυτές που ισχύουν στα μονοτροπικά συστήματα. Ωστόσο, τα πολυτροπικά δίκτυα εισάγουν μοναδικές προκλήσεις λόγω της ετερογένειας και της αλληλεξάρτησης των διαφορετικών τρόπων δεδομένων. Η οπισθοδιάδοση παραμένει το θεμέλιο της εκπαίδευσης πολυτροπικών νευρωνικών δικτύων, επιτρέποντας στο μοντέλο να ελαχιστοποιήσει τη συνάρτηση απώλειας μαθαίνοντας αποδοτικά τόσο ειδικές ανά τρόπο

αναπαραστάσεις όσο και κοινές αναπαραστάσεις. Σε περιπτώσεις όπου χρησιμοποιούνται υποδίκτυα εξειδικευμένα ανά τροπικότητα, οι ενημερώσεις των παραγώγων (gradients) πρέπει να ρέουν όχι μόνο μέσω των κοινών επιπέδων σύντηξης αλλά και μέσω κάθε υποδικτύου ανεξάρτητα. Αυτό εξασφαλίζει ότι τα ειδικά χαρακτηριστικά κάθε τρόπου βελτιστοποιούνται, ενώ τα επίπεδα συνένωσης καταγράφουν διαδράσεις μεταξύ των τρόπων. Με την ενημέρωση των βαρών σε όλα τα επίπεδα, η οπισθοδιάδοση διευκολύνει τη συνεκπαίδευση των εξειδικευμένων και κοινών στοιχείων του μοντέλου, ενισχύοντας την ικανότητά του να εξάγει συμπληρωματικές πληροφορίες και να βελτιώνει τη συνολική του απόδοση. Επιπλέον, η γενίκευση παραμένει κρίσιμη στην πολυτροπική μηχανική μάθηση, καθώς καθορίζει πόσο καλά μπορεί να αποδώσει ένα μοντέλο που έχει εκπαιδευτεί σε ένα συγκεκριμένο σύνολο δεδομένων όταν εφαρμοστεί σε άγνωστα δεδομένα. Η απουσία ή ο θόρυβος δεδομένων, που αποτελούν κοινό φαινόμενο στα πραγματικά πολυτροπικά συστήματα, σε συνδυασμό με την εγγενή ετερογένεια των μορφών δεδομένων, μπορούν να επηρεάσουν αρνητικά τη γενίκευση του μοντέλου.

0.4 Ορισμός του Προβλήματος: Ανισόρροπη Εκμάθηση Πολυτροπικών Δεδομένων

Το πρόβλημα της ανισορροπίας μεταξύ τροπικοτήτων μελετήθηκε συστηματικά για πρώτη φορά από τους Wang et al. [22] στην εργασία τους με τίτλο *"What Makes Training Multimodal Classification Networks Hard?"*. Οι συγγραφείς εντόπισαν δύο βασικούς παράγοντες που ευθύνονται για την υποβάθμιση της απόδοσης των πολυτροπικών δικτύων σε σύγκριση με τα αντίστοιχα μονοτροπικά. Πρώτον, η αυξημένη χωρητικότητα των πολυτροπικών δικτύων, δηλαδή ο μεγαλύτερος αριθμός παραμέτρων και οι πολύπλοκες αρχιτεκτονικές που απαιτούνται για την επεξεργασία ολοκληρωμένων πληροφοριών από πολλαπλές τροπικότητες, συχνά οδηγούν σε υπερπροσαρμογή (overfitting). Δεύτερον, κάθε τροπικότητα τείνει να υπερπροσαρμόζεται ή να γενικεύει με διαφορετικό ρυθμό από τις υπόλοιπες, λόγω διαφορών στην πολυπλοκότητα και στον όγκο της πληροφορίας που παρέχει. Επεκτείνοντας αυτή τη θεμελίωση, οι Wu et al. [23] διατύπωσαν την Υπόθεση Greedy Learner, υπογραμμίζοντας ότι τα πολυτροπικά μοντέλα φυσικά δίνουν προτεραιότητα στις τροπικότητες που μαθαίνουν ταχύτερα, παραμελώντας εκείνες που μαθαίνουν με πιο αργό ρυθμό. Οι Huang et al. [29] εισήγαγαν την έννοια του modality competition για να εξηγήσουν γιατί τα πολυτροπικά δίκτυα αποδίδουν χειρότερα σε σχέση με τα μονοτροπικά, ιδιαίτερα όταν εκπαιδεύονται με προσέγγιση late-fusion concatenation. Διαπίστωσαν ότι κατά την εκπαίδευση, οι τροπικότητες ανταγωνίζονται για τη μάθηση των αναπαραστάσεων, με αποτέλεσμα μόνο ένα υποσύνολο των τροπικοτήτων, συνήθως οι κυρίαρχες, να μαθαίνουν αποτελεσματικά. Αυτό το φαινόμενο προκύπτει λόγω διαφορών στη δυναμική εκμάθησης χαρακτηριστικών και της τυχαίας αρχικοποίησης των παραμέτρων του δικτύου, που ευνοούν δυσανάλογα ορισμένες τροπικότητες. Επιπλέον, οι ασθενέστερες τροπικότητες, ειδικά εκείνες που παρουσιάζουν ανεπαρκή δομή δεδομένων, συχνά παραμελούνται, οδηγώντας σε υποβαθμισμένες αναπαραστάσεις χαρακτηριστικών.

Για την ανάλυση της ανισορροπίας μεταξύ τροπικοτήτων, θεωρούμε ένα πολυτροπικό

σύνολο εκπαίδευσης δεδομένων $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ με N δείγματα, όπου κάθε δείγμα αποτελείται από χαρακτηριστικά $x_i = \{m_i^k\}_{k=1}^{M}$ από M διαφορετικές τροπικότητες και μία ετικέτα y_i. Ο στόχος της πολυτροπικής μάθησης είναι η ελαχιστοποίηση της συνάρτησης απωλειών:

$$L(S(\{F_k(x)\}_{k=1}^{M}), y) = \frac{1}{N}\sum_{i=1}^{N} \ell(S(\{F_k(\partial_k; m_i^k)\}_{k=1}^{M}), y_i), \tag{4}$$

όπου ℓ είναι η συνάρτηση απώλειας. Κατά την οπισθοδιάδοση, η ενημέρωση των παραμέτρων κάθε τροπικότητας k πραγματοποιείται μέσω:

$$\nabla_{\partial_k} L(\Phi^M(x), y) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\partial \ell(\Phi^M(x_i), y_i)}{\partial \Phi^M(x_i)} \cdot \frac{\partial \Phi^M(x_i)}{\partial F_k(\partial_k)} \cdot \frac{\partial F_k(\partial_k)}{\partial \partial_k}\right). \tag{5}$$

Ο όρος $\frac{\partial \Phi^M(x_i)}{\partial F_k(\partial_k)}$ απλοποιείται σε ένα μοναδιαίο πίνακα για το αντίστοιχο μπλοκ τροπικότητας. Ωστόσο, η κοινός όρος $\frac{\partial \ell(\Phi^M(x_i), y_i)}{\partial \Phi^M(x_i)}$ διαχέεται σε όλα τα χαρακτηριστικά, γεγονός που οδηγεί σε μεγαλύτερη ενίσχυση της κυρίαρχης τροπικότητας εις βάρος των ασθενέστερων. Σε ένα μοντέλο συνένωσης, η διαδικασία βελτιστοποίησης προτιμά τροπικότητες των οποίων τα χαρακτηριστικά ευθυγραμμίζονται ισχυρά με το κοινό gradient. Το gradient μιας τροπικότητας k ευθυγραμμίζεται σταθερά με την κοινό αν ισχύει:

$$\frac{\partial \ell(\Phi^M(x_i), y_i)}{\partial \Phi^M(x_i)} \cdot F_k(\partial_k) \gg \frac{\partial \ell(\Phi^M(x_i), y_i)}{\partial \Phi^M(x_i)} \cdot F_j(\partial_j), \quad \forall j \neq k. \tag{6}$$

Όταν η παραπάνω συνθήκη ισχύει για μια κυρίαρχη τροπικότητα k, τα χαρακτηριστικά της λαμβάνουν μεγαλύτερες ενημερώσεις κατά την οπισθοδιάδοση, ενώ οι ασθενέστερες τροπικότητες παραμένουν υποεκπαιδευμένες. Για τις ασθενέστερες τροπικότητες, η συμβολή τους στο κοινό gradient είναι μικρή, οδηγώντας σε περιορισμένες ενημερώσεις των παραμέτρων τους. Αυτό το φαινόμενο προκαλεί ανισόρροπη εκπαίδευση, καθώς οι κυρίαρχες τροπικότητες υπαγορεύουν την κατεύθυνση της βελτιστοποίησης και μειώνουν την ικανότητα των ασθενέστερων τροπικοτήτων να συνεισφέρουν αποτελεσματικά στη μάθηση του μοντέλου. Η ασυμφωνία κατεύθυνσης μεταξύ των βαθμίδων των τροπικοτήτων επιδεινώνει περαιτέρω το πρόβλημα [20] [22] [23]. Όταν οι βαθμίδες μιας ασθενέστερης τροπικότητας είναι ορθογώνιες ή αντικρουόμενες προς αυτές των ισχυρότερων τροπικοτήτων, οι ενημερώσεις της καθίστανται μη αποδοτικές. Αυτό οδηγεί στη δημιουργία μιας συνένωσης χαρακτηριστικών $\Phi^M(x)$ που είναι έντονα μεροληπτική υπέρ των κυρίαρχων τροπικοτήτων, περιορίζοντας τη γενίκευση και την αξιοποίηση της πολυτροπικής πληροφορίας.

0.5 Εξισορρόπιση Πολυτροπικής Εκπαίδευσης μέσω Τεχνικών Βελτιστοποίησης

Υπάρχουν ποικίλες τεχνικές βελτιστοποίησης στη βιβλιογραφία για την αντιμετώπιση του προβλήματος της ανισορροπίας μεταξύ τροπικοτήτων. Ορισμένες μέθοδοι βασίζονται στην άμεση επίδραση στις παραγώγους (gradients) προκειμένου να ενισχύσουν την ασθενέστερη τροπικότητα κατά τη διαδικασία μάθησης [22] [23] [26] [27] [30] [31] [32]. Άλλες επικεντρώνονται στην δυναμική βελτιστοποίηση της συνάρτησης απώλειας [20] [12] [33], ενώ

Μέθοδος	Τεχνική	Συχνότητα	Ενεργοποίηση	Μηχανισμός	Τροπικότητες
OGM-GE [26]	Gradient Modulation	Κάθε επανάληψη	Εφαρμόζεται μεταξύ συγκεκριμένων εποχών	Τα gradients κλιμακώνονται για κάθε τροπικότητα χρησιμοποιώντας συντελεστές από το discrepancy ratio. Οι λόγοι καθοδηγούν τη διαδικασία τροποποίησης εντοπίζοντας τις ισχυρότερες τροπικότητες και η κυρίαρχη τροπικότητα τιμωρείται.	Ήχος, Όραση
	Generalization Enhancement	Κάθε επανάληψη	Προαιρετικά ενεργοποιείται μεταξύ συγκεκριμένων εποχών	Ενσωματώνει έγχυση Γκαουσιανού θορύβου στις βαθμίδες για τη βελτίωση της γενίκευσης.	
	Discrepancy Ratio	Κάθε επανάληψη	Κατά τη διάρκεια της εμπρόσθιας διάδοσης	Ποσοτικοποιεί την απόκλιση μεταξύ τροπικοτήτων στην εκμάθηση.	
	Learning Rate Decay	Κάθε εποχή	Μετά από ορισμένο βήμα εποχών	Μειώνει το ρυθμό εκμάθησης κατά έναν παράγοντα.	
ACC [20]	Gradient Modulation	Κάθε επανάληψη	Εφαρμόζεται μεταξύ συγκεκριμένων εποχών	Τα gradients κλιμακώνονται για κάθε τροπικότητα χρησιμοποιώντας συντελεστές βασισμένους στο discrepancy ratio. Η ασθενέστερη τροπικότητα ενισχύεται.	Ήχος, Όραση
	Discrepancy Ratio	Κάθε επανάληψη	Κατά τη διάρκεια της εμπρόσθιας διάδοσης	Ποσοτικοποιεί την απόκλιση μεταξύ τροπικοτήτων στην εκμάθηση.	
	Learning Rate Decay	Κάθε εποχή	Μετά από ορισμένο βήμα εποχών	Μειώνει το ρυθμό εκμάθησης κατά έναν παράγοντα.	
AGM [27]	Modality Masking with Shapley values	Κάθε επανάληψη	Κατά τη διάρκεια της εμπρόσθιας διάδοσης	Πραγματοποιεί τρεις εμπρόσθιες διελεύσεις: μία με όλες τις τροπικότητες, μία χωρίς το κείμενο και μία χωρίς τον ήχο, για να απομονώσει τις μεμονωμένες συνεισφορές των τροπικοτήτων με βάση τιμές εμπνευσμένες από τη μέθοδο Shapley.	Ήχος, Κείμενο
	Competition Strength	Κάθε επανάληψη	Κατά τη διάρκεια της εμπρόσθιας διάδοσης	Ποσοτικοποιεί τη δύναμη κάθε τροπικότητας βάσει της μονοτροπικής συνεισφοράς της σε κάθε εμπρόσθια διέλευση.	
	Gradient Modulation	Κάθε επανάληψη	Εφαρμόζεται μεταξύ συγκεκριμένων εποχών	Τα gradients κάθε τροπικότητας κλιμακώνονται ξεχωριστά χρησιμοποιώντας συντελεστές που υπολογίζονται από λόγους ανταγωνιστικής ισχύος.	
	Learning Rate Decay	Κάθε εποχή	Μετά από ορισμένο βήμα εποχών	Μειώνει το ρυθμό εκμάθησης κατά έναν παράγοντα.	
	Adaptive Gradient Clipping	Κάθε επανάληψη	Ενεργοποιείται όταν οι τιμές των gradientsv υπερβαίνουν προκαθορισμένα όρια	Κλιμακώνει τις βαθμίδες σε μέγιστη νόρμα 1.0, σταθεροποιώντας τις ενημερώσεις.	
PMR [20]	Prototypical Loss Adjustment	Κάθε επανάληψη	Εφαρμόζεται μεταξύ συγκεκριμένων εποχών	Μετρά το σφάλμα ταξινόμησης για κάθε τροπικότητα χρησιμοποιώντας την απόσταση μεταξύ μονοτροπικών χαρακτηριστικών και των κατηγορικών τους πρωτοτύπων.	Ήχος, Όραση
	Prototypical Regularization Term	Κάθε επανάληψη	Προαιρετικά ενεργοποιείται μεταξύ συγκεκριμένων εποχών	Μειώνει την εντροπία των κατανομών κλάσεων της ταχύτερα μαθαίνουσας τροπικότητας για να μετριάσει την κυριαρχία της.	
	Imbalanced Ratio	Κάθε επανάληψη	Κατά τη διάρκεια της εμπρόσθιας διάδοσης	Ποσοτικοποιεί την ανισορροπία μεταξύ των τροπικοτήτων	
	Learning Rate Decay	Κάθε εποχή	Μετά από ορισμένο βήμα εποχών	Μειώνει το ρυθμό εκμάθησης κατά έναν παράγοντα.	
RECONBOOST [21]	Alternating Technique	Κάθε φάση εκπαίδευσης	Όταν επιλέγεται νέα τροπικότητα για εκπαίδευση	Εκπαιδεύει έναν εκτιμητή τροπικότητας τη φορά, επιτρέποντας στο σύνολο να επικεντρωθεί σε ασθενείς ή μη αποδοτικές τροπικότητες.	Κείμενο, Ήχος, Όραση
	Ensemble Forward Pass	Κάθε βήμα	Κατά τη διάρκεια της εμπρόσθιας διάδοσης	Συγκεντρώνει προβλέψεις από όλες τις τροπικότητες του συνόλου.	
	Boosting Scheme	Κάθε στάδιο	Σε κάθε φάση εκπαίδευσης όταν προστίθενται νέες τροπικότητες	Ρυθμίζει δυναμικά τη συνεισφορά κάθε τροπικότητας χρησιμοποιώντας μία παράμετρο ενίσχυσης.	
	Global Rectification Scheme	Μετά από κάθε στάδιο	Από το πρώτο στάδιο	Προσαρμόζει το μοντέλο του συνόλου, βελτιώνοντας όλες τις προστιθέμενες τροπικότητες.	
	Memory Consolidation Regularization	Κάθε επανάληψη	Κατά τη διάρκεια της οπισθοδιάδοσης	Κανονικοποιεί τις εξόδους των νέων τροπικοτήτων χρησιμοποιώντας σφάλμα ελαχίστων τετραγώνων ώστε να μην χάνεται χρήσιμη πληροφορία.	

Πίνακας 1. *Συνολική επισκόπηση των μεθόδων βελτιστοποίησης που εξετάζονται στην παρούσα έρευνα όπως αυτές παρουσιάστηκαν στις αντίστοιχες δημοσιεύσεις.*

ορισμένες τεχνικές στοχεύουν στην επιλογή των πιο χρήσιμων τροπικοτήτων, απορρίπτοντας εκείνες που συνεισφέρουν λιγότερο στην απόδοση του μοντέλου [24] [34] [35] [36]. Άλλες προσεγγίσεις υιοθετούν ένα εναλλασσόμενο μαθησιακό πρότυπο, κατά το οποίο κάθε φορά εκπαιδεύεται μία μόνο τροπικότητα, αποφεύγοντας έτσι τις συγκρούσεις κατά τον αλγόριθμο οπισθοδιάδοσης [21] [25].

Στη συγκεκριμένη έρευνα επικεντρωνόμαστε σε τέσσερις μεθόδους βελτιστοποίησης. Δύο από αυτές, οι OGM-GE [26] και AGM [27], ασχολούνται με την άμεση τροποποίηση των gradients κάθε υποδικτύου τροπικότητας και την ποσοτικοποίηση της συνεισφοράς κάθε τροπικότητας, ώστε να επιτευχθεί καλύτερη ισορροπία μεταξύ τους. Οι άλλες δύο, PMR [20] και ReconBoost [21], βασίζονται σε τεχνική πολλαπλών απωλειών. Συγκεκριμένα, η PMR εφαρμόζει ένα σχήμα ποινής-επιβράβευσης, ενώ η ReconBoost υιοθετεί το εναλλασσόμενο μαθησιακό πρότυπο προκειμένου να ενισχύσει τη μάθηση των ασθενέστερων τροπικοτήτων. Ο Πίνακας 1 επεξηγεί συνοπτικά τους βασικούς μηχανισμούς που χρησιμοποιούνται στις μεθόδους αυτές.

0.6 Αποτελέσματα Πειραμάτων και Ανάλυση

Η πειραματική αξιολόγηση των τεσσάρων μεθόδων (OGM-GE, AGM, PMR, ReconBoost) πραγματοποιήθηκε στο σύνολο δεδομένων CMU-MOSI [37] και CMU-MOSEI [38], με στόχο τη μελέτη της απόδοσής τους στην αντιμετώπιση του προβλήματος της ανισορροπίας μεταξύ τροπικοτήτων για ταξινόμηση συναισθήματος. Τα πειράματα σχεδιάστηκαν ώστε να απαντήσουν στα εξής ερευνητικά ερωτήματα:

- Ποια μέθοδος ενισχύει την ισορροπία μεταξύ τροπικοτήτων, αυξάνοντας την ακρίβεια και μειώνοντας τις απώλειες, σε σύγκριση με το βασικό μοντέλolate concatenation with joint training.
- Ποιες στρατηγικές βελτιστοποίησης συμβάλλουν στην ανάδειξη των αδύναμων τροπικοτήτων.
- Πώς επιδρούν παράγοντες όπως ο ρυθμός ενημέρωσης παραμέτρων βελτιστοποίησης, η επιλογή optimizer, η διάρκεια εκπαίδευσης και η εκτεταμένη διάρκεια εφαρμογής των προτεινόμενων διαμορφόσεων στη σύγκλιση των μοντέλων.
- Πώς επιδρά η χρήση ενός συνόλου ανάπτυξης για βοηθητικούς υπολογισμούς των μεθόδων στην αμερόληπτη εκπαίδευση του δικτύου και την καθολική εφαρμογή τους ανεξάρτητα από το batch size.

0.6.1 Βασική Σύγκριση των Μεθόδων

Στον Πίνακα 2, παρουσιάζονται τα βασικά αποτελέσματα των τεσσάρων μεθόδων στην τυπική τους εφαρμογή, χωρίς επιπλέον παραμετροποιήσεις. Τα μοντέλα στα πειράματά μας εκπαιδεύονται χρησιμοποιώντας Adam, με διαφορετικούς ρυθμούς μάθησης και έναν προγραμματιστή ρυθμού μάθησης (scheduler) ReduceLROnPlateau από το PyTorch. Ο scheduler μειώνει τον ρυθμό μάθησης κατά έναν παράγοντα 0.1 όταν το validation loss δεν

βελτιώνεται για μια καθορισμένη περίοδο εποχών. Για το σύνολο δεδομένων CMU-MOSI, η περίοος αυτή έχει οριστεί σε 5, ενώ για το CMU-MOSEI έχει οριστεί σε 20. Χρησιμοποιούμε batch size 16 για το CMU-MOSI και 32 για το CMU-MOSEI. Η τεχνική Early Stopping εφαρμόζεται με περίοδο υπομονής 8 εποχών. Αυτές οι ρυθμίσεις εφαρμόζονται σε όλα τα πειράματα, και οποιαδήποτε απόκλιση θα αναφέρεται ρητά. Ως αντικειμενική συνάρτηση βελτιστοποίησης χρησιμοποιείται Cross-Entropy Loss σε όλα τα πειράματα. Οι επιμέρους συγκρίσεις πραγματοποιούνται με το μοντέλο Late Concatenation, ενώ τα μοντέλα Ensemble και Uni-Pre Finetuned προστίθενται ως ισχυρά μοντέλα στη βιβλιογραφία για συγκριτική αξιολόγηση.

Τροπικότητα	**Μέθοδος**	**CMU-MOSI**		**CMU-MOSEI**	
		Ακρίβεια(%)	**Απώλεια(%)**	**Ακρίβεια (%)**	**Απώλεια(%)**
Ήχος, Εικόνα	Ensemble	47.96 ± 4.94	84.90 ± 1.50	32.63 ± 0.29	166.05 ± 0.75
	Uni-Pre Finetuned	51.46 ± 1.64	84.62 ± 1.07	32.55 ± 0.28	166.27 ± 0.49
	Late Concatenation	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	OGM	53.30 ± 3.12	85.31 ± 2.62	32.67 ± 0.33	166.76 ± 0.86
	OGM-GE	52.48 ± 1.27	84.88 ± 1.23	32.40 ± 0.30	167.12 ± 0.80
	ACC	52.39 ± 2.42	85.98 ± 2.33	32.56 ± 0.37	166.70 ± 1.19
	AGM	53.73 ± 3.65	84.86 ± 1.81	32.71 ± 0.44	166.17 ± 0.64
	PMR	51.52 ± 2.85	85.24 ± 1.64	32.44 ± 0.44	166.65 ± 0.57
	ReconBoost	47.29 ± 5.19	87.63 ± 2.73	33.14 ± 0.37	167.99 ± 1.18
Κείμενο, Εικόνα	Ensemble	73.35 ± 1.72	66.67 ± 1.82	43.94 ± 0.67	136.22 ± 1.17
	Uni-Pre Finetuned	75.72 ± 0.93	64.62 ± 1.19	45.22 ± 0.41	130.93 ± 0.64
	Late Concatenation	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	OGM	75.04 ± 0.96	63.54 ± 0.69	44.15 ± 0.43	133.15 ± 0.35
	OGM-GE	73.50 ± 0.79	64.44 ± 1.26	43.58 ± 0.40	133.74 ± 0.15
	ACC	74.67 ± 1.68	64.36 ± 1.85	44.12 ± 0.21	133.13 ± 0.24
	AGM	74.61 ± 0.86	63.83 ± 1.12	44.15 ± 0.39	132.58 ± 0.34
	PMR	75.51 ± 0.50	64.51 ± 1.42	44.29 ± 0.17	131.94 ± 0.56
	ReconBoost	74.79 ± 1.04	63.20 ± 1.26	44.78 ± 0.29	130.88 ± 0.75
Ήχος, Κείμενο, Εικόνα	Ensemble	72.62 ± 1.25	71.45 ± 1.00	40.40 ± 1.36	141.68 ± 1.32
	Uni-Pre Finetuned	75.65 ± 0.87	64.69 ± 1.10	44.65 ± 0.37	131.51 ± 0.63
	Late Concatenation	74.87 ± 1.23	63.05 ± 1.65	44.46 ± 0.41	131.87 ± 0.60
	ReconBoost	75.09 ± 0.88	63.18 ± 1.30	44.42 ± 0.53	130.79 ± 0.76

Πίνακας 2. *Σύγκριση απόδοσης των προτεινόμενων μεθόδων και των βασικών μοντέλων σε διαφορετικούς συνδυασμούς τροπικοτήτων ('Ηχος-Εικόνα, Κείμενο-Εικόνα και 'Ηχος-Κείμενο-Εικόνα) στα σύνολα δεδομένων CMU-MOSI και CMU-MOSEI υπό τυπικές συνθήκες εκπαίδευσης. Οι μέθοδοι περιλαμβάνουν τα Ensemble, Uni-Pre Finetuned, Late Concatenation (Baseline), OGM, OGM-GE, ACC, AGM, PMR και ReconBoost. Τα αποτελέσματα προκύπτουν ως ο μέσος όρος 5 ανεξάρτητων τρεξιμάτων και παρουσιάζονται με βάση την ακρίβεια (Accuracy) και την απώλεια (Loss), καθώς και την τυπική απόκλιση. Ο πίνακας επισημαίνει τα μοντέλα με την καλύτερη απόδοση για κάθε μετρική, χρησιμοποιώντας ένα χρωματικά κωδικοποιημένο σύστημα. Οι σκούρες αποχρώσεις αντιπροσωπεύουν την καλύτερη απόδοση, οι μεσαίες αποχρώσεις δείχνουν τη δεύτερη καλύτερη απόδοση, ενώ οι ανοιχτές αποχρώσεις υποδηλώνουν την τρίτη καλύτερη απόδοση.*

Τα πειραματικά αποτελέσματα δείχνουν ότι το Late Concatenation μοντέλο, που χρησιμοποιείται ως το βασικό μοντέλο σύγκρισης, αποτελεί το αποδοτικότερο μοντέλο ήχου-εικόνας στο CMU-MOSI, με καμία από τις εξεταζόμενες μεθόδους να ξεπερνά την απόδοσή του. Αντίστοιχα, στο CMU-MOSEI, οι μέθοδοι AGM και ReconBoost παρουσιάζουν τη μεγαλύτερη ακρίβεια. Στα μοντέλα κειμένου-ήχου, το Uni-Pre Finetuned επιτυγχάνει την καλύτερη συνολική ισορροπία μεταξύ ακρίβειας και απώλειας. Το PMR βελτιώνει το βασικό μοντέλο, ενώ το OGM (CMU-MOSI) και το ReconBoost (CMU-MOSEI) ξεπερνούν τη Late Concatenation μέθοδο, προσφέροντας καλύτερη ακρίβεια και μικρότερες απώλειες. Οι

δυναμικές μέθοδοι βελτιστοποίησης φαίνονται πιο αποτελεσματικές όταν υπάρχει κυρίαρχη τροπικότητα, αλλά δεν ξεπερνούν το Uni-Pre Finetuned, που παραμένει το ισχυρότερο μοντέλο. Στα τριτροπικά μοντέλα, το Uni-Pre Finetuned συνεχίζει να αποδίδει καλύτερα, με την προσθήκη μιας τρίτης τροπικότητας (ήχου) να έχει ελάχιστη επίδραση στην τελική απόδοση. Επιβεβαιώνεται ότι το κείμενο είναι η κυρίαρχη τροπικότητα τόσο στο σύνολο CMU-MOSEI, όσο και στο CMU-MOSI, όπου παρατηρείται μικρή αύξηση ακρίβειας με την τρίτη τροποκότητα (του ήχου) χωρίς όμως σημαντική μείωση της απώλειας.

Τέλος, τα αποτελέσματά μας επιβεβαιώνουν από προηγούμενες παρατηρήσεις [39], ότι οι εξεταζόμενες μέθοδοι αποτυγχάνουν να βελτιώσουν συστηματικά την απόδοση σε όλες τις περιπτώσεις. Παρότι χρησιμοποιούνται αρχιτεκτονικές transformer, καθώς και μοντέλα αναγνώρισης συναισθήματος αντί για ταξινόμησης, τα ευρήματά τους καταδεικνύουν παρόμοιους περιορισμούς των τεχνικών βελτιστοποίησης. Άλλες μελέτες [40] [41] αμφισβητούν τη γενική αποτελεσματικότητα των μεθόδων OGM-GE, AGM, PMR υπό διαφορετικά επίπεδα ανισορροπίας μεταξύ των τροπικοτήτων. Αυτό εγείρει ερωτήματα σχετικά με τη γενίκευση και αποτελεσματικότητα τέτοιων μεθόδων σε διάφορα προβλήματα πολυτροπικής ανάλυσης συναισθήματος.

0.6.2 Ανάλυση Παραμέτρων Εκπαίδευσης

Για την εις βάθος κατανόηση των δυναμικών κάθε μεθόδου, μελετήθηκαν οι επιδράσεις κρίσιμων παραμέτρων εκπαίδευσης:

Μέθοδος	**Ήχος-Εικόνα**				**Κείμενο-Εικόνα**			
	CMU-MOSI		**CMU-MOSEI**		**CMU-MOSI**		**CMU-MOSEI**	
	Ακρίβεια (%)	**Απώλεια (%)**	**Ακρίβεια (%)**	**Απώλεια (%)**	**Ακρίβεια (%)**	**Απώλεια (%)**	**Ακρίβεια (%)**	**Απώλεια (%)**
Ενημέρωση Βελτιστοποίησης Κάθε Επανάληψη								
Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
OGM	53.30 ± 3.12	85.31 ± 2.62	32.67 ± 0.33	166.76 ± 0.86	75.04 ± 0.96	63.54 ± 0.69	44.15 ± 0.43	133.15 ± 0.35
OGM-GE	52.48 ± 1.27	84.88 ± 1.23	32.40 ± 0.30	167.12 ± 0.80	73.50 ± 0.79	64.44 ± 1.26	43.58 ± 0.40	133.74 ± 0.15
Ενημέρωση Βελτιστοποίησης Κάθε 4 Επαναλήψεις								
Baseline	53.67 ± 1.87	85.48 ± 1.26	32.59 ± 0.22	167.46 ± 0.77	73.32 ± 0.86	67.20 ± 0.83	43.66 ± 0.38	133.97 ± 0.41
OGM	52.95 ± 1.65	85.79 ± 0.97	32.72 ± 0.33	166.59 ± 0.76	73.64 ± 1.19	66.65 ± 1.08	43.83 ± 0.40	133.70 ± 0.51
OGM-GE	47.84 ± 1.52	87.15 ± 0.87	32.48 ± 0.17	167.94 ± 0.46	73.85 ± 1.04	66.41 ± 0.80	43.94 ± 0.39	133.91 ± 0.35

Πίνακας 3. *Επίδοση των μοντέλων Ήχου-Εικόνας και Κειμένου-Εικόνας στα σύνολα δεδομένων CMU-MOSI και CMU-MOSEI χρησιμοποιώντας τις μεθόδους OGM και OGM-GE υπό διαφορετικές συχνότητες ενημερώσεων βελτιστοποίησης. Τα αποτελέσματα προκύπτουν ως ο μέσος όρος 5 ανεξάρτητων τρεξιμάτων και παρουσιάζονται με βάση την ακρίβεια (Accuracy) και την απώλεια (Loss), καθώς και την τυπική απόκλιση.*

Συχνότητα Ενημέρωσης Παραμέτρων Βελτιστοποίησης: Η συχνότητα των ενημερώσεων βελτιστοποίησης επηρεάζει σημαντικά τη δυναμική εκμάθησης των νευρωνικών δικτύων, διαμορφώνοντας την ταχύτητα σύγκλισης, τη σταθερότητα και τη γενίκευση. Πιο συχνές ενημερώσεις, όπως αυτές που βασίζονται σε μικρότερα batch sizes, επιτρέπουν στο μοντέλο να προσαρμόζεται γρήγορα στις αλλαγές των gradients, αλλά ενδέχεται να εισάγουν υψηλή διακύμανση, οδηγώντας σε αστάθεια ή θορυβώδη βελτιστοποίηση. Αντίθετα, λιγότερο συχνές ενημερώσεις, όπως εκείνες που χρησιμοποιούν μεγαλύτερα batc sizes, προσφέρουν πιο σταθερή βελτιστοποίηση, αλλά μπορεί να επιβραδύνουν τη σύγκλιση και να δυσκο-

λεύονται σε δυναμικές συνθήκες μάθησης. Τα αποτελέσματα του Πίνακα 3 δείχνουν ότι η ενημέρωση σε κάθε επανάληψη γενικά οδηγεί σε καλύτερη απόδοση τόσο σε όρους ακρίβειας όσο και απωλειών σε όλα τα μοντέλα και σύνολα δεδομένων που δοκιμάστηκε.

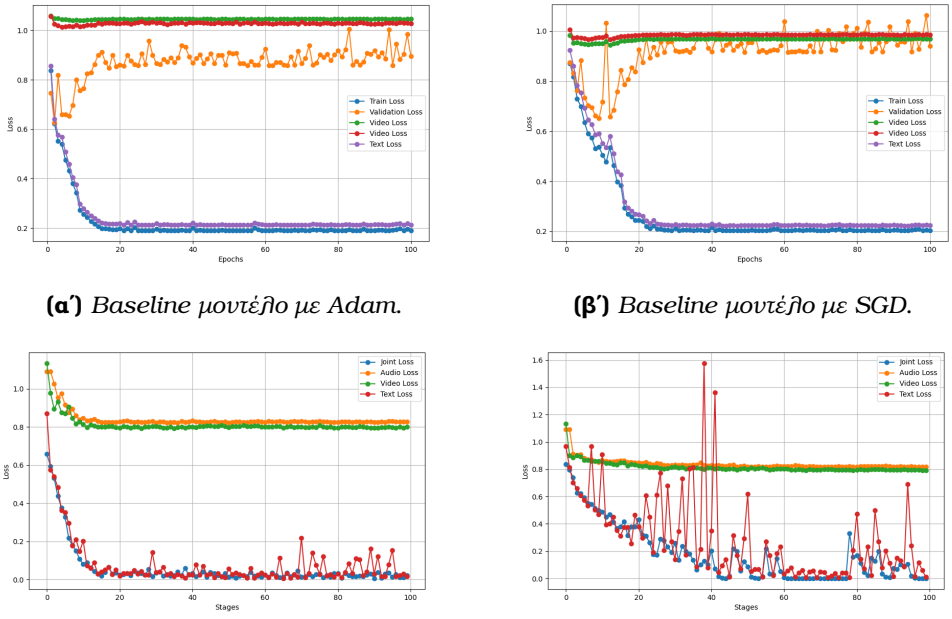
Επιλογή Optimizer: Η κύρια διαφορά μεταξύ των Adam [42] και SGD έγκειται στον τρόπο με τον οποίο διαχειρίζονται τους ρυθμούς μάθησης και ενημερώνουν τις παραμέτρους του μοντέλου. Ο SGD χρησιμοποιεί έναν σταθερό, καθολικό ρυθμό μάθησης για όλες τις παραμέτρους του μοντέλου και τις ενημερώνει με βάση την τρέχουσα βαθμίδα. Αντίθετα, ο Adam προσαρμόζει δυναμικά τον ρυθμό μάθησης για κάθε παράμετρο, διατηρώντας κινητούς μέσους όρους τόσο της κλίσης (πρώτη ροπή) όσο και του τετραγώνου της κλίσης (δεύτερη ροπή). Αυτό είναι ιδιαίτερα χρήσιμο στην περίπτωσή μας, καθώς όπως είπαμε οι παράμετροι ενός πολυτροπικού νευρωνικού δικτύου συγκλίνουν με διαφορετικούς ρυθμούς. Επιπλέον, ο Adam συνήθως συγκλίνει ταχύτερα χάρη στις προσαρμοστικές ενημερώσεις του, καθιστώντας τον ιδανικό για περιπτώσεις με θορυβώδεις ή αραιές βαθμίδες [42]. Στην περίπτωσή μας, τα χαρακτηριστικά κειμένου έχουν προκύψει από το μοντέλο BERT [43], το οποίο έχει συχνά προεκπαιδευτεί με τους βελτιστοποιητές Adam ή AdamW [43], έτσι η χρήση του Adam στο πολυτροπικό μας δίκτυο διατηρεί τη συνοχή στη δυναμική της βελτιστοποίησης.

Τροπικότητα	**Μέθοδος**	**CMU-MOSI**		**CMU-MOSEI**	
		Ακρίβεια(%)	**Απώλεια(%)**	**Ακρίβεια (%)**	**Απώλεια(%)**
Ήχος, Εικόνα	Baseline	49.97 ± 2.59	85.38 ± 1.40	32.51 ± 0.16	167.24 ± 0.41
	OGM	49.82 ± 2.38	85.04 ± 1.25	32.37 ± 0.30	167.45 ± 0.42
	OGM-GE	49.42 ± 1.43	85.55 ± 0.71	32.63 ± 0.20	167.15 ± 0.60
	ACC	50.55 ± 1.80	84.86 ± 0.78	32.56 ± 0.31	167.56 ± 0.31
	AGM	49.59 ± 1.17	85.15 ± 0.77	32.42 ± 0.23	166.94 ± 0.63
	PMR	51.25 ± 1.85	85.74 ± 1.42	32.56 ± 0.31	167.56 ± 0.31
	ReconBoost	49.88 ± 3.01	94.17 ± 3.60	32.42 ± 0.02	175.56 ± 0.80
Ήχος, Εικόνα	Baseline	73.21 ± 2.81	63.91 ± 2.15	44.46 ± 0.54	131.12 ± 0.40
	OGM	73.53 ± 1.79	65.11 ± 3.86	44.39 ± 0.48	131.93 ± 0.86
	OGM-GE	74.17 ± 0.90	64.08 ± 1.73	44.33 ± 0.72	131.88 ± 0.71
	ACC	74.67 ± 1.02	62.96 ± 0.30	44.43 ± 0.54	131.46 ± 0.89
	AGM	73.76 ± 1.17	65.75 ± 1.31	44.10 ± 0.13	130.95 ± 0.81
	PMR	74.55 ± 0.79	63.95 ± 0.52	44.29 ± 0.26	132.11 ± 1.01
	ReconBoost	73.45 ± 5.19	71.11 ± 7.90	44.37 ± 0.68	133.18 ± 1.09
Ήχος, Κείμενο, Εικόνα	Baseline	73.24 ± 2.24	65.15 ± 2.35	44.80 ± 0.54	130.78 ± 0.48
	ReconBoost	73.07 ± 2.36	76.91 ± 5.95	44.30 ± 0.61	137.20 ± 4.35

Πίνακας 4. *Επίδοση των μοντέλων Ήχου-Εικόνας, Κειμένου-Εικόνας Ήχου-Κειμένου-Εικόνας στα σύνολα δεδομένων CMU-MOSI και CMU-MOSEI χρησιμοποιώντας τις μεθόδους OGM και OGM-GE υπό του βελτιστοποιητή SGD. Τα αποτελέσματα προκύπτουν ως ο μέσος όρος 5 ανεξάρτητων τρεξιμάτων και παρουσιάζονται με βάση την ακρίβεια (Accuracy) και την απώλεια (Loss), καθώς και την τυπική απόκλιση.*

Από τους Πίνακες 2 και 4 διαφαίνεται ότι η επιλογή optimizer διαδραματίζει καθοριστικό ρόλο τόσο στην ακρίβεια ταξινόμησης όσο και στη σταθερότητα της εκπαίδευσης. Οι μέθοδοι ACC και PMR με SGD επιτυγχάνουν βελτιωμένη απόδοση και μειωμένη απώλεια, παρουσιάζοντας σταθερές τιμές τυπικής απόκλισης σε σύγκριση με τα αντίστοιχα βασικά μοντέλα στο σύνολο δεδομένων CMU-MOSI. Αντίθετα, στο CMU-MOSEI, τα αποτελέσματα είναι συγκρίσιμα, χωρίς κάποια από τις μεθόδους να επιτυγχάνει σαφή υπεροχή. Όλα τα υπόλοιπα μοντέλα δεν φαίνεται να ευνοούνται από τη χρήση του SGD, καθώς όχι μόνο αδυνατούν να ξεπεράσουν τις επιδόσεις του βασικού μοντέλου με SGD, αλλά παρουσιάζουν χαμηλότερη απόδοση σε σύγκριση με τα αντίστοιχα μοντέλα που χρησιμοποιούν Adam. Αξιοσημείωτη

είναι η περίπτωση του Text-Vision μοντέλου, όπου το OGM-GE μοντέλο καταφέρνει να ξεπεράσει την απόδοση του αντίστοιχου μοντέλου με Adam, υποδεικνύοντας ότι η ενίσχυση με Γκαουσιανό θόρυβο δεν αξιοποιείται πλήρως στην περίπτωση του Adam. Ωστόσο, το ReconBoost, το οποίο βασίζεται σε εναλλασσόμενη μονοτροπική εκπαίδευση σε συνδυασμό με πολλαπλές απώλειες, παρουσιάζει έντονες διακυμάνσεις στην εκπαίδευση των μοντέλων στο CMU-MOSI, γεγονός που επιβεβαιώνεται και από τις αντίστοιχες γραφικές παραστάσεις του Γραφήματος 1 που ακολουθούν.



(α΄) *Baseline μοντέλο με Adam.*

(β΄) *Baseline μοντέλο με SGD.*

(γ΄) *ReconBoost μοντέλο με Adam.*

(δ΄) *ReconBoost μοντέλο με SGD.*

Γράφημα 1. *Συνάρτηση απωλειών με χρήση Adam έναντι SGD για μία εκτέλεση στο Baseline και ReconBoost μοντέλο με τρεις εισερχόμενες τροπικότητες στο σύνολο δεδομένων CMU-MOSI.*

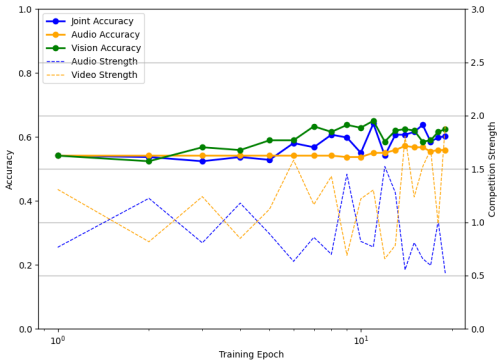
Batch Size και Χρήση Συνόλου Ανάπτυξης: Η επιλογή του batch size επηρεάζει άμεσα τη διακύμανση των gradients. Έρευνες [44] [45] [46] έχουν δείξει ότι η εκπαίδευση με μεγάλα batch sizes συχνά συνδέεται με σύγκλιση σε πιο απότομα ελάχιστα κοντά στην αρχική κατάσταση, με αποτέλεσμα χειρότερη γενίκευση σε σύνολα δοκιμών. Βάσει αυτών των παρατηρήσεων, αλλά και πολυτροπικών υλοποιήσεων για ανάλυση συναισθήματος όπως το Self-MM [47] και το Multimodal Multi-Loss Fusion Network (MMML) [48], έχουμε υιοθετήσει batch sizes 16 για το CMU-MOSI και 32 για το CMU-MOSEI. Η επιλογή αυτή εξυπηρετεί την αποφυγή θορύβου gradients, την αποτελεσματική εκπαίδευση μέσω mini-batch gradient descent και την πρακτική εφαρμογή πολυτροπικών αρχιτεκτονικών late concatenation fusion.

Μέθοδος	Ήχος-Εικόνα				Κείμενο-Εικόνα			
	CMU-MOSI		CMU-MOSEI		CMU-MOSI		CMU-MOSEI	
	Ακρίβεια (%)	Απώλεια (%)	Ακρίβεια (%)	Απώλεια (%)	Ακρίβεια (%)	Απώλεια (%)	Ακρίβεια (%)	Απώλεια (%)
Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
OGM	49.13 ± 5.20	86.11 ± 1.24	32.68 ± 0.26	166.52 ± 0.73	74.49 ± 0.86	64.37 ± 1.17	44.16 ± 0.24	132.02 ± 0.82
OGM-GE	50.79 ± 3.98	85.40 ± 1.40	32.31 ± 0.14	167.81 ± 0.56	74.49 ± 0.79	63.54 ± 0.56	43.45 ± 0.52	132.99 ± 0.79
AGM	52.25 ± 1.97	84.32 ± 1.24	32.71 ± 0.31	166.81 ± 1.01	74.46 ± 1.93	64.26 ± 1.69	43.84 ± 0.11	133.26 ± 0.63
PMR	53.76 ± 2.15	84.55 ± 1.14	32.44 ± 0.25	167.19 ± 0.92	74.72 ± 1.22	63.33 ± 1.95	44.60 ± 0.53	130.60 ± 0.36

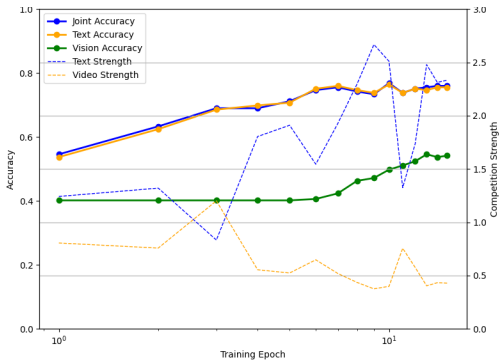
Πίνακας 5. *Επίδοση των μοντέλων 'Ηχου-Εικόνας και Κειμένου-Εικόνας στα σύνολα δεδομένων CMU-MOSI και CMU-MOSEI χρησιμοποιώντας ένα σύνολο ανάπτυξης για αμερόληπτους υπολογισμούς του ρυθμού απόκλισης τροπικοτήτων στις μεθόδους OGM και OGM-GE, της δύναμης τροπικότητας στην μέθοδο AGM και του ρυθμού ανισοροπίας στα μοντέλα PMR. Τα αποτελέσματα προκύπτουν ως ο μέσος όρος 5 ανεξάρτητων τρεξιμάτων και παρουσιάζονται με βάση την ακρίβεια (Accuracy) και την απώλεια (Loss), καθώς και την τυπική απόκλιση.*

Τα πειράματα με το σύνολο ανάπτυξης (development set) αποσκοπούν στην αξιολόγηση της ευαισθησίας των μεθόδων στο batch size και στην εξερεύνηση ενός πιο ευέλικτου τρόπου εφαρμογής τους, χωρίς να περιορίζονται από το μέγεθος αυτό. Τελικός στόχος είναι η εφαρμογή μεθόδων που παραμένουν αποτελεσματικές ανεξάρτητα από τους περιορισμούς batch size, αποφεύγοντας παράλληλα τη μεροληψία προς τα χαρακτηριστικά των τροπικοτήτων ή τις μαθησιακές δυναμικές. Αυτό επιτυγχάνεται μέσω του υπολογισμού του ρυθμού ανισορρποπίας στην μέθοδο PMR, του ρυθμού απόκλισης στις μεθόδους OGM, OGM-GE, των μετρικών ισχύος στη μέθοδο AGM καθώς και των απαραίτητων πολλαπλασιαστών gradients σε δεδομένα που το μοντέλο δεν έχει δει (σύνολο ανάπτυξης), διασφαλίζοντας έτσι αντικειμενική αξιολόγηση της απόδοσής του. Το σύνολο ανάπτυξης αποτελείται από 100 δείγματα του συνόλου εκπαίδευσης για το CMU-MOSI και 200 για το CMU-MOSEI, τα οποία αποκόπτονται από το σύνολο εκπαίδευσης και το δίκτυο δεν επεξεργάζεται ποτέ κατά την διάρκεια εκμάθησης. Εξετάζουμε όχι μόνο την απόδοση αλλά και τις τάσεις των επιμέρους μετρικών και της απώλειας κατά την εκπαίδευση. Στα μοντέλα όλων των μεθόδων παρατηρήσαμε σταθεροποίηση των τάσεων των επιμέρους μετρικών, χωρίς να αδυνατούν να υποδείξουν την κυρίαρχη τροπικότητα, που είναι και ο σκοπός τους. Μάλιστα στην περίπτωση των μοντέλων AGM με χρήση συνόλου ανάπτυξης παρατηρούμε ισοδύναμες αποδόσεις με αυτές της κλασσικής εφαρμογής του αλγορίθμου, ενώ τα μοντέλα PMR βελτιώνουν την απόδοσή τους. Ενδεικτικά, παραθέτουμε τις γραφικές παραστάσεις απόδοσης και μέτρησης ισχύος τροπικότητας των μοντέλων AGM στο CMU-MOSI, με και χωρίς τη χρήση του συνόλου ανάπτυξης (Γράφημα 2).

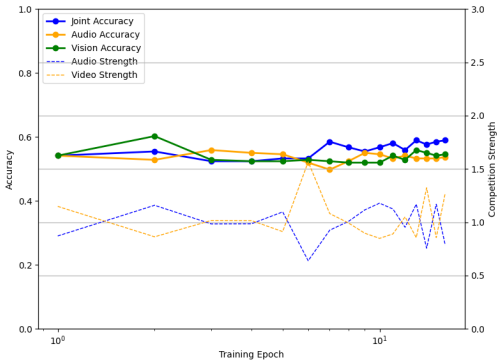
Εκτεταμένη Διάρκεια Εκπαίδευσης: Η εκτεταμένη εκπαίδευση μπορεί να οδηγήσει σε καλύτερη σύγκλιση, ιδιαίτερα για μοντέλα που χρησιμοποιούν SGD, τα οποία συνήθως απαιτούν περισσότερες επαναλήψεις για να επιτύχουν σταθερότητα. Ωστόσο, η υπερβολική εκπαίδευση μπορεί επίσης να οδηγήσει σε υπερπροσαρμογή, όπου το μοντέλο απομνημονεύει τα δεδομένα εκπαίδευσης αντί να μαθαίνει γενικεύσιμα πρότυπα overfitting. Η απόδοση των μοντέλων του Πίνακα 6 δεν ξεπερνούν αυτήν των αντίστοιχων μοντέλων του Πίνακα 5. Παρόλα αυτά, σε συνδυασμό με τις γραφικές απεικονίσεις του Σχήματος 3, αναδεικνύεται η ωφέλεια του μοντέλου OGM-GE από την επέκταση της διάρκειας εκπαίδευσης. Διαφαίνεται ότι ο συνδυασμός παρατεταμένης εκπαίδευσης, σταθερών ενημερώσεων των συντελεστών



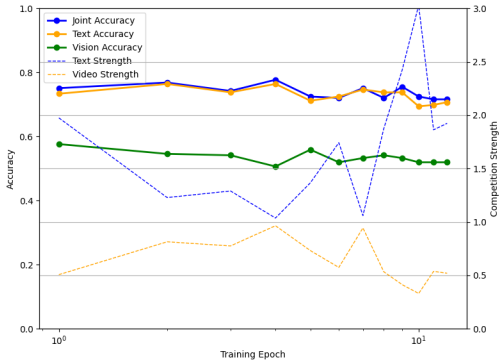
(α΄) *Μουτέλο Ήχου-Εικόνας.*



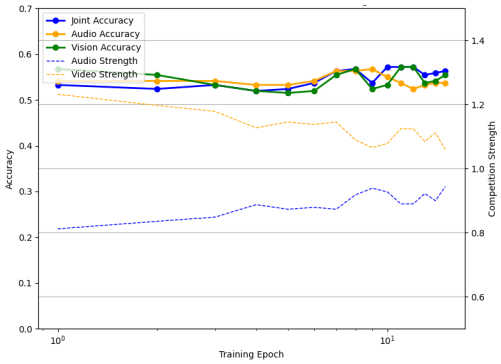
(β΄) *Μουτέλο Κειμένου-Εικόνας.*



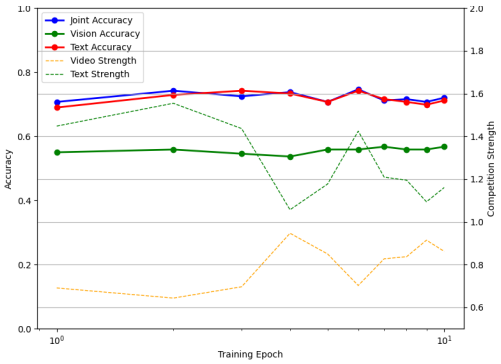
(γ΄) *Μουτέλο AGM Ήχου-Εικόνας.*



(δ΄) *Μουτέλο AGM Κειμένου-Εικόνας.*



(ε΄) *Μουτέλο AGM Ήχου-Εικόνας με χρήση συνόλου ανάπτυξης.*



(ϛ΄) *Μουτέλο AGM Κειμένου-Εικόνας με χρήση συνόλου ανάπτυξης.*

Γράφημα 2. *Σύγκριση των βασικών μουτέλων AGM και του AGM με χρήση συνόλου ανάπτυξης στο CMU-MOSI. Η πρώτη γραμμή αντιπροσωπεύει τα βασικά μουτέλα, η δεύτερη το μουτέλο AGM για τις περιπτώσεις Ήχου-Εικόνας και Κειμένου-Εικόνας, ενώ η τρίτη το AGM με χρήση συνόλου ανάπτυξης για την ενημέρωση της ισχύος και των συντελεστών των τροπικοτήτων. Κάθε γράφημα περιλαμβάνει τις μονοτροπικές ισχείς και τις τιμές ακρίβειας επικύρωσης του μουτέλου για μία εκτέλεση.*

gradients κάθε πέντε επαναλήψεις στο σύνολο ανάπτυξης και εξερεύνησης με ενίσχυση θορύβου [44] [49] [50] [51] [52] επιτρέπει στο OGM-GE να επιτυγχάνει την καλύτερη απόδοση παρουσία κυρίαρχης τροπικότητας. Στο βασικό και OGM μοντέλο, το validation loss αρχίζει να αυξάνεται σημαντικά μετά την 20ή εποχή, αποκλίνει από την απώλεια εκπαίδευσης και υποδεικνύει υπερπροσαρμογή overfitting. Αυτό υπογραμμίζει τη σημασία της ενσωμάτωσης early stopping για τη διατήρηση της ικανότητας γενίκευσης του μοντέλου και την αποφυγή υποβάθμισης της απόδοσης σε σενάρια εκτεταμένης εκπαίδευσης, γεγονός που επιβεβαιώνεται και από την γραφική απωλειών του Σχήματος 1.

Μέθοδος	**Ήχος-Εικόνα**		**Κείμενο-Εικόνα**	
	Ακρίβεια (%)	**Απώλεια (%)**	**Ακρίβεια (%)**	**Απώλεια (%)**
Παρατεταμένη Εκπαίδευση για 100 Εποχές				
Baseline	53.09 ± 2.22	85.54 ± 1.35	72.68 ± 2.08	68.54 ± 1.62
OGM	52.95 ± 2.25	85.61 ± 1.31	72.97 ± 1.99	68.44 ± 1.62
OGM-GE	50.47 ± 2.22	86.82 ± 0.98	73.59 ± 0.64	67.56 ± 1.40

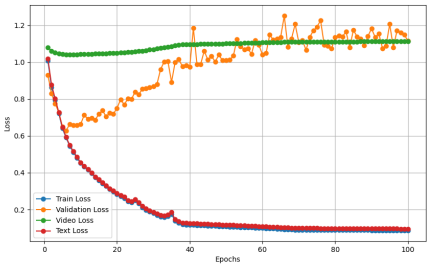
Πίνακας 6. *Αποτελέσματα των μοντέλων 'Ηχου-Εικόνας και Κειμένου-Εικόνας για 100 εποχές εκπαίδευσης χωρίς early stopping στο σύνολο CMU-MOSI. Τα αποτελέσματα προκύπτουν ως ο μέσος όρος 5 ανεξάρτητων τρεξιμάτων και παρουσιάζονται με βάση την ακρίβεια (Accuracy) και την απώλεια (Loss), καθώς και την τυπική απόκλιση.*

Μέθοδος	**Ήχος-Εικόνα**		**Κείμενο-Εικόνα**	
	Ακρίβεια (%)	**Απώλεια (%)**	**Ακρίβεια (%)**	**Απώλεια (%)**
Διαμόρφωσης για Συγκεκριμένο Αριθμό Εποχών				
Baseline	32.55 ± 0.44	166.74 ± 0.88	43.99 ± 0.39	133.11 ± 0.39
OGM	32.67 ± 0.33	166.76 ± 0.86	44.15 ± 0.34	133.11 ± 0.39
OGM-GE	32.40 ± 0.30	167.12 ± 0.80	43.58 ± 0.40	133.74 ± 0.15
Εφαρμογή Καθόλη τη Διάρκεια Εκπαίδευσης				
OGM	32.59 ± 0.40	166.61 ± 0.93	44.02 ± 0.43	133.15 ± 0.47
OGM-GE	32.43 ± 0.29	167.30 ± 0.68	43.13 ± 0.53	134.83 ± 0.21

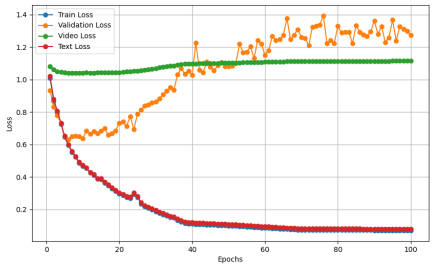
Πίνακας 7. *Απόδοση των μοντέλων 'Ηχου-Εικόνας και Κειμένου-Εικόνας στο σύνολο δεδομένων CMU-MOSEI με την εφαρμογή της προτεινόμενης διαμόρφωσης για συγκεκριμένο αριθμό εποχών έναντι της εφαρμογής της καθόλη τη διάρκεια της εκπαίδευσης. Τα αποτελέσματα προκύπτουν ως ο μέσος όρος 5 ανεξάρτητων τρεξιμάτων και παρουσιάζονται με βάση την ακρίβεια (Accuracy) και την απώλεια (Loss), καθώς και την τυπική απόκλιση.*

Εκτεταμένη Διάρκεια Προτεινόμενων Διαμορφόσεων: Μέσα από την απεικόνιση των συναρτήσεων απωλειών στα γραφήματα 1 και 3, διαπιστώνουμε ότι οι πρώτες εποχές είναι καθοριστικές για την προσπάθεια ισορροπημένης αξιοποίησης των τροπικοτήτων, καθώς μετά από ένα συγκεκριμένο σημείο τα μοντέλα συγκλίνουν. Τα αποτελέσματα του Πίνακα 7 μας οδηγούν στο συμπέρασμα ότι ένα καλά επιλεγμένο παράθυρο διαμόρφωσης (π.χ., οι πρώτες 5 εποχές) ευθυγραμμίζεται με την περίοδο όπου τα gradients είναι πιο ασταθείς, επιτρέποντας στον επιλεγμένο μηχανισμό διαμόρφωσης να σταθεροποιήσει τη διαδικασία εκπαίδευσης. Αντίθετα, η παρατεταμένη προσαρμογή τους μπορεί να παρεμβαίνει στη φυσική σταθεροποίηση της διαδικασίας βελτιστοποίησης, ιδιαίτερα στις μεταγενέστερες εποχές όπου τα gradients είναι ήδη μικρά. Οι μηχανισμοί διαμόρφωσης, όπως οι OGM και OGM-GE, είναι πιο αποτελεσματικοί όταν εφαρμόζονται στρατηγικά στα αρχικά στάδια της εκπαίδευσης.

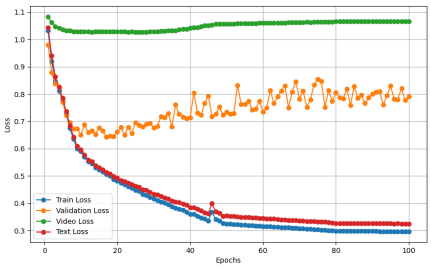
Η επέκταση της εφαρμογής τους σε μεταγενέστερες εποχές ενέχει τον κίνδυνο να διαταράξει τη σύγκλιση, καθώς σε αυτό το στάδιο τα gradients έχουν ήδη προσαρμοστεί για την ελαχιστοποίηση της συνάρτησης απωλειών.



(α΄) *Απεικόνιση απωλειών για το Baseline Μοντέλο.*



(β΄) *Απεικόνιση απωλειών για το OGM Μοντέλο.*



(γ΄) *Απεικόνιση απωλειών για το OGM-GE Μοντέλο.*

Γράφημα 3. *Απεικόνιση απωλειών για το μοντέλο Κειμένου-Εικόνας στο (α) Baseline Μοντέλο (β) OGM Μοντέλο (γ) OGM-GE Μοντέλο του CMU-MOSI συνόλου για 100 εποχές.*

0.7 Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Αν και οι εξεταζόμενες μέθοδοι δυναμικής βελτιστοποίησης δείχνουν δυναμική στη βελτίωση των ασθενέστερων τροπικοτήτων και στη ρύθμιση των διαδικασιών εκπαίδευσης, η ασυνέπειά τους σε διαφορετικές διαμορφώσεις και η αδυναμία τους να ξεπεράσουν ισχυρά βασικά μοντέλα αναδεικνύουν τη σύνθετη φύση της πολυτροπικής βελτιστοποίησης. Ενώ συχνά θεωρείται ότι οι πολυτροπικές προσεγγίσεις υπερτερούν των μονοτροπικών μοντέλων, η παρούσα μελέτη επιβεβαιώνει ότι η ανισορροπία τροπικοτήτων μπορεί να περιορίσει σημαντικά την απόδοση. Η βελτιστοποίηση των πολυτροπικών νευρωνικών δικτύων παραμένει ένα ανοιχτό ζήτημα έρευνας.

Ωστόσο, η ανάπτυξη αποτελεσματικών μεθόδων ποσοτικοποίησης της συνεισφοράς κάθε τροπικότητας, ώστε να θεσπιστούν κανόνες βελτιστοποίησης που θα επιτρέπουν την ανεξάρτητη ρύθμιση κάθε τροπικότητας αποδεικνύεται ως ένα πολλά υποσχόμενο σενάριο.

Παράλληλα με την βελτιστοποίηση, θα πρέπει να διερευνηθούν τρόποι βελτίωσης της πληροφορίας που εξάγεται από τις δευτερεύουσες τροπικότητες, πιθανώς μέσω μηχανισμών cross-modal attention που ενισχύουν τη μοναδική τους συνεισφορά.

Επιπλέον, η μελλοντική έρευνα θα πρέπει να διερευνήσει εάν εναλλακτικές μέθοδοι συγχώνευσης αποφέρουν μεγαλύτερα οφέλη όταν συνδυάζονται με τεχνικές προσαρμοστικής βελτιστοποίησης, αξιολογώντας την αποτελεσματικότητά τους σε διαφορετικές στρατηγικές

συγχώνευσης.

Η μελλοντική έρευνα πρέπει να εξετάσει ακόμα αν είναι εφικτή μία ενιαία στρατηγική βελτιστοποίησης ή αν απαιτούνται εξειδικευμένες προσεγγίσεις για διαφορετικά σύνολα δεδομένων και εφαρμογές, καθώς διαπιστώνεται ότι ρυθμίσεις εκπαίδευσης που αποδίδουν καλά στο CMU-MOSI μπορεί να μην γενικεύονται αποτελεσματικά στο CMU-MOSEI ή σε άλλα πολυτροπικά σύνολα δεδομένων.

Αξίζει να διερευνηθεί και η ανάπτυξη υβριδικών προσεγγίσεων που συνδυάζουν προεκπαίδευση μονοτροπικών μοντέλων με προσαρμοστική πολυτροπική βελτιστοποίηση, καθώς η σταθερά υψηλή απόδοση του Uni-Pre Finetuned μοντέλου υποδηλώνει ότι αυτή η στρατηγική μπορεί να είναι ιδιαίτερα αποτελεσματική.

Ακόμα, αναπτύσσοντας τεχνικές βελτιστοποίησης που μειώνουν την ανάγκη εκτεταμένου πειραματισμού με υπερπαραμέτρους εξασφαλίζεται η πρακτικότητα και η επεκτασιμότητα των πολυτροπικών μοντέλων.

Ο κύριος στόχος θα πρέπει να είναι η ανάπτυξη δυναμικών μεθόδων που προσαρμόζονται αυτόματα στα μαθησιακά χαρακτηριστικά κάθε τροπικότητας, μειώνοντας την εξάρτηση από την προσαρμογή υπερπαραμέτρων και τη δομή των δεδομένων, ενώ παράλληλα διατηρούν υψηλή απόδοση σε διαφορετικές αρχιτεκτονικές. Η παρούσα διπλωματική συμβάλλει σε αυτό, αναλύοντας μεθόδους βασισμένες σε προσαρμογή παραγώγων και συναρτήσεις απωλειών, υπό διάφορες συνθήκες εκπαίδευσης, για την ανάλυση συναισθήματος.

Chapter 1

Introduction

Machine Learning advancements across diverse domains enable systems to learn from data and improve performance over time. Machine Learning forms the foundation for understanding and addressing complex real-world challenges through data-driven approaches.

1.1 Towards Multimodal Machine Learning

The field of Artificial Intelligence (AI) has seen remarkable progress in recent decades, introducing to systems novel ways of processing and understanding information. At the core of this progress is Machine Learning (ML), which allows systems to learn from data and make predictions or take decisions without needing to be explicitly programmed. Machine learning has driven advancements in numerous tasks, such as image recognition, natural language processing, with applications across domains like healthcare, education, and entertainment. These advancements rely on Artificial Neural Networks (ANNs), designed to identify patterns in data and perform on new, unseen data. As ML continues to evolve, the ability to process and combine diverse sources of information has become increasingly essential, opening new possibilities in machine learning. Multimodal Machine Learning represents a paradigm shift in the field.

Multimodal systems are able to analyze and integrate information from multiple data sources, such as text, images, and audio. This approach provides a more comprehensive understanding of complex problems by leveraging complementary information of different modalities. For example, in a sentiment recognition task from a video, relying solely on visual data might miss important emotional cues present in speech tone, while focusing only on audio could overlook critical facial expressions. Multimodal learning has enabled breakthroughs in tasks like video captioning, medical diagnosis, and autonomous driving, where data from one source often fail to address the problem effectively due to the absence of critical information unique to other data streams. The importance of Multimodal Machine Learning lies in its potential to bridge the gap between isolated streams of information, aiming to better replicate human understanding. However, the integration of multimodal data introduces new challenges, such as effectively combining diverse data streams and managing the unique nature of the information each stream provides synchronously. Driven by these challenges, training of Multimodal Neural Networks aims

to harmonize diverse data streams and unlock the full potential of synchronous machine learning applications.

1.2 Motivation

Leveraging the distinct characteristics of each modality to achieve accurate and comprehensive results makes Multimodal Learning a cornerstone of sentiment analysis and emotion classification. To enhance the performance of Multimodal Neural Networks, recent studies have underscored the need for tailored optimization techniques that account for the unique learning dynamics of each modality. Since individual modalities exhibit distinct learning capabilities and progression rates, multimodal models often require optimization strategies specifically designed to adapt to and balance these varying paces of learning effectively [22] [23] [29]. Traditional optimization methods, applying a common optimization approach across all modalities, fail to address this issue effectively. The challenge lies in designing optimization strategies to prevent over-reliance to one stronger modality and promote the exploration of the weaker modalities. This imbalance undermines the learning dynamics, resulting in models that are biased toward dominant modalities while neglecting others. Numerous approaches in literature have been proposed to explore the unique characteristics of multimodal learning and design novel mechanisms to achieve a more balanced integration of modalities.

Inspired by the challenges posed by unbalanced multimodal learning, this thesis investigates the optimization of multimodal neural networks under the scope of sentiment analysis. Motivated by the growing recognition in the literature of dynamic optimization techniques as a promising solution to these issues, we aim to explore a number of novel optimization strategies suggested to mitigate difficulties of multimodal learning. On-the-fly Gradient Modulation with Generalization Enhancement (OGM-GE)[26] and Adaptive Gradient Modulation (AGM)[27] directly modify the gradients of the unimodal subnetworks by scaling them with dynamically computed factors during the optimization step. Prototypical Modal Rebalance (PMR), a multi-loss optimization technique [20], influences optimization indirectly by modifying the loss functions. ReconBoost is a multimodal alternating learning paradigm [21] that dynamically interchanges unimodal models during training following also a multi-loss approach. This variety of approaches allows us to explore optimization dynamics from multiple angles, highlighting how indirect loss-based adjustments or direct gradient modifications address unbalanced multimodal learning.

1.3 Thesis Contribution

This thesis addresses the challenges of modality imbalance in Multimodal Neural Networks by providing a unified evaluation of modality balancing algorithms. Building on insights from dynamic optimization methods, we systematically test the algorithms under both balanced and imbalanced modality conditions to explore their effectiveness in handling real-world scenarios. Through detailed analysis of each algorithm's mechanisms, strengths, and limitations, we aim to uncover how optimization strategies influence the

performance and robustness of multimodal systems. By focusing on the task of sentiment analysis, this research examines the impact of dynamic methods on optimizing unimodal performance and evaluates whether these improvements contribute to enhancing the overall performance of the multimodal neural network. This work makes the following contributions:

- Investigation of dynamic optimization methods: A comprehensive analysis of four dynamic optimization techniques— OGM-GE [26] and AGM [27] gradient-modification methods, PMR [20] and ReconBoost [21] multi-loss optimization methods—specifically tailored for unbalanced multimodal learning in sentiment classification.
- Evaluation across varied imbalance scenarios: Experiments conducted on three distinct imbalance scenarios including one dominant and one weak modality, two weak modalities, and three modalities with varying contributions, providing a detailed understanding of how these methods perform under modalities with varying strengths.
- Analysis of influencing factors: An exploration of factors affecting the efficacy of the dynamic optimization methods, like the choice of optimizer, alongside with proposals to enhance the applicability of the algorithms such as the use of a development set for unbiased auxiliary computations.

The findings of this thesis aim to provide valuable insights into the dynamic optimization of multimodal neural networks under the scope of sentiment analysis, while advancing the understanding of modality imbalance as a key challenge in multimodal learning research.

1.4 Thesis Outline

This thesis is structured to provide a comprehensive exploration of the challenges and optimization techniques for multimodal neural networks:

- **Chapter 2** provides a theoretical background on machine learning and neural networks, focusing on key concepts like optimization and generalization of deep learning models.
- **Chapter 3** introduces multimodal machine learning, exploring fusion architectures and the challenges of training them, within the scope of multimodal sentiment analysis.
- **Chapter 4** introduces the modality imbalance phenomenon observed in multimodal learning, providing an in-depth analysis of its implications.
- **Chapter 5** presents a comprehensive overview of state-of-the art algorithms for multimodal optimization with a special focus on methods that dynamically adjust the gradients during training [26] [27] and multi-loss optimization approaches [20] [21].

- **Chapter 6** provides a comprehensive evaluation of dynamic optimization methods, detailing the key mechanisms, training patterns, and limitations of each dynamic optimization algorithm.
- **Chapter 7** concludes the thesis, summarizing the key findings and outlining potential directions for future work in the field of multi-modal optimization.

Chapter 2

Machine Learning

Machine Learning (ML) is a cornerstone of artificial intelligence, enabling systems to learn from data and improve performance autonomously. This chapter provides an overview of fundamentals, focusing on deep learning, neural network architectures, and the core principles that drive modern advancements in this field.

2.1 Introduction

While AI focuses on the simulation of human intelligence, including reasoning and decision-making, ML algorithms aim to learn autonomously from data. As defined by Tom M. Mitchell, "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" [53]. This highlights the main goal of machine learning: leveraging data and performance feedback to solve tasks effectively. Unlike traditional programming, ML systems do not require predefined rules, but employ algorithms to discover patterns and relationships autonomously.

Deep learning builds on these principles by employing artificial neural networks (ANNs) as its foundational structure. Defined by Ian Goodfellow as "a class of machine learning techniques that use hierarchical neural network architectures to model high-level abstractions in data" [54], deep learning excels at processing unstructured data such as images, text, and audio. Artificial Neural Network, inspired by the structure of the human brain, is the computational cell of deep learning, enabling systems to capture intricate patterns and dependencies. This capability has driven breakthroughs in tasks like image recognition, natural language processing and emotion recognition, indicating the impact of deep learning in addressing complex real-world challenges.

Towards a better understanding of machine learning, this chapter introduces the different types of learning paradigms. Following this, the role of neural networks as the computational backbone of many ML systems is discussed, with a focus on their architecture and mechanisms. The core concepts of loss functions and the backpropagation algorithm are then examined, as they form the foundation for training neural networks. An analysis of optimization techniques, which drive the learning process by minimizing the loss function, is included, followed by an exploration of model generalization and robustness. By understanding these components, we can address the challenges of training

robust and efficient systems, which is central to the focus of this thesis.

2.2 Types of Machine Learning

Learning algorithms can be classified into three main categories based on how they interact with data, whether rewards are given, feedback is provided, or labels are applied: Supervised Learning, Unsupervised Learning and Reinforcement Learning. In this section, we provide an overview of the main characteristics of each category, with a particular focus on supervised learning tasks.

2.2.1 Supervised Learning

Supervised algorithms involve learning the relationship between a set of input variables and an output variable based on a labeled training dataset [55]. In this approach, the learner is provided with an input $x \in \mathcal{X}$ and a corresponding output $y \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ represent the input and output spaces, respectively. The goal is to approximate a true mapping function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that links each input to its corresponding output. This is achieved by training a model on a dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, which consists of human-labeled input-output pairs. The goal is, by learning from these examples, to use new, unseen inputs to predict the values of the outputs. Supervised learning primarily focuses on two types of tasks based on the nature of output: regression and classification. In regression we predict quantitative outputs, while in classification qualitative outputs are predicted.

Classification is a process of categorizing data or objects into predefined categories according to their features or attributes. The main objective here is to build a model that can accurately assign a label or category to a new observation based on its features. A classification model is trained on a labeled dataset, where each sample is associated to one or more labels. There are two major categories of classification problems: Single-label and Multi-label classification. Single-label classification methods maps the output approximation to a unique target label out of a number of individual labels. This set of algorithms can be further divided into binary and multi-class classification based on the number of potential existing categories. We refer to binary classification when the input data can match to only one out of two target labels, while it is a multi-class problem when the input can be assigned to one among a pool of true labels. In multi-label classification problems the model can associate the input value with more than one target values, meaning that each sample can belong to two or more categories at a time. For example, an audio recording with a raised voice and fast pace could be assigned to two emotions at time, "Angry" and "Disgusted".

Regression is a supervised machine learning technique, that predicts the value of the dependent variable for new, unseen data. It models the relationship between the input features and the target variable, allowing for the estimation or prediction of real or continuous values. Figure 2.1 illustrates the fundamental difference between classification and regression in machine learning tasks. Classification groups data into predefined

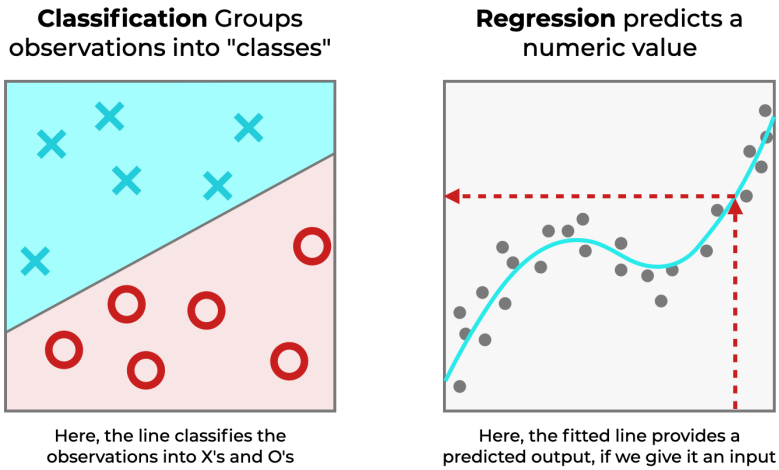


Figure 2.1. *Schematic difference between classification and regression. Picture was taken from [1]*

categories. Here, two classes are available, one marked with X and one with O. The classification algorithm creates a decision boundary to divide the data points into two distinct groups. On the other hand, the plotted curve on the right side formulates a relationship between the input and output samples. Then, the model attempts to predict a numeric outcome based on that relationship.

2.2.2 Unsupervised Learning

Unsupervised learning is another key paradigm of machine learning, distinguished by its use of unlabeled data. It has no interest in making predictions, because there is no associated output label [56]. The objective is to uncover hidden patterns, structures, or relationships within the data, without referring to any specific response. Clustering algorithms such as k-means algorithm partitions data into k clusters based on their spatial properties while Principal Component Analysis performs dimensionality reduction and transforms data into a set of orthogonal components, highlighting the most important features. By identifying underlying structures, unsupervised learning is fundamental in tasks such as feature extraction, making it ideal for exploratory data analysis and preprocessing.

2.2.3 Reinforcement Learning

In Reinforcement learning, an agent interacts with the environment, aiming to learn an optimal policy, by receiving feedback in the form of rewards or penalties, to make sequences of decisions [57]. Through repeated experiences, the agent tries to improve its strategy and maximize a cumulative reward over time. Q-learning algorithm [58] is a well-known model-free algorithm that learns the optimal action-value function that way.

2.3 Neural Networks

This section introduces the fundamental concepts of neural networks, beginning with their definition and expanding to specific architectures and their applications.

2.3.1 Definition of Artificial Neural Network

An Artificial Neural Network (ANN) is a machine learning model inspired by the human brain, designed to recognize patterns in data. ANNs consist of interconnected units called neurons, organized into layers: an input layer receives the data, one or more hidden layers process it, and an output layer provides the result. Each connection between neurons is assigned a weight, which is adjusted during training to minimize error. A neuron can be formulated mathematically as follows:

$$y = \phi\left(\sum_{j=1}^{n} w_j x_j - u\right) \tag{2.1}$$

In this equation, y represents the output of the neuron, while ϕ denotes the activation function. The term w_j corresponds to the synapse weight matrix, and x_j is the input vector for j-th input. Lastly, u refers to the activation threshold, often called the bias of the neuron.

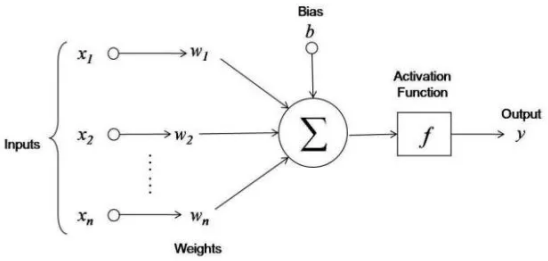


Figure 2.2. *Illustration of Artificial Neural Network. Source: [2]*

Activation Functions

Activation functions introduce non-linearity into the model, enhancing its capacity to capture complex relationships within the data.We introduce some of the fundamental activation functions commonly used in artificial neural networks:

Sigmoid [59]: The sigmoid function squashes the input to a value between 0 and 1, making it suitable for binary classification and is expressed as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

ReLU (Rectified Linear Unit) [60]: ReLU function outputs the input directly if it is positive, and 0 otherwise following:

$$\phi(x) = \max(0, x) \tag{2.3}$$

Tanh (Hyperbolic Tangent) [61]: Tahn function squashes the input to values between -1 and 1, making it a good choice for zero-centered data. It is expressed as

$$\varphi(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.4}$$

Softmax [62]: Softmax produces probabilities from logit outputs, an ideal concept for multi-class classification tasks. The softmax formulation:

$$\varphi(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{2.5}$$

Deep learning encompasses a variety of neural network architectures, each tailored to specific tasks and data types. Among these, Feedforward Neural Networks (FNNs), Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs) stand out as foundational models. Notably, Long Short-Term Memory (LSTM) networks, a specialized type of RNN, are particularly effective for capturing long-term dependencies, making them central to the models employed in this work.

2.3.2 Feed-forward Neural Networks

In a Feedforward Neural Network (FNN), the internal structure is arranged in sequential layers, where each neuron in one layer connects exclusively to all neurons in the next layer [63]. This topology rule excludes backward connections, found in many recurrent neural networks and layer-skipping. A notable example of feedforward network is the Multilayer Perceptron (MLP).

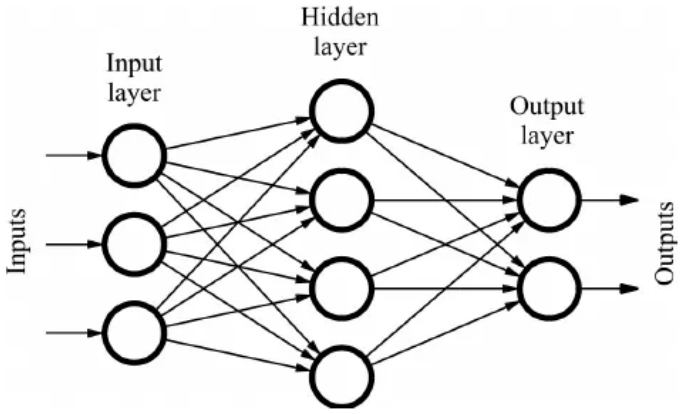


Figure 2.3. *Illustration of a Feed Forward Neural Network (FNN) architecture. Source: [3]*

2.3.3 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a type of deep learning model specifically designed to handle grid-like data, such as images. CNNs have become essential in tasks such as image classification, object detection, and even medical imaging. The architecture of CNNs is based on key concepts such as local connections, shared weights, pooling,

and multiple layers, all of which help capture spatial hierarchies in data [64]. More precisely, it comprises an input layer for raw data, convolutional layers that use filters, also known as kernels, to detect patterns or features, activation functions introducing non-linearity, pooling layers to reduce data dimensionality, and the final component, a fully connected layer, which maps the learned features to output predictions. CNN

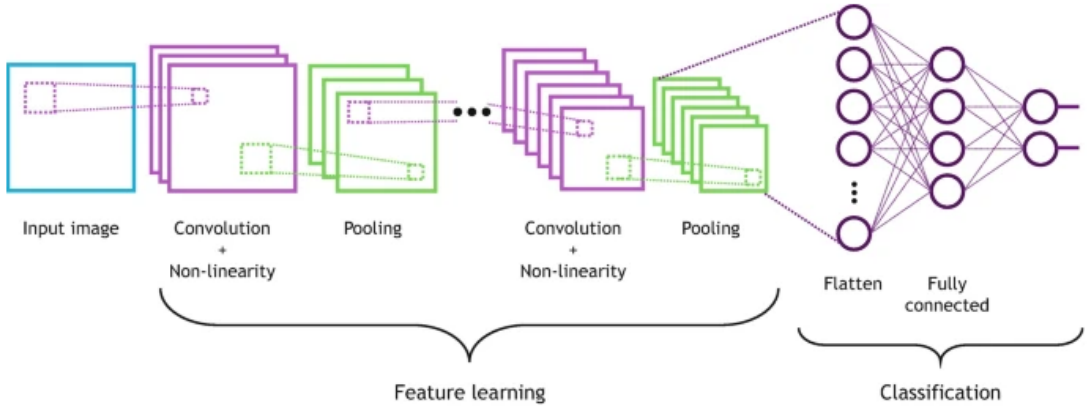


Figure 2.4. *Illustration of a Convolutional Neural Network (CNN) architecture. The first part, using convolution operations, performs feature learning. The features are then flattened and fed into a set of fully connected layers to perform the classification or the regression task. Source: [4]*

performance is enhanced by stride and padding, which control how filters move across the input, and pooling, which simplifies data by summarizing regions while retaining important information. These concepts enable CNNs to effectively process large, high-dimensional data with fewer parameters compared to fully connected networks.

2.3.4 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are designed to process sequential data by retaining information across time steps. Unlike feedforward networks, RNNs have connections that allow information to persist across time, making them ideal for tasks involving time series and sequential data. Their architecture is based on an input layer, a hidden layer, and an output layer. The hidden layer has recurrent connections, allowing information from previous time steps to influence the current time step. The hidden state at each time step t is expressed as follows [65]:

$$H_t = \phi_h(X_t W_{xh} + H_{t-1} W_{hh} + b_h) \tag{2.6}$$

The output is computed:

$$O_t = \phi_o(H_t W_{ho} + b_o) \tag{2.7}$$

In this context, H_t represents the hidden state at time step t, while X_t denotes the input vector at the same time step. The term W_{xh} refers to the weight matrix between the input and the hidden state, and H_{t-1} corresponds to the hidden state from the previous time step. Similarly, W_{hh} represents the weight matrix between hidden states, and b_h is the

bias term for the hidden state. The activation function applied to the hidden state is denoted by ϕ_h, and common examples include tanh and ReLU. Furthermore, O_t represents the output at time step t, with W_{ho} as the weight matrix between the hidden state and the output. The term b_o is the bias term for the output, and ϕ_o represents the activation function applied to the output. The diagram illustrates a Recurrent Neural Network

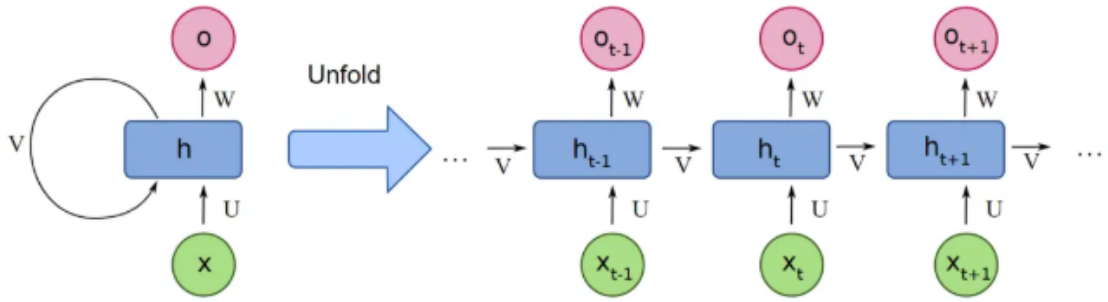


Figure 2.5. *Illustration of a Recurrent Neural Network (RNN) architecture. Source: [5]*

(RNN) in both its compact and unfolded forms. At each time step t, the network processes input x_t to update the hidden state h_t, which retains past information through recurrent connections. This structure enables the network to capture sequential dependencies efficiently.

2.3.5 LSTM architecture

The Long Short-Term Memory (LSTM) architecture, first introduced by [66], consists of memory blocks connected in a recurrent fashion. Each memory block is composed of memory cells and three types of gates: the input gate, the forget gate, and the output gate [7]. The input gate manages the entry of new information into the memory cell. The forget gate controls the portion of stored information to be erased, ensuring that irrelevant or outdated information is discarded. The output gate is responsible for providing the stored information from the memory cell when the output must be computed. At the core of the LSTM architecture lies the memory cell, which serves as the central unit. It stores information and determines whether to reject, retain, or retrieve it based on the gate mechanisms, allowing LSTMs to effectively handle long-term dependencies in sequential data. The LSTM unit, as illustrated in Figure 2.6, consists of several key components that work together to handle sequential data effectively. The *forget gate* controls which parts of the previous cell state (C_{t-1}) should be retained, using the formula:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.8}$$

Next, the *input gate* determines how much of the new input should contribute to the cell state, calculated as:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.9}$$

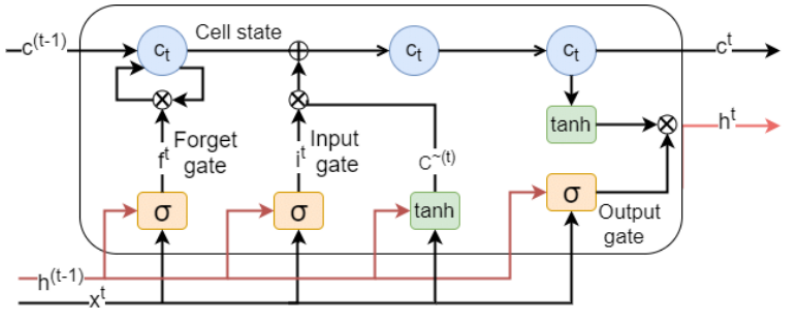


Figure 2.6. *Schematic representation of Long Short-Term Memory unit cell. Source: [6].*

To update the cell state, a *candidate cell state* is computed, providing a new candidate value for the cell state (C_t) through the equation:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.10}$$

The updated cell state combines the forget gate and input gate outputs, following the rule:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.11}$$

The *output gate* determines the information to be output based on the current cell state, given by:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.12}$$

Finally, the *hidden state* is computed, representing the output of the LSTM at time step t, using the equation:

$$h_t = o_t \odot \tanh(C_t) \tag{2.13}$$

In these equations: x_t represents the input at time step t, while h_{t-1} and C_{t-1} are the hidden and cell states from the previous time step, respectively. The functions σ and tanh denote the sigmoid and hyperbolic tangent activation functions, respectively, while $\odot$ represents element-wise multiplication. The weight matrices W_f, W_i, W_C, W_o and biases b_f, b_i, b_C, b_o are learned parameters that adapt during training to optimize the LSTM's performance.

LSTMs exceed traditional RNNs because of their ability to control the flow of information with their gates and capture dependencies over long sequences through their memory cells. Let us consider the illustration of Figure 2.7. The state of the input, forget, and output gates are displayed below, to the left and above the hidden layer respectively. For simplicity, all gates are either entirely open ('O') or closed ('—'). Black nodes are sensitive to the inputs, while white ones are insensitive. The sensitivity of the output layer can be switched on and off by the output gate without affecting the cell. In other words, if the input gate is open, but the forget gate is closed, the cell will receive new information from the current input and update its state, while still retaining relevant information from the past. This advantage makes LSTMs a powerful tool to handle long-term dependencies in sequential data in tasks like speech recognition and sentiment analysis.

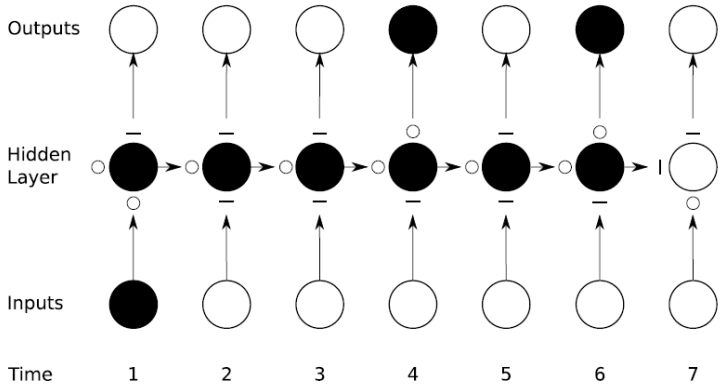


Figure 2.7. *Preservation of gradient information by LSTM as represented in [7].*

2.4 Loss Function

Focusing on supervised learning, this section introduces the fundamental concept of loss . The loss function represents the difference between the predicted output of a model and the true target value. Loss quantifies the error and guides the optimization of the model parameters to minimize the error. The loss function compares the output y_t with the corresponding target $\hat{y}_t$ at time step t and is defined as:

$$L(y, \hat{y}) = \sum_{t=1}^{T} L(y_t, \hat{y}_t) \tag{2.14}$$

The loss at time step t is expressed as $L(y_t, \hat{y}_t)$, and T corresponds to the total number of time steps. This equation represents the overall summation of losses at each time-step. The choice of loss function plays a crucial role in the learning process and is considered problem dependent. We present loss functions popular in literature for classification and regression tasks [67].

Binary Cross Entropy Loss [68]: Binary Cross Entropy Loss measures the performance of a binary classification model by quantifying the difference between predicted probabilities and true binary labels. It penalizes confidently incorrect predictions more heavily, making it sensitive to prediction confidence. The log-loss function is defined as:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \tag{2.15}$$

Here, N is the number of samples, y_i is the true label (0 or 1), and $\hat{y}_i$ is the predicted probability of class 1 for the i-th sample. Binary cross-entropy can be extended [59] by applying the sigmoid activation function to raw logits before calculating the loss. It is particularly suited for multi-label classification tasks, where a single sample can belong to multiple classes. The sigmoid cross-entropy loss for one observation is:

$$L(y_i, \hat{y}_i) = -\left(y_i \log(\sigma(\hat{y}_i)) + (1 - y_i) \log(1 - \sigma(\hat{y}_i))\right) \tag{2.16}$$

Here, $\sigma(\hat{y}_i) = \frac{1}{1+e^{-\hat{y}_i}}$ maps logits to probabilities. It treats each label independently, making it ideal for tasks with overlapping class memberships.

Cross-Entropy Loss [62]: Designed for multi-class classification, this loss applies the softmax function to logits to produce a probability distribution across all classes and measures the divergence from the true one-hot encoded labels. It is defined as:

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log\left(\frac{e^{\hat{y}_i}}{\sum_{j=1}^{C} e^{\hat{y}_j}}\right) \tag{2.17}$$

Here, C is the total number of classes, y_i is the true label (1 for the correct class, 0 otherwise), and the softmax function ensures the predicted probabilities for all classes sum to 1. This loss formulation is ideal for multi-class problems.

Mean Squared Error (MSE): One of the most common loss functions for regression tasks, it calculates the average of the squared differences between the predicted and true values. It is particularly useful for penalizing larger errors more severely due to squaring the error. The Mean Squared Error (MSE) is given by:

$$L(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.18}$$

where N is the number of observations, y_i represents the actual value for the i-th observation, and $\hat{y}_i$ denotes the predicted value for the i-th observation.

Mean Absolute Error (MAE): It computes the average of the absolute differences between predicted and true values, making it also really popular among regression problems. Mean Absolute Error is more robust to outliers compared to Mean Squared Error. The Mean Absolute Error (MAE) is given by:

$$L(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.19}$$

where N represents the number of observations, y_i is the actual value for the i-th observation, and $\hat{y}_i$ is the predicted value for the i-th observation.

In multi-task learning [69], the model performs twice or more related tasks at once. The total loss in this case is expressed as a weighted sum of the individual losses from each task as shown below:

$$L_{\text{total}} = a_1 L_1 + a_2 L_2 \tag{2.20}$$

Where L_1 and L_2 are the loss functions for Task 1 and Task 2, respectively, and a_1 and a_2 are the weights assigned to the respective tasks. Each loss contributes differently to the overall optimization, but by optimizing this total loss, the network learns to perform both tasks simultaneously.

2.5 Backpropagation Algorithm

As described in [28], during forward step, the input data is passed through each network layer. Each layer computes the weighted sum of its inputs and applies the

activation function. Then it passes the result to the next layer, eventually generating the output. The error is then calculated based on how far the predicted output deviates from the actual target value. In the backward step, the algorithm begins at the output layer and moves backward toward the input layer, systematically updating the weights to reduce the error following the *gradient descent algorithm*. At each layer, the gradient of the error with respect to the weights is calculated to determine how much each weight contributes to the overall error. This weight adjustment is determined by applying the *chain rule* to compute how the error at the output propagates through each layer of the network. Specifically, for each weight, the gradient of the error is computed as the product of the partial derivatives of the loss function with respect to the output, the output with respect to the activation, and the activation with respect to the weights. Mathematically, for a given weight w, the gradient is:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial w} \tag{2.21}$$

where L is the loss, y is the output of the network, z is the weighted input to the activation function, and w is the weight. This chain rule application ensures that the influence of the weights at each layer is properly accounted for in the gradient computation. In backpropagation, biases are updated alongside weights by calculating their gradients, allowing the model to adjust the activation threshold of neurons and shift the output to better fit the data.

2.5.1 Gradient Descent Algorithm

The equation provided represents the weight update rule of the gradient descent algorithm, which is integral to the backpropagation process. Specifically, it adjusts the weights Δw by scaling the gradient of the loss function $\frac{\partial L}{\partial w}$ with respect to the weights using a learning rate η.

$$\Delta w_{ij} = -\eta \frac{\partial L}{\partial w_{ij}} \tag{2.22}$$

This iterative process minimizes the error by guiding the network toward a set of weights that reduce the overall loss. In the context of backpropagation, gradient descent acts as the optimization mechanism that updates the weights during each iteration based on the error gradients propagated backward through the network. The process of forward and backward pass is repeatedly executed for many epochs and the weights are continuously updated in small steps until the error converges to a minimum. The most important hyperparameter is the learning rate η as it controls the extent to which the model parameters are adjusted concerning the loss gradient, thus the selection of the learning rate hyperparameter is a crucial point.

2.5.2 Challenges of Backpropagation

One main challenge of the back-propagation is the Exploding Gradient Problem, where gradients grow exponentially making the learning process unstable. Another issue is

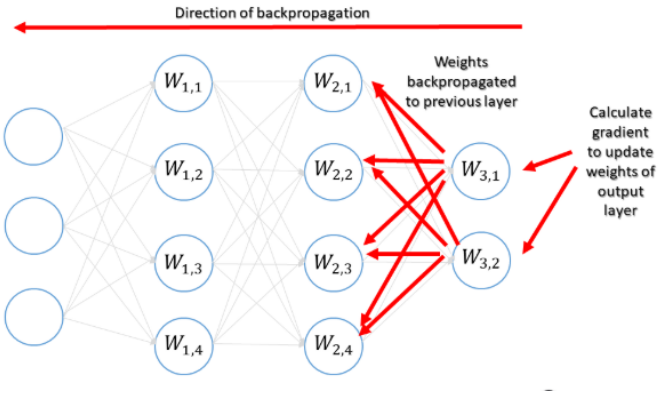


Figure 2.8. *Illustration of the backpropagation algorithm in supervised training. The diagram shows the direction of error propagation (red arrow) from the output layer to the input layer, along with the calculation of gradients used to update weights in each layer. Image taken from [8]*

Vanishing Gradient Problem, where gradients can become very small as they propagate backward through many layers, causing the learning process to slow down [70].

The Exploding Gradient Problem indicates a phenomenon where gradients grow exponentially through layers. As gradients increase, the weight updates are getting larger, preventing convergence. One effective way to handle exploding gradients is Gradient Clipping. A threshold value is set for the gradient magnitude. If the computed gradients surpass this value, they are scaled down to prevent very large updates. Thus, networks where long sequences are present, can be protected from the phenomenon of exploding gradients. The gradient clipping formula is given by [71]:

$$g \leftarrow \frac{g}{\|g\|} \times \text{threshold} \quad \text{if } \|g\| > \text{threshold} \tag{2.23}$$

where g represents the gradient vector, $\|g\|$ denotes the norm of the gradient vector, and the threshold specifies the maximum allowable value for the gradient norm.

During backpropagation, gradients are calculated using the chain rule of calculus, which multiplies the gradients of each layer as we move from the output layer back to the input layer. During this multiplication gradients can shrink exponentially as they propagate through each layer. If the gradients become too small, the weights will barely change during backward pass. The weight updates for the layers closer to the input will become negligible. Thus, earlier layers may freeze during the training process, preventing later layers to learn feature patterns effectively.

The choice of initial weights and activation function can address the problem of vanishing/exploding gradients [72] [73]. Batch normalization [74] can also mitigate such problems. It acts as a scaling method that make the inputs of each layer have zero mean value and unit variance. Thus, gradients maintain a consistent scale leading to a more stable training process. The addition of a regularization term [59] such as L2 regularization to weights can also be helpful in controlling the size of gradients during backpropagation. This term represents a penalty applied to large weights in order to

reduce their magnitude. To address the issue of vanishing gradient more effectively, Long-Short Term Memory Networks (LSTMs) [75] were introduced. LSTMs remain one of the most prominent architectures for capturing long-term dependencies in sequential data.

2.6 Optimization of Neural Networks

Optimization refers to the process of adjusting the parameters of a model in order to minimize the loss function L. Primary goal of an optimization technique is to find the best set of parameters that allow the model to perform well on training data as well as on new, unseen data. Based on the kind of parameters they aim to improve, optimization techniques can be divided to: Weight Optimization Methods, Gradient-free Optimization Methods, Constrained Optimization, Regularization-base Optimization.

Searching through literature we can further divide weight optimization methods into two categories: First-order Optimization Methods and Second-order Optimization Methods. First-order methods rely on calculating the first-order derivatives of loss with respect to each model parameter, while second-order optimization algorithms rely on second-order derivatives. Although, second-order methods can converge faster theoretically, calculations of the Hessian matrix are time and memory expensive when dealing with large parameter spaces such as in LSTMs. Also, first-order methods scale better with large dataset. With these factors in mind, in this section we represent some of the most notable first-order optimization techniques.

Stochastic Gradient Descent: Unlike full gradient descent, Stochastic Gradient Descent (SGD) [76] is an optimization algorithm that updates the model parameters based on the gradient of the loss computed for a single or a few data points (batch of data) at each iteration. This method leads to faster updates, thus faster convergence, becoming this way a good optimization choice for large datasets. The update rule for SGD is given by:

$$\partial_{t+1} = \partial_t - \eta \nabla_{\partial} L(\partial_t; \hat{y}_i, y_i) \tag{2.24}$$

where ∂_t represents the parameters at iteration t; η is the learning rate, controlling how large each update is; and $\nabla_w L(w_t; \hat{y}_i, y_i)$ and $\nabla_b L(b_t; \hat{y}_i, y_i)$ are the gradients of the loss function L with respect to the weights and biases, computed using a single data point $(\hat{y}_i, y_i)$ or a batch of data points. Stochastic Gradient Descent has been proved really efficient for large datasets, as it updates models parameters for each training sample or mini-batch of the dataset. Each point or batch is selected randomly, which makes it stochastic. The stochastic nature of SGD introduces noise to data, which can help the model escape local minima.

Momentum: Momentum [77] is variation of Stochastic Gradient Descent that uses a moving average of the gradients to smooth the path of parameter updates. The update rule is:

$$v_{t+1} = \beta v_t + \nabla_\partial L(\partial_t), \tag{2.25}$$

$$\partial_{t+1} = \partial_t - \eta v_{t+1}. \tag{2.26}$$

where v_t represents the velocity (or momentum) at step t, which smooths the updates and avoid oscillations; v_{t+1} is the updated velocity term at step $t+1$; and β is the momentum coefficient, controlling the contribution of previous updates to the current step.

AdaGrad: Adagrad [78] scales the learning rate for each parameter based on the sum of past squared gradients. The update rule is:

$$\partial_{t+1} = \partial_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \nabla_\partial L(\partial_t) \tag{2.27}$$

Where G_t is the sum of the squares of the past gradients, and ϵ is a small constant added to avoid division by zero.

Adaptive moment estimation (Adam): Adam [42] maintains both an exponentially decaying average of past gradients and squared gradients. The update rules are:

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla_\partial L(\partial_t), \tag{2.28}$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2)(\nabla_\partial L(\partial_t))^2, \tag{2.29}$$

$$\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \beta_1^t}, \tag{2.30}$$

$$\hat{v}_{t+1} = \frac{v_{t+1}}{1 - \beta_2^t}, \tag{2.31}$$

$$\partial_{t+1} = \partial_t - \frac{\eta \hat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}} + \epsilon}. \tag{2.32}$$

Here, m_t represents the first moment (mean of the gradients) at step t, v_t is the second moment (variance of the gradients) at step t, and β_1, β_2 are the exponential decay rates for the first and second moments, respectively. The terms $\hat{m}_{t+1}$ and $\hat{v}_{t+1}$ are the bias-corrected estimates of the first and second moments, which are used to adjust the learning rate dynamically. The constant ϵ is a small value added to ensure numerical stability in the denominator. Adam dynamically adjusts the learning rate for each parameter based on the first and second moments of the gradients (i.e., the mean and variance). This helps the optimizer to handle varying feature scales and gradients, which is common when multiple modalities are present, as they behave differently.

2.7 Model Generalization

Generalization refers to the ability of a model to make successful prediction when it receives new, unseen data that have a similar however distribution as the training data. Generalization is really crucial for a neural network, as it ensures that the model has actually learned, not just memorized the training data, and is capable to recognize

the patterns in unseen entries. The goal is to balance fitting the training data well while preventing the model from memorizing them. While minimizing the training loss is crucial, trying to minimize the generalization error is also really important. By the term generalization error, we refer to the difference between the model's performance on the training set and the testing set as illustrated in Figure 2.9. Generalization ability is highly influenced by the terms of bias and variance components in errors.

Total error of a model can be broken down into three main components: bias, variance, and irreducible error. The total error can be represented as:

$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error} \tag{2.33}$$

Let $f(x)$ represent the true function, $\hat{f}(x)$ the predicted function, and D the training data. The expected squared error for a prediction is given by:

$$E[(f(x) - \hat{f}(x))^2] = (f(x) - E[\hat{f}(x)])^2 + E[(\hat{f}(x) - E[\hat{f}(x)])^2] + \sigma^2 \tag{2.34}$$

The term $(f(x) - E[\hat{f}(x)])^2$ represents the bias, which measures the error introduced by approximating the true function $f(x)$ with the expected prediction $E[\hat{f}(x)]$. The term $E[(\hat{f}(x) - E[\hat{f}(x)])^2]$ represents the variance, quantifying the variability of the model's predictions around their expected value. Finally, σ^2 denotes the irreducible error, which captures the inherent noise in the data that cannot be explained by the model.

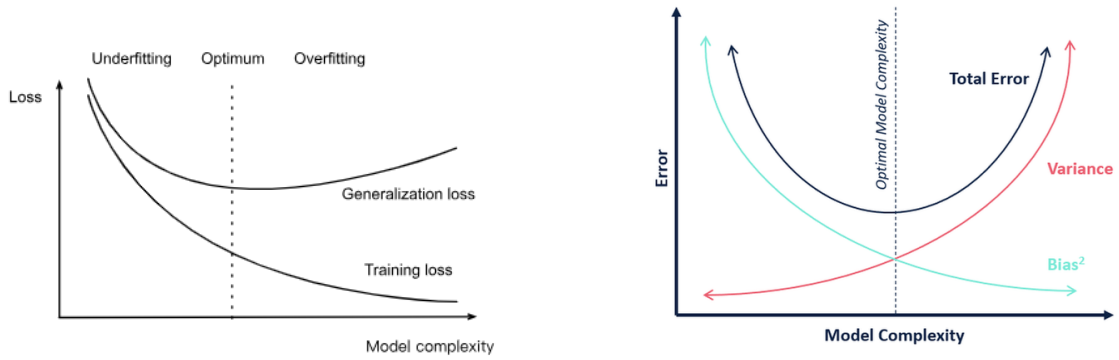


Figure 2.9. *Illustration of generalization error. The left figure shows generalization error as the difference between training and test error, with the underfitting region on the left and overfitting on the right. The right figure illustrates the relationship between bias, variance, total error, and model complexity, where the optimal model complexity lies between underfitting and overfitting. Sources: [9], [10].*

2.7.1 Underfitting and Overfitting

Bias arises when the model fails to capture important data patterns considering data to be extremely simple. High bias can lead to the phenomenon of underfitting. On the other hand, when a model is sensitive to even the smallest fluctuations in training data, we speak about variance. Having high variance can lead to overfitting, where the model

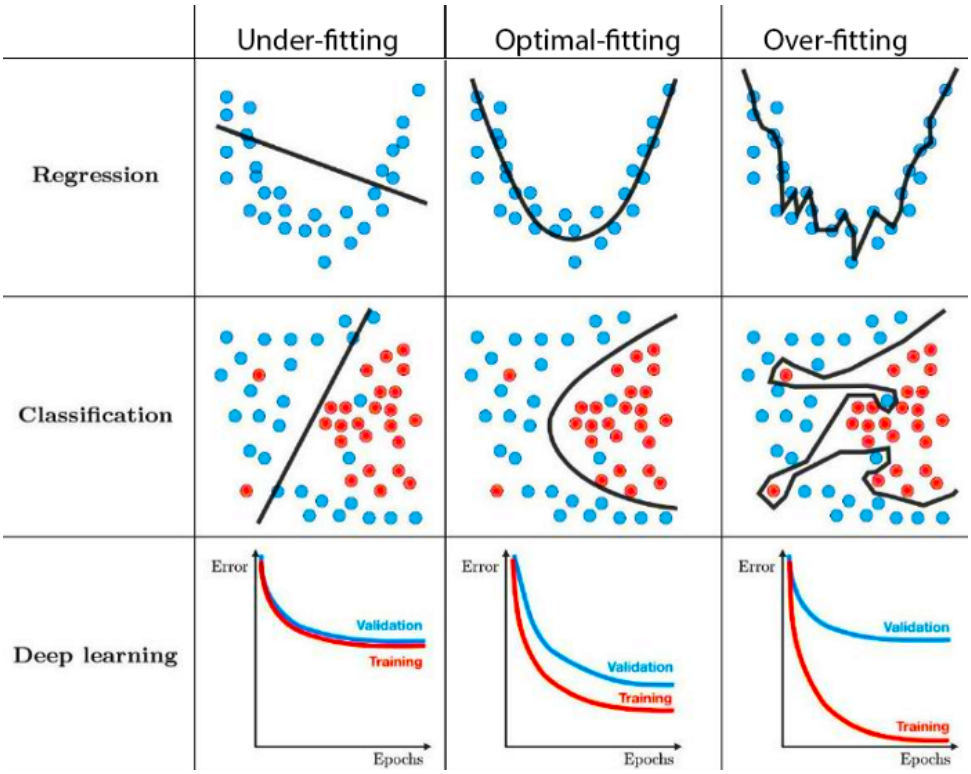


Figure 2.10. *Comparison of underfitting, optimal fitting, and overfitting across regression, classification, and deep learning models. Source: [11]*

memorize the training data leading to poor performance on unseen data. Overfitting is common among complex models, for example models with a large number of parameters, because they suffer from increased variance. Performance in simpler models may be reduced due to higher bias and lower variance, leading to underfitting as seen in Figure 2.9. The trade-off between bias and variance is a challenge in machine learning [79]. Balancing bias and variance is essential for minimizing the total error and ensuring that the model can generalize well, avoiding both overfitting and underfitting.

There are various techniques proposed in literature to handle these phenomena. Cross-validation [80], regularization techniques [81], dropout [82], data augmentation [83], proper tuning of the hyperparameters have been applied over the years. Here, we focus on another notable technique, early stopping [51]. It is a common practice in models with a large number of parameters, especially when there is no clear knowledge of the number of epochs needed to achieve a good generalization performance.

Early stopping evaluates the performance of the model on a validation set. The validation set is a subset of the data, separate from the training set. As training progresses, the error on the validation set is calculated at regular intervals alongside the training loss. At first, both losses decrease, but after a certain point the training loss continues to decrease while the validation is rising. When the validation error starts to increase, early stopping interrupts the training of the model as illustrated. Thus, the parameters are captured before overfitting occurs.

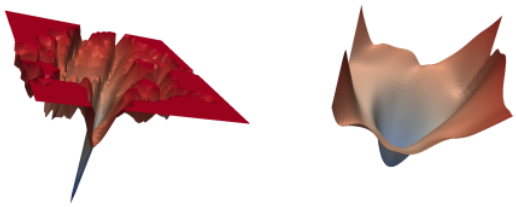


Figure 2.11. *(left) A sharp minimum to which a trained model converged. (right) A wide minimum to which a trained model converged. Image taken from [12]*

2.7.2 Role of Loss Landscape in Generalization

In understanding the role of the loss surface in optimization and generalization, it is critical to consider the geometry of the minima as illustrated in Figure 2.11. Local minima and saddle points in the loss surface can hinder backpropagation, causing the algorithm to get stuck in a local minimum of the error function rather than finding its global minimum. The algorithm struggles to move the weights in the opposite direction of the gradient and thus minimize the error, resulting in poor performance.

Generalization performance has a strong connection to the geometry around the minimizers—sets of parameter values that minimize the loss function. Sharp minimizers are surrounded by steep slopes. Thus, each small change in the model parameters can increase drastically the loss, leading to poor generalization ability. Flat minimizers on the other hand are associated with better generalization.

The role of weight decay in influencing the sharpness or flatness of minimizers has been investigated in [84]. Weight decay is a regularization technique commonly used in machine learning to prevent overfitting by penalizing large weight values during training. The loss function is augmented with an additional term that discourages large weights, typically by adding a penalty proportional to the L2 norm of the weights. This modifies the objective function to:

$$L_{new} = L_{original} + \lambda \sum w_i^2 \tag{2.35}$$

where λ is the regularization strength, and w_i represents the weights. Weight decay is closely related to L2 regularization, where the sum of the squared weights is added to the loss function. Adding weight decay significantly alters the geometry of the loss landscape, often leading to flatter minima and better generalization, especially when using large batch sizes [84].

2.8 Summary

This chapter provided an overview of the foundational principles of machine learning, including the key paradigms, neural network architectures, and the critical components of training such as loss functions, optimization techniques, and generalization. These principles set the stage for developing robust and efficient machine learning models. In

transitioning to multimodal machine learning, these challenges become more difficult to mitigate. As we move into Chapter 3, we study multimodal machine learning with a focus on sentiment analysis tasks, examining how these foundational principles evolve in the context of multimodal interactions and optimization.

Chapter 3

Multimodal Machine Learning

Advancements in machine learning have shift interest to real-world data, highlighting their multimodal nature, as they combine diverse sources of information that collectively provide a richer understanding.

3.1 Introduction

Multimodal machine learning focuses on the integration of such data types to enable a more human-like machine intelligence. A modality refers to a kind of information representing a specific aspect of a phenomenon. According to [85] a modality refers to the way in which something happens or is experienced, often corresponding to a specific sensor, input type, or data format. For instance, by the term of text modality we refer to data describing linguistic and semantic information. The need of integrated data arise from

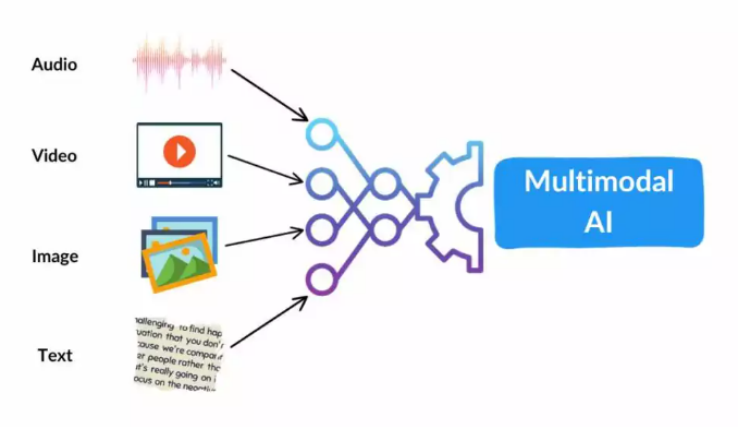


Figure 3.1. *Illustration of a Multimodal AI system, integrating multiple data modalities—audio, video, image, and text—into a unified neural network. Source: [13]*

the weakness of unimodal systems to capture the full spectrum of information needed for complex tasks. Understanding human behavior in a video can be approached by a analyzing the visual cues present in the video. However, by incorporating the contextual text or spoken language, the system can enhance performance and capture determinant insights for the task. Multimodal machine learning aims to combine different modalities to achieve a nuanced understanding, managing to handle even missing data by leveraging only available modalities, a common case in real-world scenarios.

Multimodal machine leaning involves algorithms that can represent each modality effectively, model the relationships between modalities, capture complementary information through integration, predict outcomes that leverage information from all modalities. Multimodal Neural Networks integrate various modalities like audio, image, and text, to tackle complex challenges in fields such as video understanding [86], Visual Question Answering (VQA) [87], image captioning [88], multimodal classification [89], and cross-modal retrieval and verification [90] [91] [92], finding application across various domains including healthcare, human-computer interaction, autonomous vehicles, sentiment analysis.

This chapter aims to provide theoretical foundations of multimodal machine learning, alongside with practical implementations. It addresses key challenges and introduces neural networks architectures, tailored for multimodal training, setting the stage for understanding the employment of multimodal neural networks to sentiment analysis and emotion recognition.

3.2 Multimodal Data Processing: Principles and Challenges

Multimodal data introduces several complex challenges, requiring models to process multiple data streams simultaneously. These challenges arise primarily from the heterogeneity across modalities, as each varies in structure, representation, and noise levels. For instance, text is sequential and discrete, while images are spatial and continuous. Addressing heterogeneity involves designing representations that preserve the unique features of each modality while allowing cross-modal interactions. Understanding heterogeneity helps mitigate biases and modality-specific noise [14].

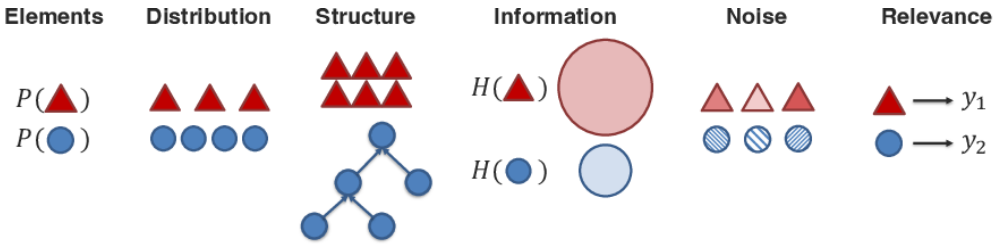


Figure 3.2. *The figure depicts how different data elements (represented as red triangles and blue circles) vary in their distribution patterns, hierarchical structures, and information content. Noise is introduced through less relevant or distorted data points, while relevance indicates how different elements contribute to distinct target outputs. This visualization highlights the challenges of handling heterogeneous data in machine learning models. Source: [14]*

Despite this diversity, modalities often have complementary, shared or correlated information, creating meaningful connections to support each other. By dynamically interacting, modalities create enriched, context-aware insights that surpass the capabilities of any single modality. Successfully managing these principles is crucial for building models capable of leveraging both the unique strengths of each modality and the combined insights from their interactions, unlocking the full potential of multimodal data [14].

By fusing modalities, a joint representation aims to capture cross-modal interactions

between elements of each modality. Unlike fusion strategies, coordination [85] focuses on the interchange of cross-modal information between modalities. This approach preserves the original number of modality-specific representations, each enriched with contextual data from other modalities. Rather than merging information into a single, unified representation as in fusion, fission creates multiple, independent representations that capture details and reveal internal structures within each modality, by breaking down modalities into more granular representations. In our setup, fusion is preferred as it provides a straightforward, integrated representation that combines complementary information from each modality, while allowing the investigation of learning dynamics for both the shared representation and each modality individually.

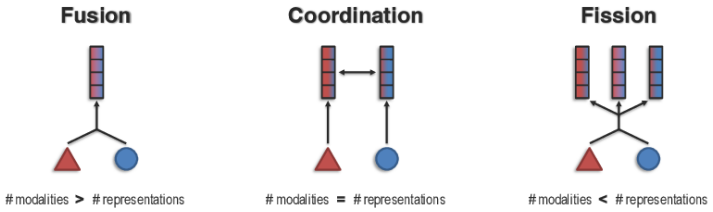


Figure 3.3. *Illustration of Fusion, Coordination, and Fission paradigms in multimodal representation learning, depicting different relationships between the number of modalities and learned representations. Source: [14]*

Alignment synchronizes elements across modalities discretely, such as spoken words with tone pitches in video, and can be local, referring to specific pairs of elements, or global, within the entire dataset [93] [94]. Continuous alignment applies to data without clear breaks, such as video streams [95]. In addition to alignment, structural reasoning plays a crucial role in interpreting multimodal data. These mechanisms not only facilitate the integration of diverse information but also enhance the ability to draw meaningful insights from complex multimodal datasets [96] [97].

Having explored some of the fundamental challenges and principles in multimodal data processing, the next section presents popular multimodal architectures designed to effectively integrate and leverage these diverse modalities for enhanced learning.

3.3 Multimodal Architectures

In this section, we discuss networks architectures commonly used in multimodal processing, focusing on fusion methods of modalities and Long Short-Term Memory (LSTM) networks, which are central to our experiments. Pioneering works in this area include fusion methods for integrating diverse modalities, such as Ngiam et al. (2011) [98], which introduced joint learning across modalities in deep networks, and Srivastava and Salakhutdinov (2012) [99], who demonstrated the use of Deep Boltzmann machines for multimodal fusion across text and images. Wöllmer et al. (2013) [100] applied LSTMs to audio-visual datasets, advancing sentiment and emotion recognition tasks by capturing long-range dependencies across modalities.

3.3.1 Fusion Strategies in Multimodal Learning

One of the main challenges in multimodal processing is how to effectively combine information originating from a different stream of data into a coherent representation. Fusion techniques are typically classified in Early, Late, Hybrid and Intermediate fusion.

Early Fusion: In early fusion, features from each modality are combined at the input level, following a joint process by a single network. This is why early fusion is called feature-level fusion in other words. Early fusion is helpful in cases where different modalities contain complementary information, as it can capture cross-modal interactions across them [85]. This, however, can lead to high-dimensional feature space. For example, concatenation of raw pixels and audio spectrograms can lead to very large input vectors making the model more challenging to handle and increasing computational requirements.

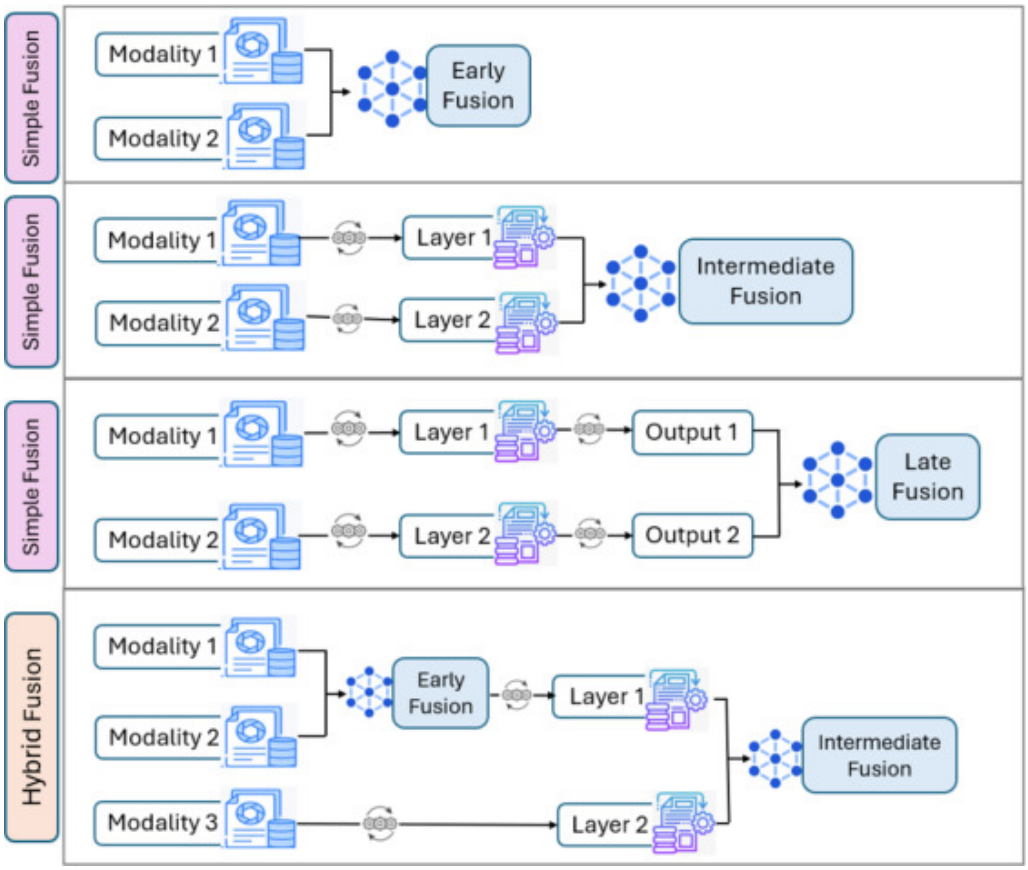


Figure 3.4. *Schematic representation of multimodal fusion strategies. Early, Late and Intermediate fusion take place only in one stage of the topology, while modalities in hybrid fusion can be integrated at various stages. Source: [15]*

Late Fusion: In fusion with abstract modalities, features from each modality are processed independently by networks. Then, their outputs are combined at a later stage, typically before the final prediction layer that performs the task. Late fusion provides each network with the ability to specialize in a specific modality. For instance, an LSTM can process text to capture language patterns, while a CNN can process visual data to detect facial expressions. This way, each network uses the characteristics of each modality at

its maximum. However, late fusion models often lack of the ability to exploit nuanced interactions between modalities [101].

Hybrid Fusion: Hybrid fusion combines aspects of both early and late fusion, allowing for interactions between modalities at multiple levels of the network. In hybrid fusion, modalities are processed separately at first, then partially merged at specific points in the network, and may continue processing independently until the final fusion layer. This approach aim to capture both low-level and high-level cross-modal relationships and can be particularly helpful in cases where modalities need specialized processing but still benefit from periodic cross-modal interactions [102].

Intermediate Fusion: This approach includes one main fusion point at a middle layer within the network topology. Each modality is processed individually in the initial layers and then combined with the others at an intermediate stage [103]. After the fusion, the network typically continues with the fused representation. Here stands the main difference between hybrid and intermediate fusion. While in hybrid fusion there might be multiple fusion points throughout the network, allowing modalities to interact at several stages and even re-separate from the others before the final layer, in intermediate fusion the model retains the fused representation until the final layer.

3.3.2 Attention Mechanisms in Multimodal Learning

Self-attention mechanisms [16] are critical components in multimodal architectures allowing models to focus on the most relevant aspects of each modality. Self-attention mechanisms can be applied within each modality independently before the fusion stage, capturing effectively intra-modality dependencies. Multimodal Transformers [17] extend this approach through cross-attention layers that model complex interactions between modalities. Cross-attention focuses on interactions between modalities by aligning features from one modality based on features from another. For instance, in a text-audio fusion task, the model can use cross-attention to focus on specific words when analyzing audio cues and vice versa. Modality-Specific Attention offers attention scores for each modality separately and combines them only at a later stage. This approach has been adopted in scenarios where one modality might be more informative than others under different contexts as for example in [38]. Co-attention mechanisms allow each modality to inform the attention map of other modalities. This mutual approach is often used in visual question answering (VQA) applications [104].

3.3.3 Temporal Dynamics: LSTMs and Sequential Processing

Long Short-Term Memory networks are widely used in multimodal applications due to their ability in handling sequential data. LSTMs manage to capture long-term dependencies, which is extremely important when multiple modalities like text, audio and video are present simultaneously. In multimodal neural networks, LSTMs can be used in various fusion strategies each with unique advantages for different types of multimodal integration.

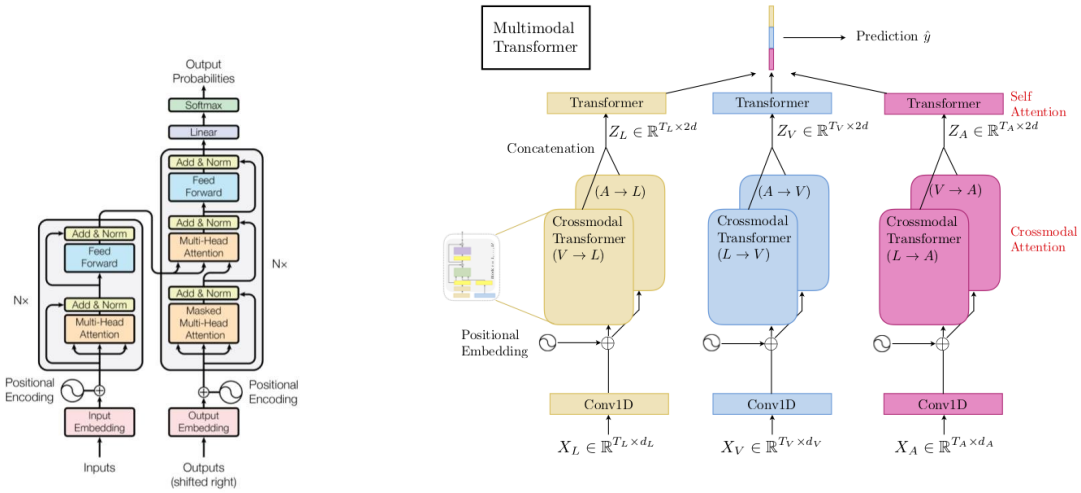


Figure 3.5. *Schematic illustration of a standard Transformer model (left) and a multimodal transformer (right). The standard Transformer architecture consists of an encoder-decoder structure with multi-head attention and feed-forward layers, widely used for sequence-to-sequence tasks such as natural language processing [16]. MulT architecture [17] processes multiple input modalities (e.g., text, image, and audio) through independent subnetworks before applying cross-modal attention mechanisms to fuse information and make predictions.*

In their work [98] Ngiam et al. demonstrated early fusion with LSTMs in a Audio-Visual Speech Recognition Framework that uses audio spectrograms and visual features at the input stage. Zadeh et al. [38] proposed DFG, where separate LSTMs process text, audio, and visual data individually, and their outputs are combined in a dynamic late fusion layer.In [105], a foundational approach to multimodal emotion recognition is presented, where LSTM networks process audio and visual modalities separately before combining them at an intermediate fusion layer. All the aforementioned frameworks highlight the beneficial choice of LSTMs in multimodal architectures, particularly those focused on sentiment analysis and emotion recognition, as explored in this thesis.

In late fusion architectures each modality is processed separately, allowing each encoder to focus in capturing unique features without interference from the other modalities. Understanding the flow of information over time is important in multimodal tasks like sentiment analysis, where LSTMs excel at handling sequential data, capturing both short-term and long-term dependencies within each modality. For example, LSTMs can capture sequential language cues in text, tonal shifts in audio, and gesture dynamics in visual data, all of which are crucial for understanding sentiment and emotions. Late fusion with LSTMs, also, allows each modality to be optimized separately before fusion, making the final sentiment or emotion classification more manageable and interpretable. This approach aligns with a primary goal of this thesis: to investigate how each modality is influenced by the choice of optimization policy in multimodal neural networks.

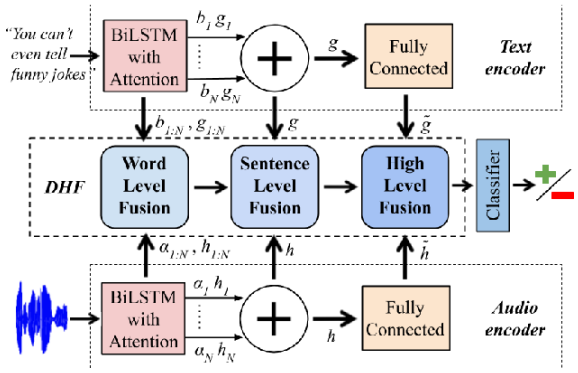


Figure 3.6. *Deep Hierarchical Fusion (DHF) [18] architecture for multimodal sentiment analysis. The model integrates textual and acoustic features using BiLSTMs with attention mechanisms and fuses them at word, sentence, and high levels before classification.*

3.3.4 Deep Fusion Architectures

Deep fusion architectures have emerged as a powerful paradigm for integrating multimodal information, particularly in vision-language tasks. Flamingo [106] employs gated cross-attention layers to integrate image and text features dynamically, leveraging frozen pretrained encoders to enable zero-shot and few-shot learning for applications like image captioning. ViLBERT [107] extends BERT [43] to a multimodal setting with a dual-stream transformer that processes visual and linguistic data separately, enhancing cross-modal interactions through co-attentional transformer layers. This approach excels in visual question answering (VQA) and referring expressions comprehension. UNITER [108] refines cross-modal fusion with joint pretraining on vision-language datasets, employing a single-stream transformer that enhances word-region alignment and masked language modeling, achieving state-of-the-art results in image-text retrieval and phrase grounding. The paper "Deep Hierarchical Fusion with Application in Sentiment Analysis" [18] introduces a deep hierarchical fusion (DHF) network for sentiment analysis, integrating textual and acoustic modalities. Using BiLSTM networks, DHF propagates both fused and unimodal representations across multiple levels—word, sentence, and sentiment—achieving state-of-the-art performance on the CMU-MOSI dataset [37].

3.4 Training Multimodal Neural Networks

The foundational principles of backpropagation, optimization, and generalization remain consistent with those in unimodal systems, but multimodal networks introduce unique challenges due to the heterogeneity and interdependence of modalities. In this section, we explore how these techniques are adapted to multimodal learning, while ensuring robust and effective learning across multiple data modalities.

Backpropagation remains the cornerstone of training multimodal neural networks, allowing the model to minimize loss by learning effectively modality-specific and shred representations. In scenarios where modality-specific sub-networks are utilized, gradients must flow not only through the shared fusion layers, but through each sub-network

independently as well. This ensures that modality-specific features are refined while the fusion layers capture cross-modality interactions. By updating the weights across all layers, backpropagation facilitates the joint optimization of modality-specific and shared components, enhancing the model's ability to extract complementary information and improve overall performance.

Complexities introduced by backpropagation, such as gradient conflicts, where gradients from different modalities point in opposing directions, preventing the network from converging effectively, and gradient misalignment, which occurs when temporal or structural inconsistencies between modalities lead to misaligned updates in shared layers, can impact optimization of multimodal neural networks. Also, given the diversity of modalities and the varying scales of modality-specific features, optimization of modality specific learning dynamics has proved challenging. Studying modality specific optimization strategies is the center of this thesis, as research indicates that balancing contributions across modalities during training is more beneficial than joint optimization [22] [23] [29].

Additionally, missing or noisy data, a common phenomenon in real-world multimodal systems, alongside with the natural heterogeneity across modalities can harm the generalization of the model. Generalization remains a critical aspect in multimodal machine learning, as it determines how well a model trained on a specific dataset can perform on unseen data. Besides traditional techniques to improve generalization such as data augmentation and regularization, multimodal learning introduces novel approaches like Meta-learning, where models are trained to adapt quickly to new data [109]. Multimodal transformers, also, have demonstrated improved generalization when trained on both language and vision modalities, allowing them to adapt to tasks with unpaired modalities or limited data by leveraging cross-modal knowledge [110].

3.5 Multimodal Applications in Sentiment Analysis and Emotion Recognition

Sentiment analysis and emotion recognition are key fields in machine learning for understanding human affect. While sentiment analysis captures overall positivity or negativity, emotion recognition identifies specific emotions. With multimodal approaches integrating text, audio, and visuals, these fields provide deeper insights into human affection. Given the vast number of behavioral signals involved in expressing emotions, recent research has shifted towards a multimodal approach to achieve more accurate emotion recognition. By integrating data from multiple channels—such as text, audio, and visual cues—models can capture a more nuanced understanding of emotional states. Similarly, in sentiment analysis, it is now widely recognized that emotions and sentiments are rarely communicated solely through text. Verbal and non-verbal cues, such as tone and facial expressions, play a critical role in communicating emotions. This realization has driven the evolution of multimodal sentiment analysis, where diverse data modalities are combined to provide a richer interpretation of sentiment [102]. This development aligns with real-world applications like video-based social media or interviews, where non-verbal cues

significantly contribute to understanding emotional intensity. This section explores the sentiment analysis and emotion recognition field, focusing on state-of-the-art multimodal applications, and presents key datasets such as CMU-MOSI and CMU-MOSEI that drive research in sentiment detection.

3.5.1 Sentiment Analysis

Sentiment analysis focuses on determining the emotional tone or polarity of information, historically applied to text data. The term has been associated with opinion mining and thus the two terms have been used interchangeably in many studies [111]. Early approaches relied on lexicon-based methods [111], using pre-defined dictionaries to sum the polarity of individual words, but faced limitations handling context sensitivity. Machine learning techniques like Naive Bayes and SVMs [112] introduced supervised learning on labeled data, yet they were limited by their reliance on handcrafted features. Advances like word embeddings (e.g., Word2Vec [113], GloVe [114]) captured semantic relationships and linguistic nuances more effectively, while deep learning models, including CNNs [115] and LSTMs [116], enabled the extraction of sequential dependencies to understand sentiments in sentences and paragraphs. The introduction of transformer-based models like BERT [43] was a revolution in text-based sentiment analysis by leveraging bidirectional context to detect cues like irony or sarcasm.

Sentiment analysis tasks can be framed as classification or regression. Sentiment classification includes categorization of data to discrete sentiment categories (e.g., positive, negative, neutral) [111], which is common in applications like social media monitoring. Regression models, in contrast, predict sentiment intensity on a continuous scale, which is valuable for tracking nuanced emotional trends, such as analyzing fluctuations in user sentiment over time on social media platforms.

3.5.2 Emotion Recognition

Emotion recognition in machine learning aims to identify human emotional states through text, speech or other behavioral signals. Unlike sentiment analysis, which primarily determines the positivity or negativity of a statement, emotion recognition seeks to classify segments into discrete emotion groups, such as happiness, sadness or fear, following Paul Ekman's Basic Emotions Model [117], enabling machine learning frameworks to map input data into discrete emotion categories, each reflecting different emotional qualities.

Emotion recognition often leverages the strengths of natural language processing (NLP), audio processing, and computer vision. For textual analysis, models such as recurrent neural networks (RNNs) or transformer-based models like BERT [43] are used to associate linguistic information with various emotions [118]. To extract information from speech tone or facial expressions emotion recognition models often use convolution neural networks (CNNs) for image-based analysis [119] or LSTM networks for sequential audio or video data [120].

Utterance: *"Become a drama critic!"*

Emotion: *Joy* **Sentiment:** *Positive*

Text	Audio	Visual
Ambiguous	Joyous tone	Smiling Face

Utterance: *"Great, now he is waving back"*

Emotion: *Disgust* **Sentiment:** *Negative*

Text	Audio	Visual
Positive/Joy	Flat tone	Frown

Figure 3.7. *Multimodal sentiment and emotion recognition example. The figure illustrates how text, audio, and visual cues contribute to emotion and sentiment classification. In the first case, the textual information is ambiguous, but the joyous tone and smiling face confirm a positive sentiment (Joy). In contrast, the second case shows a mismatch where the text suggests positivity, but the flat tone and frown lead to the correct classification of negative sentiment (Disgust). This highlights the importance of cross-modal integration for accurate sentiment analysis and emotion recognition. Source: [19].*

Sentiment analysis and emotion recognition, while distinct in their objectives, often utilize the same underlying frameworks and methodologies, as they both aim to interpret human affect through text, audio, and visual data. As discussed earlier, models like LSTMs, CNNs, and Transformers have been applied successfully to both tasks, with minimal adaptation. This overlap in techniques underscores the interconnected nature of these fields. Therefore, in the next section, we introduce state-of-the-art multimodal models without strict distinction between their use in sentiment analysis or emotion recognition, reflecting their shared applicability.

3.5.3 State-of-the-Art Multimodal Sentiment Analysis Models

The state-of-the-art multimodal models for sentiment and emotion analysis demonstrate the power of integrating multiple modalities such as text, audio, and video to capture the complexities of human communication. Attention mechanisms [16] play a pivotal role across many of these frameworks, dynamically prioritizing the most relevant cues from each modality to enhance interpretability and accuracy. For instance, the Multimodal Transformer [17] leverages attention mechanisms to align and fuse unaligned multimodal inputs, making it highly effective in capturing intricate dependencies between verbal and non-verbal cues. The Memory Fusion Network (MFN) [121] excels in temporal integration by dynamically tracking patterns across modalities, ensuring comprehensive analysis of sequential data like conversational interactions. Similarly, the Tensor Fusion Network (TFN) [122] utilizes tensor-based fusion to model both intra- and inter-modality interactions, enabling nuanced predictions in tasks requiring fine-grained understanding. Additionally, Self-MM [47] leverages self-supervised learning to enhance cross-modal representation learning, reducing reliance on labeled data and making it a robust frame-

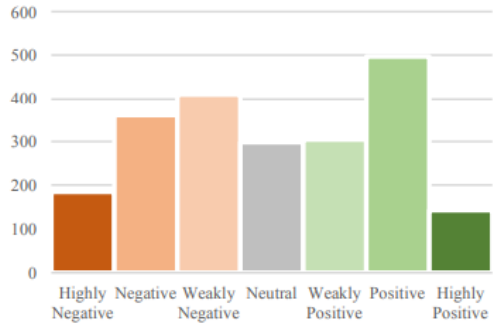
work for multimodal sentiment and emotion analysis. Together, these models underscore the advancements in multimodal learning, offering robust methodologies to analyze and interpret affective data.

3.5.4 Benchmark Datasets for Multimodal Sentiment Analysis

In the field of multimodal sentiment analysis and emotion detection, the CMU-MOSI and CMU-MOSEI datasets are among the most widely used resources, providing valuable benchmarks for evaluating models that detect sentiment across text, audio and video. Here, a detailed overview of each dataset used in our experiments is presented.

CMU-MOSI Dataset

The CMU Multimodal Opinion Sentiment and Intensity (MOSI) [37] dataset, introduced by Zadeh et al. (2016), is a foundational resource in multimodal sentiment analysis. The dataset was developed to support research on sentiment intensity and subjective opinion detection. It provides a collection of short video segments where individuals express a wide range of opinions on topics such as movies. Each segment in the dataset is annotated for sentiment intensity, making MOSI valuable for sentiment analysis research.



(a) *Distribution of Sentiment Categories in CMU-MOSI. The chart displays the counts of segments across sentiment categories, ranging from Highly Negative to Highly Positive [37].*

	Spoken words	Verbal-only prediction	Visual gestures	Visual-only prediction	Multimodal Dictionary prediction	Ground Truth annotation
1	And quite honestly I wish I've seen this over the summer.	0.14	Smile, Head nod	1.97	1.4	1.6
2	Now I'm not gonna lie there're a few parts that have great action sequences now even though it is an animated film it did have some great fight scenes.	2.3	Head shake	-0.77	1.44	1.4

(b) *Example of Multimodal Sentiment Prediction in CMU-MOSI. Verbal-only, visual-only, and multimodal model predictions are shown alongside ground truth annotations [37].*

Figure 3.8. *Visualizations from the CMU-MOSI dataset, highlighting the sentiment distribution and multimodal prediction capabilities.*

The CMU-MOSI dataset includes 3,702 video segments, categorized into 2,199 opinion segments and 1,503 objective segments. Each opinion segment is annotated on a sentiment intensity scale ranging from -3 to +3, where -3 indicates strong negativity, -2 negativity, -1 weak negativity, 0 represents neutrality, +1 weak positivity, +2 positivity, and +3 signifies strong positivity. The continuous nature of this scale makes the dataset ideal for sentiment regression tasks or categorical sentiment classification. The dataset includes three data modalities for each segment: textual data (transcriptions), audio features (e.g., pitch, energy, and MFCCs), and visual data (e.g., facial expressions, action units, and head orientation), providing valuable non-verbal cues for sentiment analysis.

CMU-MOSEI Dataset

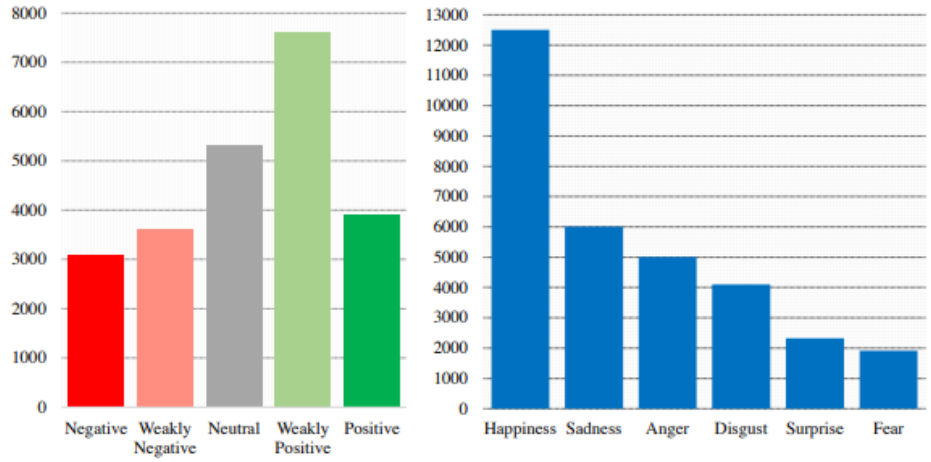


Figure 3.9. *Distribution of Sentiment and Emotions in CMU-MOSEI. The left chart shows sentiment categories, discretized from the original -3 to +3 scale. The right chart shows emotion categories (Happiness, Sadness, Anger, Disgust, Surprise, Fear), with sufficient data for each emotion [123].*

The CMU Multimodal Opinion Sentiment and Emotion Intensity (CMU-MOSEI) [123] dataset, introduced by Zadeh et al. (2018), extends CMU-MOSI by providing 23,453 video segments sourced from 1,000 speakers across 250 topics on YouTube videos. In addition to sentiment annotations (-3 to +3 scale), each segment is labeled for the presence of six emotions: happiness, sadness, anger, fear, disgust, and surprise. Each emotion is scored on a scale from 0 to 3, where 0 represents the absence of an emotion, 1 weak presence, 2 presence, and 3 strong presence. Each video segment includes three modalities, providing a robust foundation for sentiment and emotion analysis. Text transcriptions deliver verbal content. Acoustic features, such as Mel-frequency cepstral coefficients (MFCCs), pitch, and intensity, capture vocal tones reflecting emotional or sentimental intensity. Visual data, including facial expressions, head movements, and eye gaze, offers non-verbal cues critical for interpreting sentiment and emotion. By integrating sentiment and emotion labels, CMU-MOSEI is particularly useful for multitask learning models that perform both sentiment and emotion analysis simultaneously, improving model robustness and generalization. Its large scale, multimodal design, and dual labels for sentiment and emotion make it a valuable benchmark for evaluating multimodal models requiring sophisticated fusion of language, visual, and acoustic signals.

3.6 Summary

Chapter 3 explores the shift towards multimodal machine learning, focusing on how different types of data (modalities) can be combined for richer and more meaningful learning. It discusses the difficulties in handling data from various sources, and highlights the importance of building models that balance and connect these modalities effectively.

Key methods are introduced, such as fusion strategies to combine modalities, attention mechanisms to focus on important features, and LSTM networks for processing sequential data. These approaches enable models to handle complex tasks like sentiment and emotion recognition. Challenges in training, such as conflicting gradients and uneven contributions from modalities, are also discussed. The chapter concludes with a look at multimodal applications and benchmark datasets like CMU-MOSI and CMU-MOSEI. These datasets provide a solid foundation for research in combining language, audio, and visual data.

The challenges discussed in Chapter 3 often lead to the phenomenon of modality imbalance, where some modalities dominate or underperform due to differences in information quality, noise, or biases. This issue will be presented in detail in Chapter 4, as it is the main research topic of this thesis. Chapter 4 will analyze the causes and effects of modality imbalance, and in Chapter 5, strategies and methods proposed to address this critical issue will be presented, paving the way for more balanced and effective multimodal learning systems.

Chapter 4

Modality Imbalance in Multimodal Learning

In the previous chapter, we explored key concepts in multimodal learning, including multimodal architectures, training strategies and the unique challenges introduced in multimodal models. One of the most critical among these is modality imbalance, where certain modalities dominate the learning process, suppressing contributions from others. This imbalance challenges the theoretical potential of multimodal models, making it a central focus of this thesis, which investigates a range of optimization methods to address this issue effectively.

4.1 Introduction

The issue of modality imbalance was first systematically analyzed by Wang et al. [22] in their work *"What Makes Training Multi-modal Classification Networks Hard?"*. They identified two key factors responsible for the degraded performance of multimodal networks compared to their unimodal counterparts, a counterintuitive observation. Firstly, the increased capacity of multimodal networks, referring to their larger number of parameters and complex architectures required to process integrated information from multiple modalities, often leads to overfitting. Secondly, each modality overfits or generalizes at a different pace from other modalities due to differences in the complexity and amount of information they provide. For instance, one modality may quickly adapt to the training data, leading to overfitting, while another may generalize better but learn at a slower pace. These discrepancies make it challenging to train them jointly using a common optimization strategy, as the model might prioritize modalities that overfit faster, neglecting those that require more gradual learning to learn effectively. Their study showed that naive joint optimization frameworks result to dominance of stronger modalities and underutilization of weaker ones.

Building upon this foundation, Wu et al. [23] formalized the Greedy Learner Hypothesis, emphasizing that multimodal models naturally prioritize faster-learning modalities, neglecting those that learn at slower rates. To quantify this phenomenon, they introduced metrics like conditional utilization rate and conditional learning speed, which illustrate how models disproportionately rely on dominant modalities. Together, these works underline the need to address modality imbalance as a fundamental challenge in multimodal learning.

Huang et al. [29] introduced the concept of modality competition to explain why multimodal networks underperform compared to unimodal models, especially under late-fusion joint training. They demonstrated that during training, modalities compete for representation, with only a subset of modalities, typically the dominant ones, being effectively learned. This phenomenon arises due to differences in feature learning dynamics and the random initialization of network parameters, which disproportionately benefit certain modalities. Furthermore, weaker modalities, particularly those with insufficient data structures, are often neglected, resulting in degraded feature representations.

Collectively, these findings provide a comprehensive foundation for understanding the challenges of modality imbalance in multimodal learning. These works highlight how multimodal networks are often dominated by stronger modalities due to overfitting, imbalanced learning rates, and competitive optimization dynamics. These phenomena limit the effective utilization of complementary information across modalities and degrade overall model performance. Building on these observations, we now describe the theoretical background that explains the relationships guiding gradients during optimization and how these dynamics lead to modality imbalances.

4.2 Problem Definition

In this section we rprovide a mathematical framework to analyze how gradient interactions during optimization lead to modality imbalance. Let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ represent a multimodal dataset containing N samples. Each sample consists of input features $x_i = \{m_i^k\}_{k=1}^{M}$ derived from M distinct modalities and a one-hot encoded label $y_i = \{c_{i,j}\}_{j=1}^{Y}$, where $c_{i,j} = 1$ indicates that the label for the i-th sample belongs to category j, and Y denotes the total number of classes. Each modality is associated with a specific feature extractor, denoted as $F_k(\partial_k)$, where F_k is a neural network parameterized by ∂_k. For the i-th sample, the features extracted from the k-th modality are represented as $F_k(\partial_k; m_i^k) \in \mathbb{R}^{d_k}$, where d_k is the dimensionality of the extracted features for modality k. To perform classification, a predictor S is defined to map the extracted features into the label space. The predictor S operates on the aggregated multimodal features, and the objective of multimodal learning is to minimize the empirical loss function:

$$L(S(\{F_k(x)\}_{k=1}^{M}), y) = \frac{1}{N}\sum_{i=1}^{N} \ell(S(\{F_k(\partial_k; m_i^k)\}_{k=1}^{M}), y_i), \tag{4.1}$$

where ℓ is the task-specific loss function, such as cross-entropy loss. The predictor S can be decomposed into two components, a fusion function f, which combines modality-specific features into a joint representation and a classifier g, which maps the fused representation to the output label space. This decomposition allows S to be written as $S = g \circ f$. For example, using concatenation as the fusion strategy and a linear model for the classifier, the predictor can be expressed as:

$$S(\{F_k(\partial_k; m_i^k)\}_{k=1}^{M}) = W \cdot [F_1(\partial_1; m_i^1) : \cdots : F_M(\partial_M; m_i^M)] \tag{4.2}$$

or

$$S(\{F_k(\partial_k; m_i^k)\}_{k=1}^M) = \sum_{k=1}^{M} W_k \cdot F_k(\partial_k; m_i^k), \tag{4.3}$$

where $[\cdot : \cdot]$ denotes concatenation, $W \in \mathbb{R}^{Y \times \sum_k d_k}$ is the weight matrix of the classifier, and $W_k \in \mathbb{R}^{Y \times d_k}$ represents the portion of W corresponding to modality k. In a fusion concatenation model, the fused representation $\Phi^M(x_i)$ for the i-th sample is constructed by concatenating features extracted from M modalities:

$$\Phi^M(x_i) = [F_1(\partial_1; m_i^1) : F_2(\partial_2; m_i^2) : \cdots : F_M(\partial_M; m_i^M)], \tag{4.4}$$

where $[\cdot : \cdot]$ denotes the concatenation operator, and $F_k(\partial_k; m_i^k)$ represents the feature representation extracted from modality k by the feature extractor parameterized by ∂_k. The gradient-based parameter update for the feature extractor parameters ∂_k for modality k is:

$$\partial_k^{t+1} = \partial_k^t - \eta \cdot \nabla_{\partial_k^t} L(\Phi^M(x), y), \tag{4.5}$$

where η is the learning rate. Substituting the loss gradient, we have:

$$\nabla_{\partial_k^t} L(\Phi^M(x), y) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\partial \ell(\Phi^M(x_i), y_i)}{\partial \Phi^M(x_i)} \cdot \frac{\partial \Phi^M(x_i)}{\partial F_k(\partial_k)} \cdot \frac{\partial F_k(\partial_k)}{\partial \partial_k^t} \right). \tag{4.6}$$

Since concatenation does not inherently mix features from different modalities, the term $\frac{\partial \Phi^M(x_i)}{\partial F_k(\partial_k)}$ simplifies to an identity matrix corresponding to the modality-specific block. However, the shared gradient $\frac{\partial \ell(\Phi^M(x_i), y_i)}{\partial \Phi^M(x_i)}$ propagates globally across all concatenated features. This shared gradient often aligns disproportionately with the dominant modality, suppressing contributions from weaker modalities. In a fusion concatenation model, the optimization process favors modalities whose features align strongly with the shared gradient. For modality k, its gradient aligns consistently with the shared gradient if:

$$\frac{\partial \ell(\Phi^M(x_i), y_i)}{\partial \Phi^M(x_i)} \cdot F_k(\partial_k) \gg \frac{\partial \ell(\Phi^M(x_i), y_i)}{\partial \Phi^M(x_i)} \cdot F_j(\partial_j), \quad \forall j \neq k. \tag{4.7}$$

When this condition is satisfied for a dominant modality k, its features receive larger updates during backpropagation, while weaker modalities are left under-optimized. For weaker modalities, the contribution to the shared gradient is small, leading to diminishing updates for their parameters. This causes weaker modalities to stuck in suboptimal regions of the parameter space, preventing them from contributing effectively to the fused representation. Since concatenation aggregates modality-specific features without balancing their contributions, the fused representation $\Phi^M(x)$ becomes biased toward dominant modalities. This reduces the diversity and robustness of the multimodal model, particularly when weaker modalities carry critical but less pronounced information.

Furthermore, the direction of gradients also contributes to modality imbalance [20] [22] [23] [124]. During backpropagation, the gradients from different modalities may not align well in the high-dimensional parameter space. If the gradients of a weaker modality point in directions that are orthogonal or even conflicting with the gradients of a dominant

modality, the updates for the weaker modality's parameters become inefficient or counter-productive. This conflict in gradient directions escalates the under-optimization of weaker modalities, further biasing the fused representation $\Phi^M(x)$ toward dominant modalities. Figure 4.1 demonstrates the interference caused by the dominant modality in multimodal

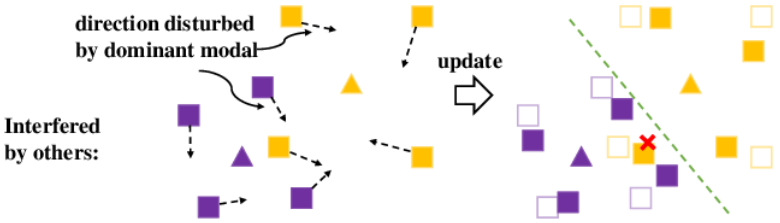


Figure 4.1. *Visualization of gradient direction distortion in multimodal learning taken from [20]. The weaker modality (purple) is influenced by the dominant modality (yellow), causing its gradient updates to deviate from the optimal learning path.*

learning. The movement of weaker modality representations (purple) is disturbed by the stronger modality (yellow), resulting in misaligned updates. The dashed arrows indicate how the update direction is influenced, causing the weaker modality to shift in directions that do not align with its optimal feature space. Additionally, the clustering structure is affected, as weaker modalities struggle to establish distinct boundaries due to interference from stronger modalities. The red cross marks a potential incorrect classification caused by this imbalance. This interference leads to ineffective feature learning, misalignment in representation space, and potential misclassifications. The visualization highlights the challenge of modality imbalance, where dominant modalities dictate the overall learning trajectory, preventing weaker modalities from contributing effectively to the final decision boundary.

4.3 Impact of Modality Imbalance

Unbalanced multimodal learning has significant implications for both model performance and practical applications. The suboptimal utilization of multimodal features, where weaker modalities often fail to contribute effectively to the learning process, can be considered particularly problematic in tasks where these modalities carry important complementary information [23]. For instance, in sentiment analysis, text data often dominates the learning process, leaving cues from audio and visual modalities underutilized, despite being crucial for a comprehensive understanding of emotions. Empirical findings [125] indicate that these networks often develop unimodal bias. This phenomenon is particularly pronounced in late fusion architectures, where modality integration occurs at deeper layers, leading to extended unimodal phases and suboptimal learning outcomes. A critical consequence of this bias is its impact on fused representations [29]. Dominance of certain modalities during optimization limits the model's ability to capture meaningful cross-modal interactions. This bias reduces the richness of representations and the model's ability to leverage the full potential of multimodal data. Additionally, modality imbalance leads to reduced generalization and compromised robustness, as models tend

to overfit to the dominant modality, failing to adapt to diverse real-world scenarios. This over-reliance on a single modality makes multimodal systems vulnerable to noisy, missing, or unreliable data of the dominant modality. When dominant modalities become less reliable,
models trained with imbalanced modality contributions struggle to adapt, resulting in poor generalization and reduced robustness across varying conditions [23] [126]. Finally, modality imbalance complicates training dynamics, introducing inefficiencies and conflicts in the optimization process [22] [124]. Imbalances slow down learning and make convergence harder to achieve, increasing the computational cost and difficulty of training multimodal networks effectively. Figure 4.2 illustrates the influence of dominant modality in multimodal learning of CMU-MOSEI dataset. The text modality exhibits the lowest loss and drives the optimization process, while audio and visual modalities remain largely stagnant. The multimodal model closely follows the text-only curve, confirming that the dominant modality dictates the overall learning dynamics. This highlights the challenge of effectively incorporating weaker modalities, as they fail to contribute meaningfully to the optimization process.

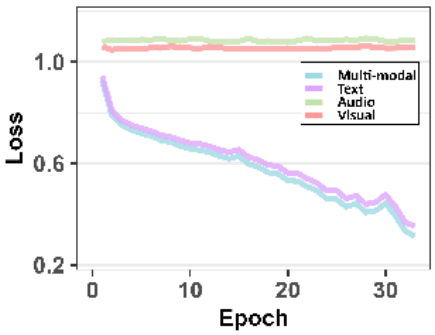


Figure 4.2. *Loss curves for audio, vision and text modalities of CMU-MOSEI dataset in late concatenation fusion model. Source: [21].*

4.4 Summary

In this chapter, we explored the phenomenon of modality imbalance in multimodal learning, leading to suboptimal utilization of weaker modalities. Through a mathematical framework, we analyzed how gradient interactions during optimization contribute to this imbalance, highlighting challenges such as gradient alignment issues, dominance of stronger modalities, and under-optimization of weaker ones. These dynamics result in biased fused representations, reduced generalization, and compromised robustness, degrading the performance of multimodal networks. In the next chapter, we will provide an overview of several approaches proposed in the literature to address modality imbalance. Building on these approaches, our experiments will investigate how modality imbalance is mitigated in different optimization scenarios, offering insights into the effectiveness of these strategies.

Chapter 5

Addressing Modality Imbalance Through Optimization

Chapter 5 provides a detailed review of optimization methods proposed in the literature to address modality imbalance in multimodal learning. To organize the discussion, we first present a brief overview of several approaches introduced to mitigate the problem. Following this, we focus on two specific categories of methods that are central to this research: dynamic gradient adjustment methods and multi-loss rebalancing approaches. The selected methods in these categories—On-the-fly Gradient Modulation with Generalization Enhancement (OGM-GE) [26], Adaptive Gradient Modulation (AGM) [27], Prototypical Modal Rebalance (PMR) [20], and ReconBoost [21]—are examined in detail, as they form the foundation for the experiments conducted in later chapters.

5.1 Overview of Optimization Methods

To ensure proportional contributions from all modalities, Wang et al.[22] introduced gradient blending through the Overfitting-to-Generalization Ratio (OGR), which evaluates the performance of each modality during training. This approach dynamically adjusts gradient contributions to prevent overfitting modalities from dominating while amplifying the contributions of weaker ones.

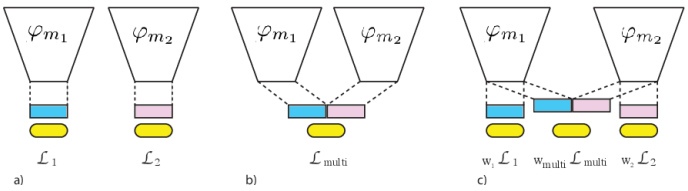


Figure 5.1. *Illustration of different training strategies for multimodal learning. (a) Independent unimodal training, where each modality is optimized separately. (b) Joint multimodal training, where a shared representation is jointly optimized using a single loss function. (c) Joint training of two modalities with Gradient Blending [22].*

Wu et al. [23] proposed two metrics: the Conditional Utilization Rate (CUR), which quantifies how much each modality contributes to the overall learning process, and the Conditional Learning Speed (CLS), which evaluates the rate at which a modality learns relative to others. These metrics enable targeted gradient adjustments, slowing down

dominant modalities and prioritizing underutilized ones. Gradient harmonization during pre-training through cross-modality gradient realignment and gradient-based curriculum learning has been proposed [30] to handle strong gradient conflicts in trimodal sample interactions. Classifier-guided Gradient Modulation [31] technique addresses the challenge of modality dominance in multimodal learning by modulating both the magnitude and direction of gradients during training. MMPareto [32] tackles gradient conflicts between multimodal and unimodal learning objectives by employing Pareto optimization [127]. It integrates gradients in a conflict-free manner, ensuring a common direction while enhancing gradient magnitude to improve generalization.

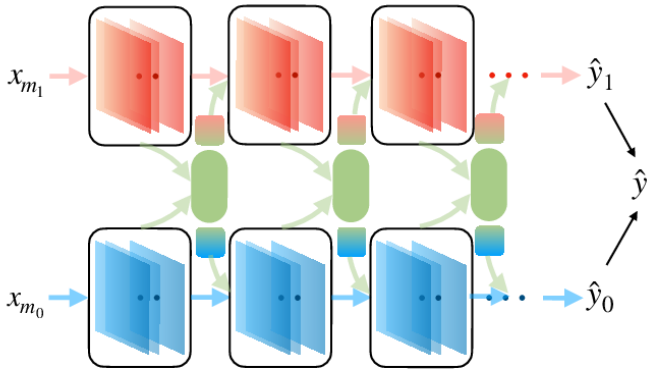


Figure 5.2. *Illustration ofmulti-modal DNN with intermediate fusion presented in [23]. Different modality streams (x_{m_0} and x_{m_1}) undergo multiple layers of transformation and interaction, leading to joint predictions ($\hat{y}_0$, $\hat{y}_1$) and an overall fused output ($\hat{y}$). The green connectors highlight the fusion pathways that facilitate cross-modal learning.*

Beyond direct gradient adjustments, MMCosine [126] introduces a multi-modal cosine loss function that performs modality-wise L2 normalization of features and weights, enhancing the discriminability and balance of multi-modal fine-grained learning. Focusing more on the loss landscape "Sharpeness-Aware Minimization" or SAM [12] algorithm finds the model parameters, which define a region combining flatness and low loss. Sharpness of the training loss landscape is measured as the maximum difference between two model losses with parameters differing only by a very small value. The maximum loss of model with parameters' values close to the original one is referred as the SAM loss and its minimum, increased by a standard L2 regularization term, identifies the "Sharpeness-Aware Minimization" problem. Applying Stochastic-Gradient Descent to the SAM loss, authors propose an efficient model-independent algorithm. Lookahead Optimizer [33] introduces a "k steps forward, 1 step back" approach that balances fast, exploratory updates with stable, recalibrated adjustments, designed to improve both convergence speed and model stability. This optimizer operates with two sets of weights: fast weights, updated k times using a base optimizer, such as SGD or Adam, for quick adjustments, and slow weights, which periodically "look back" and incorporate the fast weights' direction to shift. Although the SAM algorithm and Lookahead Optimizer were originally developed for unimodal learning scenarios, their underlying principles—such as improving loss landscape flatness and ensuring stable convergence—offer insights that could inform optimization strategies for multimodal learning.

Other studies focus on excluding modalities irrelevant for a specified task. Panda et al. propose AdaMML [24], a lightweight policy network that selects the most informative modalities dynamically, adjusting which of them are going to be used at different stages of a video sequence. Irrelevant Modality Dropout (IMD) mechanism [34] uses a relevance-checking model to filter out non-contributory audio cues in video classification tasks. Liu et al introduce an Attention-based Multi-modal Fusion Framework [35] with two modules: importance-based attention and complementary attention to emphasize critical modalities and capture inter-modal dependencies. Another approach is explored by He et al. [36], formulating modality selection as an optimization problem. Using submodular optimization techniques, this method ensures near-optimal selection of a subset of modalities that maximizes learning efficiency while maintaining computational feasibility. This method contrasts with fusion-based strategies that integrate all available modalities, as it focuses on selecting the most impactful subset rather than adjusting the contribution of all modalities. By effectively selecting a diverse and informative set of modalities, this approach indirectly mitigates modality imbalance, preventing the model from over-relying on a dominant modality.

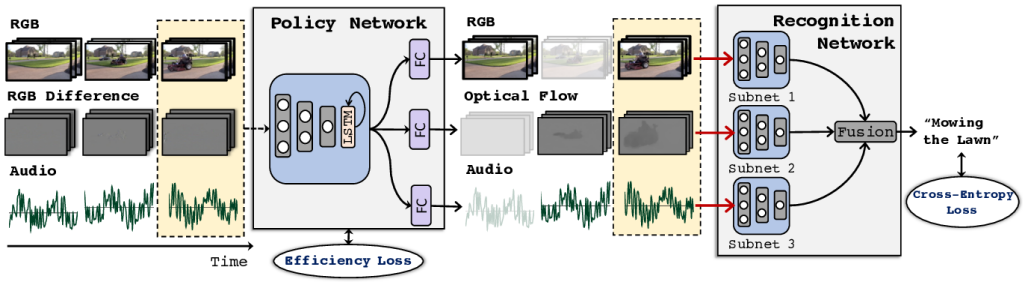


Figure 5.3. *Illustration of AdaMML [24] framework. A Policy Network dynamically selects relevant modalities, guided by an efficiency loss to optimize computational cost. Selected features are passed to a Recognition Network, where modality-specific subnets process different streams before undergoing fusion.*

Sequential learning frameworks restructure the optimization process to avoid cross-modal interference and retain previously acquired knowledge. Memory Consolidation Mechanisms (MLA) [25] transforms multimodal learning into an alternating unimodal optimization process, reducing modality interference. At the same time, it preserves cross-modal interactions through a shared network head, which undergoes continuous updates across different modalities. To maintain previously learned knowledge, a gradient adjustment mechanism regulates this optimization process. During inference, MLA employs a test-time uncertainty-based fusion strategy to seamlessly integrate multimodal information.

Influenced by the significant role of gradient modulation in the literature, we focus on OGM-GE [26] and AGM [27], which leverage dynamic gradient adjustment to address modality imbalance. Additionally, motivated by optimization techniques centered on loss function adjustments, we explore PMR [20] for modality rebalance and ReconBoost [21] combining a unimodal alternating paradigm with targeted loss rebalancing strategies. While the methods discussed above provide a general understanding of strategies to mit-

igate modality imbalance, this research specifically focuses on these four key methods, which are discussed in detail in subsequent sections as representative approaches for investigating the unbalanced multimodal learning under the scope of sentiment classification.

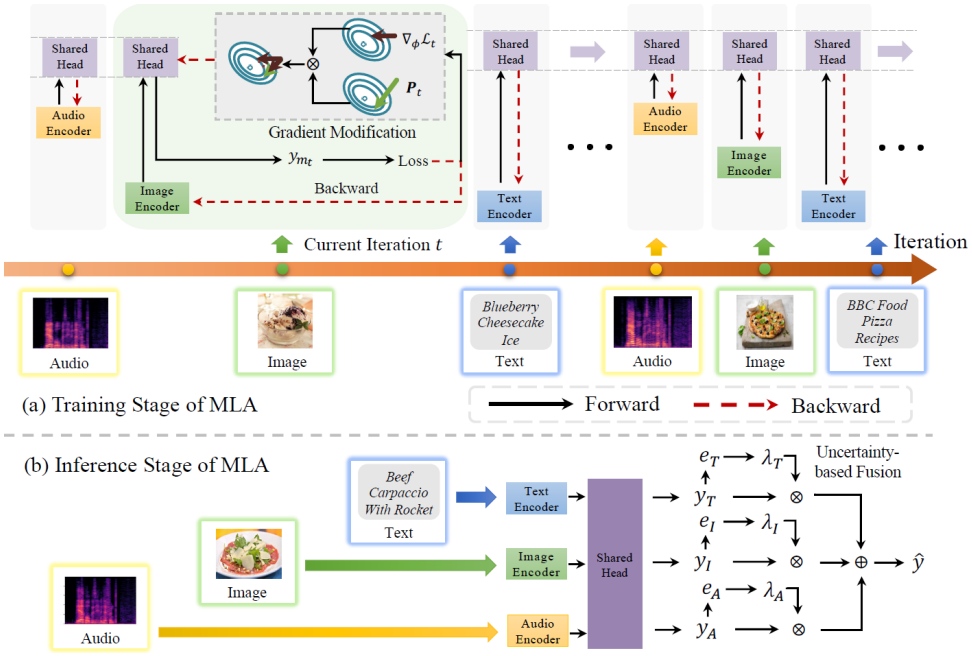


Figure 5.4. *Illustration of the Modality Learning Alternation (MLA) framework [25]. (a) Training Stage: The model alternates between unimodal learning for audio, image, and text encoders while utilizing a shared head for cross-modal representation. Gradient modification ensures that each modality learns effectively without interference. (b) Inference Stage: The learned unimodal encoders pass their representations to an uncertainty-based fusion mechanism, which dynamically assigns weights ($\hat{\jmath}$) to each modality before generating the final prediction.*

5.2 Dynamic Gradient Adjustment Methods

Dynamic gradient adjustment methods go beyond traditional optimization algorithms, such as Adam or SGD, by introducing mechanisms that actively modulate gradient contributions from different modalities based on specific conditions unique to each method. This section highlights two dynamic gradient adjustment methods central to this research: On-the-fly Gradient Modulation with Generalization Enhancement (OGM-GE) [26] and Adaptive Gradient Modulation (AGM) [27]. OGM-GE leverages gradient modulation and noise-based gradient adjustment to balance modality contributions dynamically and enhance the overall robustness of multimodal models. AGM introduces competition-free states and real-time gradient adjustments to further mitigate modality competition. These methods represent significant advancements in the optimization of unbalanced multimodal learning, particularly within the context of sentiment classification, and are detailed in the following subsections.

5.2.1 On-the-fly Gradient Modulation with Generalization Enhancement

The On-the-fly Gradient Modulation with Generalization Enhancement (OGM-GE) [26] method combines two key mechanisms—gradient modulation and generalization enhancement—to address modality imbalance in multimodal learning. Gradient modulation aims to balance contributions from each modality during training by dynamically adjusting gradient magnitudes, while generalization enhancement reduces overfitting through noise injection, improving the robustness of the model. Originally designed for bimodal audio-visual tasks, OGM-GE adapts these mechanisms to harmonize the optimization process and mitigate the dominance of stronger modalities.

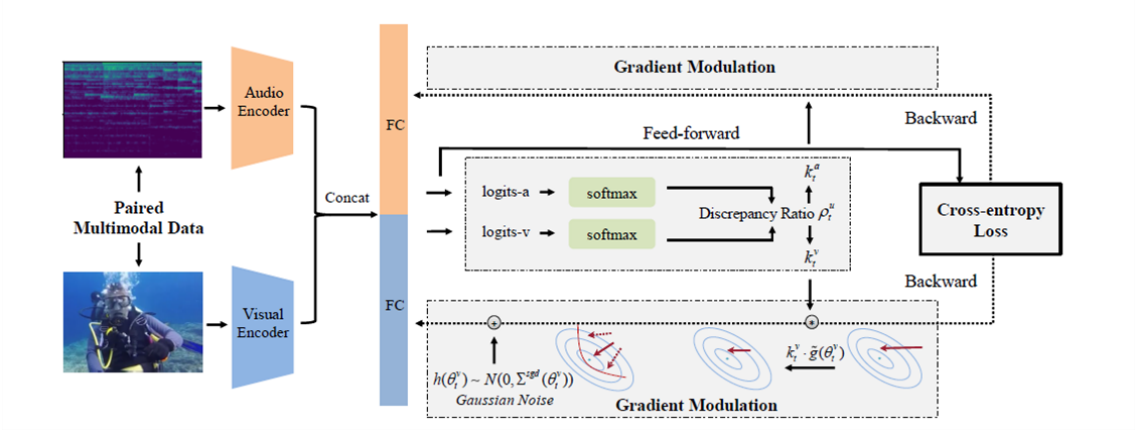


Figure 5.5. *Illustration of the On-the-fly Gradient Modulation with Generalization Enhancement Framework as presented by Peng et al. [26].*

Gradient Modulation: The method introduces discrepancy ratios to balance learning contributions across modalities, ensuring that no single modality dominates during training. Here, the modality $m \in \{a, v\}$ can represent either audio (a) or video (v) in a bimodal setup. The discrepancy ratio for modality m at the t-th step is calculated as:

$$\rho_t^m = \frac{\sum_{i \in B_t} s_i^m}{\sum_{i \in B_t} s_i^n}, \tag{5.1}$$

where n represents the other modality ($n \neq m$), and B_t denotes a randomly chosen mini-batch of size m at the t-th step. The term s_i^m represents the contribution of modality m for sample i and is computed as:

$$s_i^m = \sum_{k=1}^{M} 1_{k=y_i} \cdot \text{softmax}\left(W_t^m \cdot \varphi_t^m(\partial^m, x_i^m) + \frac{b}{2}\right), \tag{5.2}$$

where M is the total number of output classes, $1_{k=y_i}$ is an indicator function that equals 1 if the predicted class k matches the true label y_i, and 0 otherwise, W_t^m represents the weights of the linear classifier for modality m at the t-th step, $\varphi_t^m(\partial^m, x_i^m)$ is the feature representation of input x_i^m from modality m, produced by the encoder parameterized by ∂^m and b is the bias term added to the logits before applying the softmax function. The term $W_t^m \cdot \varphi_t^m(\partial^m, x_i^m) + \frac{b}{2}$ estimates the predicted logits for modality m, which are normalized

into probabilities using the softmax function. The term $\frac{b}{2}$ serves as a bias correction mechanism inspired by the Deep Boltzmann Machine framework. By incorporating $\frac{b}{2}$, the logits for each modality are adjusted to take the average of the bottom-up and top-down weights. This ensures that the uni-modal prediction more accurately reflects the individual modality's contribution to the multimodal model, without favoring any specific modality. Using the discrepancy ratios ρ_t^m as the guiding conditions for adjustment, the gradients for modality m are modulated dynamically during training. The modulation coefficient for modality m is defined as:

$$k_t^m = 1 - \tanh\left(a \cdot \mathrm{ReLU}(\rho_t^m)\right), \tag{5.3}$$

where a is a hyperparameter controlling the degree of modulation, ReLU is the Rectified Linear Unit activation function ensuring non-negative values for ρ_t^m and tanh is the hyperbolic tangent function providing smooth scaling for the modulation coefficients. The coefficient k_t^m dynamically scales the weights, reducing the contributions of stronger modalities, having higher discrepancy ratios, and amplifying weaker ones (with lower discrepancy ratios). This ensures balanced optimization across modalities, enabling the model to integrate information effectively from all available modalities.

An alteration of the Gradient Modulation described in [20] is the acceleration of the weaker modality m by multiplying its gradients with:

$$k_t^m = 1 + \tanh\left(a \cdot \mathrm{ReLU}(\rho_t^m)\right), \tag{5.4}$$

where a is a hyperparameter controlling the degree of modulation, ReLU is the Rectified Linear Unit activation function ensuring non-negative values for ρ_t^m and tanh is the hyperbolic tangent function providing smooth scaling for the modulation coefficients. This method uses the discrepancy ratio for conditional enhancement of the weaker modality each time and will be examined as an alteration of the OGM method, referred to as ACC in Chapter 6.

Generalization Enhancement: To balance the reduction in stochastic gradient noise intensity caused by the gradient modulation coefficients, OGM-GE incorporates a generalization enhancement mechanism. This is achieved by adding dynamically sampled Gaussian noise to the gradient updates. The update rule is now expressed as:

$$\partial_{t+1}^m = \partial_t^m - \eta\left(k_t^m \tilde{g}(\partial_t^m) + h(\partial_t^m)\right) \tag{5.5}$$

where k_t^m represents the modulation coefficient for modality m at step t, $\tilde{g}(\partial_t^m)$ denotes the modulated gradients for modality m, and $h(\partial_t^m) \sim \mathcal{N}(0, \Sigma^{sgd}(\partial_t^m))$ is the dynamically sampled Gaussian noise. Here, $\Sigma^{sgd}(\partial_t^m)$ represents the covariance matrix of the gradient noise. The noise term $h(\partial_t^m)$ is designed to restore and even enhance the stochastic gradient noise's intensity, which might diminish as a result of the modulation process. By introducing this noise, the generalization capacity of the model is preserved.

5.2.2 Adaptive Gradient Modulation

The Adaptive Gradient Modulation (AGM) [27] method addresses the challenge of modality competition in multimodal learning, where dominant modalities overshadow weaker ones, limiting the effective use of multimodal information. AGM dynamically modulates gradient signals to ensure balanced learning across modalities, regardless of the fusion strategy employed. AGM employs a Shapley value-inspired approach to compute mono-modal outputs, disentangling individual modality contributions even in complex fusion scenarios. Ratios derived from mono-modal outputs serve as conditional factors

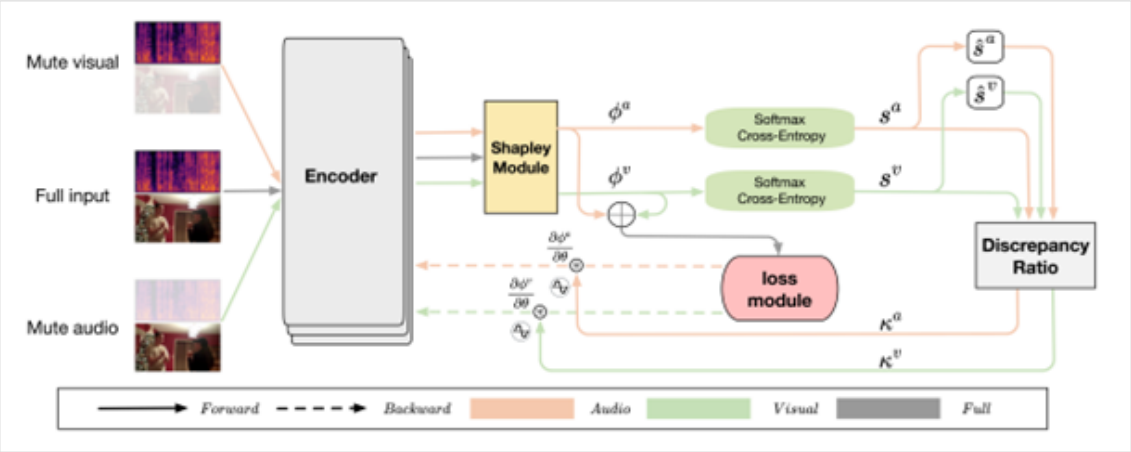


Figure 5.6. *Illustration of the Adaptive Gradient Modulation Method, as presented by Li et al. [27].*

for dynamic gradient adjustments via gradient modulation coefficients. A novel metric for modality competition further quantifies the interference between modalities, providing insights into AGM's efficiency.

Mono-Modal Contribution Analysis: Let $\phi(x)$, where $x = (x_{m_1}, \ldots, x_{m_k})$, represent a multi-modal model with k modalities, and $M := \{m_i\}_{i\in[k]}$ be the set of all modalities. We define 0_m as the absence of features from modality m. For a subset $S \subseteq M$, $\phi(S)$ denotes the output when all modalities in S are present, and those not in S are replaced by 0_m. The mono-modal response $\phi_m(x)$ for a modality m is given by:

$$\phi_m(x) = \sum_{S\subseteq M/\{m\};S\neq\emptyset} \frac{|S|!(k-|S|-1)!}{k!} V_m(S;\phi) \tag{5.6}$$

where $V_m(S;\phi) = \phi(S\cup\{m\}) - \phi(S)$. This ensures that:

$$\phi(x) = \sum_m \phi_m(x) \tag{5.7}$$

For the case of two modalities, this simplifies to:

$$\phi_{m_1}(x) = \frac{1}{2}[\phi(\{m_1, m_2\}) - \phi(\{0_{m_1}, m_2\}) + \phi(\{m_1, 0_{m_2}\})] \tag{5.8}$$

Dynamic Modulation of Gradients: Gradients are modulated based on modality ratio r_t^m derived from mono-modal information s_t^m. Discrepancy ratios guide the adjustment of

modulation coefficients κ_t^m for each modality during backpropagation:

$$r_t^m = \exp\left(\frac{1}{K-1}\sum_{m'\in[K];m'\neq m}(s_t^m - s_t^{m'})\right) \tag{5.9}$$

$$\kappa_t^m = \exp\left(-a\cdot(r_t^m - \tau_t^m)\right), \tag{5.10}$$

where a is a modulation hyper-parameter, and τ_t^m is the discrepancy ratio measured by the averaged differences of running average of the ratio for modality m relative to the other modalities.

Competition Strength Metric: By incorporating the concept of a mono-modal state the authors aim to reflect how a modality would behave without competition from other modalities. They, also, quantify this behavior through the competition strength metric. The concept of competition-less states isolates a modality's independent behavior by removing the influence of competing modalities. This insight allows AGM to measure the degree of interference in multimodal setups, guiding gradient modulation to mitigate competition effectively. For a modality m_1, its competition-less state is defined by a function $C_{m_1}(x_{m_1}; E_{m_1}/m_2)$, where E_{m_1}/m_2 denotes the environment of m_1 without m_2. For example, in the late fusion case, the environment without m_2 can be represented as $(0_{m_2}, \phi_{m_1}, T_{m_1}, D_{m_1})$. The competition strength d_m is then defined as:

$$d_m = \frac{\sum_i (C_m(x_i^m) - f_m(z_i))^2}{\sum_i (C_m(x_i^m) - C_m)^2} \tag{5.11}$$

where $f_m(z)$ is a linear predictor trained on the latent features z from the multi-modal model, and C_m is the average mono-modal concept value. The modulation coefficients and the competition strength metric serve as interpretable signals that provide insight into the training dynamics. These metrics reveal how dominant modalities affect weaker ones and guide adjustments to achieve balanced learning.

Both On-the-fly Gradient Modulation with Generalization Enhancement and Adaptive Gradient Modulation represent state-of-the-art dynamic gradient adjustment methods that address the core challenges of modality imbalance by leveraging mechanisms like gradient modulation, generalization enhancement, and competition-free states. These methods provide flexible, interpretable, and effective strategies for achieving balanced optimization across modalities, making them central to this study's focus.

5.3 Loss-Based Rebalancing Approaches

Loss-based rebalancing methods tackle modality imbalance by dynamically adjusting the optimization process to amplify the contributions of weaker modalities. Prototypical Modal Rebalance (PMR) [20] and ReconBoost [21] specifically modify loss functions to handle the dominance of stronger modalities, ensuring a more balanced learning process. PMR uses class prototypes and entropy regularization to guide weaker modalities toward improved generalization, while ReconBoost employs a modality-alternating framework and reconcilement regularization to dynamically adjust learning objectives.

5.3.1 Prototypical Modal Rebalance

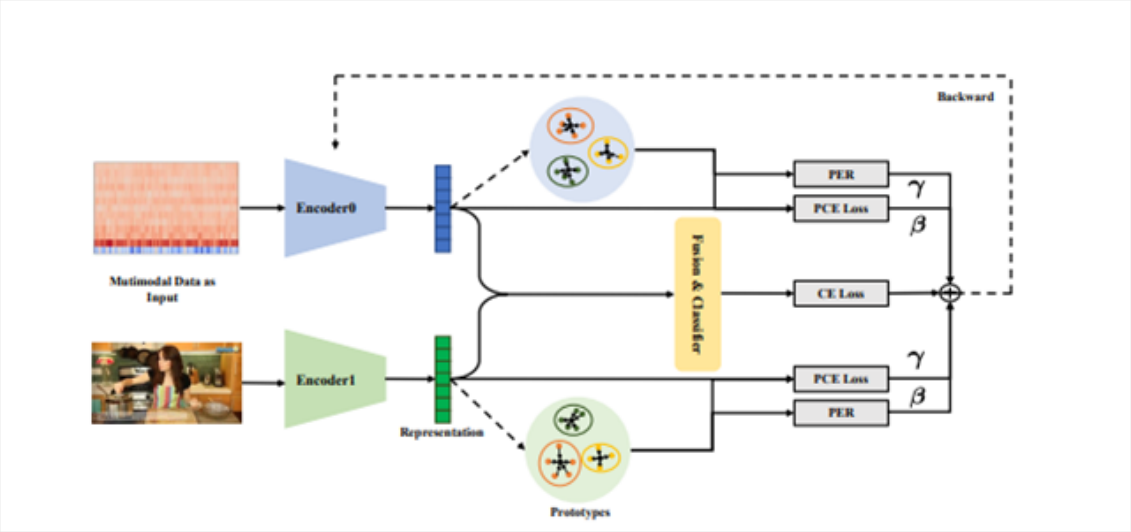


Figure 5.7. *Illustration of the Prototypical Modal Rebalance Method, as presented in Fan et al. [20].*

Prototypical Cross Entropy Loss: The authors [20] propose the prototypical modal rebalance strategy for bimodal audio-vision setups by calculating for each category of data the prototype:

$$c_k^m = \frac{1}{N_k} \sum_{i=1}^{N_k} z_{k_i}^m, \tag{5.12}$$

where z_i is the representation outputs of each encoder, and $m \in \{a, v\}$ denotes the modalities (e.g., audio and vision). After initializing the centroids, the authors calculate the Euclidean distance between each category prototype and the corresponding unimodal feature. Then they use the prototypes to produce a distribution over classes for the input data x, based on a softmax over distances to the prototypes in the embedding space for each modality. Subsequently, they find the imbalance ratio:

$$\rho_t^m = \frac{\sum_{i \in B_t^m} p_i^m}{\sum_{i \in B_t^n} p_i^n} \tag{5.13}$$

for batch data at training step t, where p represents the softmax over the Euclidean distances to the prototypes. Based on this ratio, the authors define the acceleration loss by promoting the slower-learning modality. They combine the cross-entropy loss (L_{CE}) of the multimodal representation with weighted losses of each unimodal branch. If the ratio indicates, for example, that audio ($m = a$) is learning faster than vision ($m = v$), they set $\beta = 0$, $\gamma = 1$, and control the degree of modulation through the hyperparameter a:

$$L_{\text{acc}} = L_{\text{CE}} + a \cdot \beta L_{\text{PCE}}^a + a \cdot \gamma L_{\text{PCE}}^v \tag{5.14}$$

where L^m_{PCE} for modality m is expressed:

$$L^m_{\text{PCE}}(f) = \mathbb{E}_{p(x^m,y)}\left[-\log \frac{\exp(-d(z^m, c^m_y))}{\sum_k \exp(-d(z^m, c^m_k))}\right]. \tag{5.15}$$

This term measures the prototypical classification loss for modality m, using the distance between unimodal features and their category prototypes.

Prototypical Entropy Regularization: To further mitigate dominance from faster-learning modalities, the method introduces Prototypical Entropy Regularization (PER) terms, which reduce the entropy of the faster-learning modality's class distributions:

$$L_{\text{final}} = L_{\text{acc}} - \mu \cdot \gamma H\left(p\left(-d\left(z^a, c^a_y\right)\right)\right) - \mu \cdot \beta H\left(p\left(-d\left(z^v, c^v_y\right)\right)\right) \tag{5.16}$$

where p denotes the softmax function, and d is the Euclidean distance between the unimodal representation and its category centroid. Between each training epoch, the prototypes are updated for a subset of unimodal data:

$$c^m_{k,\text{old}} = \epsilon c^m_{k,\text{old}} + (1-\epsilon)c^m_{k,\text{new}} \tag{5.17}$$

where ϵ is a momentum term controlling the update rate. Equations 5.14 and 5.16 can be applied selectively, depending on the objective—whether to solely accelerate one modality or to simultaneously accelerate one while penalizing the other.

5.3.2 ReconBoost

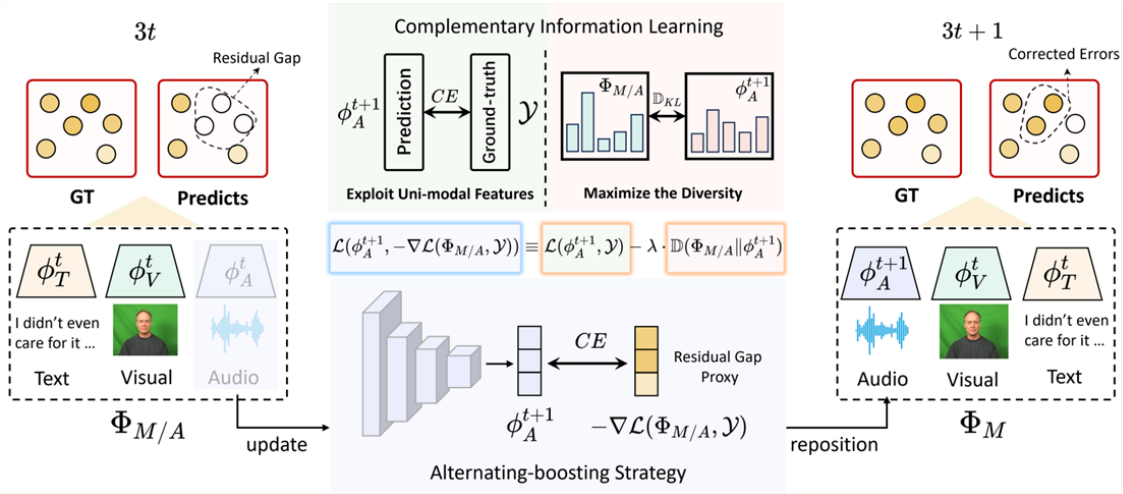


Figure 5.8. *Illustration of the ReconBoost Method, as presented in Hua et al. [21].*

ReconBoost [21] alternates the learning process across different modalities and incorporates KL-divergence-based regularization [128] to dynamically adjust the learning objective and prevent competition among modalities. By preserving only the latest model for each modality, ReconBoost prevents overfitting caused by ensembling strong learners. Additionally, the regularization term is added to maintain diversity between current and historical models, ensuring that the updated modality focuses on errors made by others,

thereby improving overall performance.

Alternating Modality Updates: Alternating updates form the core of ReconBoost's strategy to address modality competition, where stronger modalities often overshadow weaker ones. By updating one modality learner at a time while keeping others fixed, the method ensures that each modality receives focused optimization attention. This mechanism mitigates the domination of any single modality. The multi-modal learning objective is defined as:

$$L = \frac{1}{N}\sum_{i=1}^{N} \ell\left(\sum_{m=1}^{M} \phi_m(\vartheta_m; x_m^i), y_i\right), \tag{5.18}$$

where $\phi_m(\vartheta_m) = W_m \cdot F_m(\vartheta_m)$ represents the m-th modality learner, ϑ_m are the parameters of the m-th modality learner, x_m^i is the input for the m-th modality of the i-th example, and ℓ is the loss function, typically cross-entropy loss. During alternating updates, the parameters of one modality m are updated while others remain fixed:

$$\vartheta_m^{t+1} = \vartheta_m^t - \eta \cdot \nabla_{\vartheta_m^t} L_m, \tag{5.19}$$

where η is the learning rate, and L_m is the modality-specific loss. After the alternating training procedure, multimodal features are merged:

$$\Phi_M(x) = \sum_{m=1}^{M} \phi_m(\vartheta_m; x_m). \tag{5.20}$$

Reconcilement Regularization: To promote diversity between the updated modality and the rest, ReconBoost incorporates reconcilement regularization, which penalizes redundancy between modalities. This is achieved using the KL-divergence [128] term:

$$\tilde{\mathcal{L}}^s\left(\phi_m(x_m), y\right) = \frac{1}{N}\sum_{i=1}^{N}\left[\ell\left(\phi_m(\vartheta_m; x_m), y_i\right) - \lambda \cdot D_s\left(\Phi_{M/m}(x_i), \phi_m(\vartheta_m; x_m)\right)\right], \tag{5.21}$$

where ℓ represents the agreement term, $\Phi_{M/m}(x_i) = \sum_{j\neq m} \phi_j(\vartheta_j; x_j)$ is the contribution from all modalities except m, and D_s is the KL-divergence term defined as:

$$D_s\left(\Phi_{M/m}(x_i), \phi_m(\vartheta_m; x_m)\right) = \mathrm{KL}\left(\Phi_{M/m}(x_i) \parallel \phi_m(\vartheta_m; x_m)\right). \tag{5.22}$$

The parameter λ controls the trade-off between agreement and diversity. This term ensures that the updated modality aligns with but remains distinct from the ensemble prediction of the other modalities. By maintaining diversity, the model leverages complementary strengths of each modality while minimizing competition.

Boosting Perspective: ReconBoost draws inspiration from Gradient Boosting [129] [130], where each learner corrects errors made by previous learners. However, unlike traditional boosting methods that preserve historical models, ReconBoost discards old models to prevent overfitting in over-parameterized deep learning setups. The boosting-inspired objective is:

$$\tilde{\mathcal{L}}^s\left(\phi_m(x_m), y\right) \iff \mathcal{L}\left(\phi_m(x_m), -\nabla_{\Phi_{M/m}} \ell\left(\Phi_{M/m}(x), y\right)\right), \tag{5.23}$$

where $-\nabla_{\Phi_{M/m}} \ell$ represents the residual to be minimized by the current modality learner.

Memory Consolidation Regularization (MCR): Memory Consolidation Regularization (MCR) ensures that predictions from the updated modality do not deviate significantly from the previous modality, preserving learned knowledge and enhancing the performance of weaker modalities. The MCR term is:

$$L_{\text{MCR}} = \frac{1}{N} \sum_{i=1}^{N} \|\nabla_{\vartheta_m} \ell(F_m(x_m^i), y_i) - \nabla_{\vartheta_{m-1}} \ell(F_{m-1}(x_{m-1}^i), y_i)\|^2, \tag{5.24}$$

where ℓ is the task-specific loss. The MSE is used to calculate the squared difference between the gradient of the current modality learner (∇_{ϑ_m}), which represents how the current learner is optimizing its task, and the gradient of the previous modality learner ($\nabla_{\vartheta_{m-1}}$), which captures the optimization direction of the previous learner. The MSE computes the average of the squared differences between these gradients for all training samples. It penalizes large deviations between the gradients of the two learners, enforcing similarity in their optimization behavior.

Global Rectification Scheme (GRS): The Global Rectification Scheme (GRS) prevents previously updated modalities from being stuck in local minima by allowing them to continue adjusting based on the current residual error. The parameter update for modality m is:

$$\vartheta_m^t = \vartheta_m^{t-1} - \eta \cdot \nabla_{\vartheta_m^{t-1}} L(\Phi_M(x), y). \tag{5.25}$$

The complete objective combines agreement, reconcilement, MCR, and GRS terms:

$$\mathcal{L}_{\text{all}} = \sum_{m=1}^{M} \left[\mathcal{L}(\phi_m(x_m), y) - \lambda \cdot D_{\text{KL}}\left(\Phi_{M/m}(x) \parallel \phi_m(x_m)\right)\right] + a \cdot \mathcal{L}_{\text{MCR}} + \mathcal{L}_{\text{GRS}}. \tag{5.26}$$

Here, a is the weight for the MCR term. ReconBoost leverages alternating updates, reconcilement regularization, and enhancement schemes to prevent modality competition and ensure balanced learning. By alternating updates and introducing memory consolidation and rectification strategies, it ensures that all modalities contribute meaningfully to the final prediction. This makes it a robust framework for tackling the challenges of unbalanced multimodal learning.

The difference from previous methods stands in the presence of an ensemble model designed to handle our multi-modal inputs. Instead of a simultaneously multi-modal fusion approach, we employ an alternating learning paradigm. The ensemble net holds multiple models, each of which corresponds to one out of two or three modalities present. It also uses a common feature space. A common head (classifier) is used to map the aligned feature space to the final output. The ensemble net is responsible for boosting and forward propagation. The boosting loss is used during the backward pass to compute gradients and update the model's weights. It is expressed as the combination of the direct loss, that ensures the current model's predictions are accurate with respect to the ground truth, and the residual loss, which ensures that the current model complements the ensemble of previous models by focusing on correcting their errors. Two weights control the relative importance of the direct loss and residual loss, respectively. There is a common

learning rate for the boosting scheme. In the training process, the gradient alignment loss is computed by first obtaining the output from the previous model. Then, the gradient alignment loss is calculated as the mean squared error (MSE) between the detached softmax outputs of the current model and the previous model. Next, the total loss is computed by adding the boosting loss and the gradient alignment loss multiplied by a factor a. The MSE loss acts as a regularizer that minimizes the difference between the current modality's predictions and the previous modality's predictions. This encourages consistency across modalities and prevents any single modality from dominating the learning process. After each stage, the ensemble has been trained with a new modality. The GRS refines the entire ensemble network to ensure that the integrated prediction across all modalities is globally consistent and accurate. It does this by performing several epochs of fine-tuning across the whole ensemble. The GRS is controlled by the number of epochs after boosting and a specified learning rate.

5.4 Summary

The aforementioned methods were selected for their ability to tackle core challenges from different perspectives, ensuring a comprehensive exploration of solutions. OGM-GE and AGM focus on dynamic gradient modulation, while PMR and ReconBoost emphasize loss rebalancing strategies to mitigate competition and enhance weaker modalities. This selection includes methods designed for either bimodal or trimodal models, enabling the examination of various modality imbalance scenarios and their impact across different combinations of modalities. The selected methods also reflect state-of-the-art approaches in handling modality imbalance dynamically, with each contributing unique strengths: noise-based generalization (OGM-GE), Shapley-inspired disentanglement (AGM), prototype-based rebalancing (PMR), and reconcilement with alternating updates (ReconBoost). Together, they form a robust foundation for addressing the challenges of multimodal learning, as summarized in Table 5.1.

Method	Technique	Frequency	Trigger	Mechanism	Modalities
OGM-GE	Gradient Modulation	Every iteration	Applied between specific epochs	Gradients are scaled for each modality using coefficients from discrepancy ratios. These ratios guide the modulation process by identifying the stronger modalities. The dominant modality is penalized.	Audio, Vision
	Generalization Enhancement	Every iteration	Optionally activated between specific epochs	Incorporates Gaussian noise injection into gradients to promote robustness and improve generalization.	
	Discrepancy Ratio	Every Iteration	During forward pass	Quantifies the discrepancy between modalities during training.	
	Learning Rate Decay	Every epoch	After lr_decay_step epochs	Decays learning rate by a factor.	
ACC	Gradient Modulation	Every iteration	Applied between specific epochs	Gradients are scaled for each modality using coefficients (coeff_a for audio, coeff_v for visual). The weak modality is boosted.	Audio, Vision
	Discrepancy Ratio	Every Iteration	During forward pass	Quantifies the discrepancy between modalities during training.	
	Learning Rate Decay	Every epoch	After lr_decay_step epochs	Decays learning rate by a factor.	
AGM	Modality Masking with Shapley values	Every iteration	During forward pass	Performs three forward passes: one with both modalities, one without text and one without audio, to disentangle individual modality contributions based on Shapley-inspired values.	Audio, Text
	Competition Strength	Every iteration	During forward pass	Quantifies the strength of each modality present based on the mono-modal contribution during each forward pass.	
	Gradient Modulation	Every iteration	Applied between specific epochs	Gradients of each modality are scaled separately using coefficients computed from competition strength ratios. These ratios guide the modulation process by identifying and addressing stronger, more competitive modalities.	
	Learning Rate Decay	Every epoch	After lr_decay_step epochs	Decays learning rate by a factor.	
	Adaptive Gradient Clipping	Every iteration	Activated when gradient values exceed predefined thresholds	Scales gradients to a maximum norm of 1.0, stabilizing updates.	
PMR	Prototypical Loss Adjustment	Every iteration	Applied between specific epochs	Measures the classification error for each modality using the distance between unimodal features and their category prototypes	Audio, Vision
	Prototypical Regularization Term	Every iteration	Optionally activated between specific epochs	Reduces the entropy of the faster-learning modality's class distributions to mitigate dominance	
	Imbalanced Ratio	Every Iteration	During forward pass	Quantifies the imbalance between modalities during training.	
	Learning Rate Decay	Every epoch	After lr_decay_step epochs	Decays learning rate by a factor.	
RECONBOOST	Alternating Technique	Every training stage	When a new modality is selected for training.	Trains one modality learner at a time, allowing the ensemble to focus on weak or underperforming modalities.	Text, Audio, Vision
	Ensemble Forward Pass	Every step	During forward pass	Aggregates predictions from all modalities in the ensemble.	
	Boosting Scheme	Every stage	At every training stage when new modalities are added.	Dynamically adjusts the contribution of each modality using a boost rate parameter to refine ensemble predictions.	
	Global Rectification Scheme	After every stage	From first stage	Adjusts the ensemble model globally by fine-tuning all added modalities together using cross-entropy loss.	
	Memory Consolidation Regularization	Every iteration	During backward pass	Regularizes new modality outputs using soft labels derived from the ensemble's earlier predictions to align new knowledge with prior knowledge through Mean Squared Error.	

Table 5.1. *Overview of Optimization Methods central to our research. The table presents different methods along with their techniques, frequency of application, triggering conditions, underlying mechanisms, and the modalities they operate on the original implementations.*

Chapter 6

Experimental Results and Analysis

6.1 Introduction

This chapter presents a unified evaluation of the proposed methods to address optimization challenges in multimodal learning for sentiment analysis tasks. By using the same unimodal encoders across all experiments, we aim to investigate the optimization dynamics in a consistent and controlled manner. The evaluation focuses on two main categories of methods: Dynamic Gradient Adjustment Methods and Loss-Based Optimization Methods, while analyzing their sensitivity to key experimental concepts.

The Dynamic Gradient Adjustment Methods include On-the-fly Gradient Modulation with Generalization Enhancement (OGM-GE) [26], a method designed to dynamically adjust gradients to improve generalization performance and Adaptive Gradient Modulation (AGM) [27], a model quantifying modality strength to balance modality contributions during training. The Loss-Based Optimization Methods include Prototypical Modal Rebalance (PMR) [20], that guides optimization based on modality prototypes and ReconBoost [21], a framework employing a loss-alternating paradigm to iteratively optimize between reconstruction and classification objectives.

6.2 Experimental Setup

This section describes the experimental framework used to evaluate the proposed methods. We provide details on the evaluation metrics, baseline methods, and unimodal encoders utilized in all experiments. Furthermore, we outline the training configurations, including the data feature extraction, ensuring a consistent and reproducible evaluation process across all methods.

6.2.1 Evaluation Metrics

The evaluation of the models for the multiclass sentiment analysis task is based on Multiclass Accuracy. This metric calculates the percentage of correctly predicted samples over the total number of samples. Accuracy provides an intuitive measure of the model's overall performance across multiple sentiment classes. It is formally defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \times 100 \tag{6.1}$$

6.2.2 Datasets and Feature Extraction

For our experiments, we use the unaligned version of the CMU-MOSI and CMU-MOSEI datasets, where features are extracted independently for each modality and have differing sequence lengths. Both datasets provide pre-extracted features[1] for three modalities: text, audio, and video. The text features are contextualized embeddings extracted using BERT-base [43], with each word represented as a 768-dimensional vector capturing semantic and contextual information. Audio features are extracted using the COVAREP toolkit [131], providing frame-level prosodic and spectral characteristics such as pitch and energy. Visual features are obtained using OpenFace [132], which computes frame-level facial landmarks, action units, and head poses, capturing non-verbal cues like facial expressions.

The datasets are divided into training, validation, and test sets for fair evaluation. For CMU-MOSI, the splits consist of 1,281 samples for training, 229 for validation, and 689 for testing, with a development set of 100 samples extracted from the training split for auxiliary calculations during training, leaving 1,181 samples in the train set. Similarly, for CMU-MOSEI, the splits include 16,265 samples for training, 1,869 for validation, and 4,643 for testing, along with a development set of 200 samples from the training split leaving 16,165 training samples. The use of the development set will be explicitly mentioned wherever it applies in the experiments. We perform sentiment classification categorizing samples as negative (0), neutral (1), or positive (2) for the CMU-MOSI dataset. For the CMU-MOSEI dataset, the continuous range of sentiment labels, originally spanning from -3 (most negative) to $+3$ (most positive), is divided into 7 classes to enable classification tasks following [133].

6.2.3 Unimodal Encoders

In this section, we briefly present the model setup used in our experiments. For each modality included in all subsequent experiments, we employ an LSTM as the encoder. The encoder configurations for each dataset are shown below.

Modality	**CMU-MOSI Dataset**			**CMU-MOSEI Dataset**		
	Hidden Size	**Layers**	**Output Size**	**Hidden Size**	**Layers**	**Output Size**
Text	64	1	32	64	1	64
Audio	16	1	16	16	1	16
Video	32	1	32	32	1	32

Table 6.1. *LSTM Encoder Configurations for Each Modality in the CMU-MOSI and CMU-MOSEI Datasets. All LSTM layers have a dropout rate of 0.0.*

[1] GitHub Repository: `https://github.com/thuiar/MMSA`

6.2.4 Baseline Methods

We conduct experiments using three multimodal baseline approaches commonly employed in the literature: ensembles with soft voting, uni-modality prefinetuned models, and joint training with concatenation.

Ensembles with Soft Voting: Separate models are trained for each modality, one for audio, one for video and one for text input modality. After training, the predictions from these models are combined using an ensemble method with soft voting. Soft voting involves averaging the probabilistic outputs of individual models, and the class with the highest average probability is selected as the final prediction. This approach leverages the strengths of individual modality-specific models while maintaining simplicity.

Uni-Modality Pre-finetuned Models: In this setup, separate models are pre-trained on individual modalities before being fine-tuned on the target task. This method leverages the strengths of specialized pre-trained models for each modality. Each model operates independently, and the results are later combined. We choose concatenation fusion for the fine-tuning of the pre-trained encoders.

Joint Training with Concatenation: This method involves fusing features from all modalities into a single, unified representation through late concatenation. Specifically, features extracted from text, audio, and video are concatenated into a single vector, which is then used for downstream classification. The model is trained jointly on this combined representation, enabling it to learn cross-modal relationships directly. We test this approach following two distinct scenarios: (1) one joint learning rate and (2) individual learning rates, one for each modality and one for the fused representation.

These setups reflect distinct strategies for leveraging multimodal information and serve as valuable baselines for evaluating the performance of more advanced architectures. In all our baseline experiments, we consider three scenarios: audio-video bimodal case, text-video bimodal case and a trimodal case. Since the text modality appears dominant in our datasets we decide to investigate the scenario of two non-dominant modalities separately from the case were a dominant modality is present as in the text-video. As a fundamental baseline, we utilize Late Concatenation with joint training under a common learning rate for all modalities.

6.2.5 Training Details

The models in our experiments are trained using the Adam optimizer, varying learning rates and a ReduceLROnPlateau learning rate scheduler from PyTorch. The scheduler reduces the learning rate by a factor of 0.1 when the validation loss does not improve for a specified patience period. For the CMU-MOSI dataset, the patience is set to 5, while for the CMU-MOSEI dataset, it is set to 20. We use a batch size of 16 for CMU-MOSI and 32 for CMU-MOSEI. Early stopping is applied with a patience of 8 epochs. These configurations are applied across all experiments, and any deviations will be explicitly mentioned.

Cross-Entropy Loss is employed as the optimization objective across all tasks. In PyTorch, it is implemented using the `CrossEntropyLoss` function, which combines the softmax

operation and the negative log-likelihood loss. The loss is formally defined as:

$$L_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\log(p_{i,y_i}), \tag{6.2}$$

where p_{i,y_i} is the predicted probability for the correct class y_i of sample i, and N is the total number of samples. This loss function penalizes incorrect predictions proportionally to the confidence of the prediction, ensuring that the model learns to assign higher probabilities to the correct classes.

Learning rates were tuned separately for each model to ensure optimal performance, as different architectures and training methods demonstrated varying sensitivities to learning rate. A common validation-based grid search was used to select the learning rate for each setup. Our goal is to investigate the proposed optimization techniques and their effectiveness on our baseline model, not to make comparisons between them. For this cause, we represent the best models retrieved from each method implementation in our setup. All reported results represent the average performance across five different seeds. During training, the best-performing model was identified and saved based on the validation loss, which served as the primary metric for model selection and early stopping. To ensure the reproducibility and robustness of our results, all experiments are conducted using the same 5 random seeds. This allows us to report both the mean performance and the standard deviation (std), capturing the variability introduced by stochastic training processes. All experiments are performed on a system equipped with an NVIDIA GeForce GTX 1080 Ti GPU with 12GB VRAM. The GPU enables efficient training for both datasets, with each experiment completing within a reasonable computational time. Detailed hyperparameter settings for each experiment can be found in Appendix A, ensuring reproducibility and consistency across evaluations.

6.2.6 Training Configurations for Method Evaluation

Key factors such as batch size selection, optimizer choice, training duration, and extended modulation are analyzed in targeted experiments to assess their impact on model performance, convergence, and modality interactions. These evaluations aim to provide better insights into how different training strategies influence the multimodal optimization methods.

Frequency of Optimization Updates: The frequency of optimization updates significantly impacts the learning dynamics of neural networks, influencing convergence speed, stability, and generalization. More frequent updates, such as those with smaller batch sizes or higher update rates, allow the model to adapt quickly to gradient changes but can introduce high variance, leading to instability or noisy optimization. On the other hand, less frequent updates, such as larger batch sizes or delayed gradient adjustments, provide a more stable optimization but may slow down convergence and struggle in dynamic learning settings. Striking a balance in update frequency is crucial, as it affects gradient estimation accuracy and overall optimization efficiency. Here we experiment with the frequency of performing parameter updates during training, which determines how

often the model's weights are adjusted based on computed gradients.

Batch Size: The choice of batch size directly impacts the variance of gradient estimates, influencing this way the behavior of stochastic optimization algorithms. Studies [44] [45] [46] have shown that small or moderate batch training tends to converge to flat minima farther from the initial state, while large batch training often correlates with convergence to sharp minima closer to the initial state, resulting in poorer generalization on test datasets. Drawing from these observations and implementations described in works like Self-MM [47] and the Multimodal Multi-Loss Fusion Network (MMML) [48], we adopt a batch size of 16 for CMU-MOSI and 32 for CMU-MOSEI to align with dataset characteristics. For CMU-MOSI, a batch size of 16 balances gradient stability, convergence efficiency, and computational practicality. Smaller batch sizes, such as 8, often produce noisier gradients, potentially leading to slower convergence, whereas a batch size of 16 maintains smoother gradient estimates. Additionally, research indicates that moderate batch sizes like 16 or 32 effectively leverage mini-batch gradient descent, accelerating convergence while preserving generalization. In setups utilizing concatenation fusion, where features from multiple modalities are combined into a larger representation, like ours, a batch size of 16 efficiently handles fused representations without exceeding memory limits or slowing down computation. This makes it a practical choice for multimodal architectures. For CMU-MOSEI, we select a larger batch size of 32 due to the greater scale of the dataset. A batch size of 32 provides the same advantages described earlier, but it also allows for more comprehensive sampling within each batch. Experiments using the development set aim to assess the sensitivity of the methods to batch size and explore a more adaptable way to employ them without being restricted by batch size. Ultimately, the goal is to develop methods that remain effective regardless of batch size constraints, while avoiding bias toward modality characteristics or learning dynamics. This is achieved by calculating the discrepancy (OGM-GE, PMR) and strength (AGM) metrics on data that the models have not seen, ensuring an accurate evaluation of their performance. For that cause we examine not only the performance but the trends of the ratios and loss during training.

Choice of Optimizer: The main difference between Adam and SGD lies in the way they handle learning rates and update parameters. SGD uses a fixed global learning rate for all model parameters and updates them based on the current gradient. SGD, especially with momentum, often leads to better generalization by converging to flatter minima [134]. On the other hand, Adam dynamically adapts the learning rate for each parameter by maintaining moving averages of both the gradient (first moment) and the squared gradient (second moment). This is helpful in our case, since the parameters in a multimodal network may converge at different rates. Adam, also, excels in handling sparse gradients and typically converges faster due to its adaptive updates, making it ideal for tasks with noisy or sparse gradients [42]. Another reason we choose Adam as the default optimizer in our experiments is that we use pre-extracted text features are from BERT. BERT text features are often pretrained using Adam or AdamW. BERT models are typically pretrained using these optimizers due to their adaptive learning rate capabilities and effective handling of sparse gradients [43]. Using Adam for the multimodal network

maintains consistency in optimization dynamics. In this study, we compare SGD and Adam optimizers to analyze their impact on multimodal learning. By evaluating model behavior under these two distinct optimization strategies, we aim to understand their effectiveness in balancing stability, convergence speed, and generalization of the examined methods. These experiments are conducted while keeping all other training parameters constant, with only the learning rate adjusted accordingly to ensure fair evaluation for each optimizer.

Prolonging the training duration: Training duration is a critical factor in optimizing multimodal models, as longer training can influence both model performance and computational efficiency. Extended training may lead to better convergence, particularly for models using SGD, which typically requires more iterations to achieve stability. However, excessive training can also result in overfitting, where models memorize training data rather than learning generalizable patterns. To examine this, we evaluate how prolonging training duration affects performance of some methods, focusing on loss trends, convergence stability, and potential improvements in accuracy. By systematically analyzing longer training schedules, we aim to determine whether increased training provides benefits or merely inflates computational costs without meaningful performance gains.

Prolonging modulation duration: Modulation techniques such as OGM, AGM, and PMR can be applied only during the early training epochs to stabilize learning and balance modality contributions. However, the impact of extending modulation throughout the entire training duration remains an open question. By maintaining modulation beyond the initial epochs, models may experience more consistent gradient adjustments, potentially leading to better long-term convergence and improved modality balance. Conversely, excessive modulation could disrupt natural learning dynamics, preventing the model from adapting effectively as training progresses. To investigate this, we analyze whether prolonged modulation leads to sustained improvements or diminishing returns, focusing on key metrics such as final accuracy, and loss trends. This evaluation helps determine if modulation should be restricted to early training phases or maintained throughout the entire training process for optimal performance.

6.3 Investigating Unimodal Learning Dynamics

Before applying advanced optimization techniques, we first conduct a baseline analysis to examine the learning dynamics of individual modalities including the evaluation of unimodal models independently to assess their standalone performance. These experiments serve as a foundation for understanding modality-specific behaviors and provide a reference point for tuning dynamic optimization methods examined later in this study.

Table 6.2 compares the effect of optimization frequency on unimodal performance across the CMU-MOSI and CMU-MOSEI datasets. For CMU-MOSI, updating every iteration leads to higher accuracy across all modalities, especially in video and text, while maintaining relatively stable loss values. The text modality in particular benefits from more frequent updates, achieving similar accuracy but significantly lower loss compared to updates every 4 iterations. For CMU-MOSEI, the optimization frequency does not cause

Modality	CMU-MOSI			CMU-MOSEI		
	lr	Accuracy (%)	Loss	lr	Accuracy (%)	Loss
Optimization Step Every 4 Iterations						
Audio	$1e^{-3}$	42.74 ± 2.66	85.85 ± 0.39	$5e^{-4}$	32.85 ± 0.37	167.83 ± 1.25
Video	$5e^{-4}$	47.87 ± 3.96	86.64 ± 1.03	$1e^{-4}$	32.42 ± 0.19	168.64 ± 0.64
Text	$1e^{-4}$	73.99 ± 0.95	65.79 ± 0.92	$1e^{-4}$	43.38 ± 0.55	134.98 ± 0.48
Optimization Step Every Iteration						
Audio	$5e^{-4}$	44.29 ± 4.30	86.14 ± 1.48	$5e^{-4}$	32.67 ± 0.34	168.08 ± 0.72
Video	$5e^{-4}$	50.15 ± 2.49	85.88 ± 2.00	$5e^{-4}$	32.37 ± 0.34	168.39 ± 0.67
Text	$5e^{-4}$	73.91 ± 1.41	64.81 ± 1.92	$5e^{-4}$	43.94 ± 0.61	133.84 ± 0.72

Table 6.2. *Best performance of unimodal models on CMU-MOSI and CMU-MOSEI datasets with different optimization frequencies. The results present accuracy and loss for each modality (Audio, Video, and Text) using two optimization settings: updates every 4 iterations and updates every 1 iteration.*

significant variations in accuracy, as values remain stable across settings. However, the audio and video loss values are slightly lower when updating every iteration, indicating that more frequent updates may contribute to better convergence for these modalities. Overall, more frequent updates (every iteration) appear to enhance learning stability, particularly for text and video modalities in CMU-MOSI, without negatively impacting CMU-MOSEI performance. CMU-MOSEI appears to have better-balanced learning dynamics, allowing all modalities to train effectively with the same learning rate. Choosing the correct learning rate per modality is crucial for optimizing multimodal models, as different data types have different gradient behaviors. Table 6.2 also provides useful information for tuning the learning rate of our models in later experiments. To establish the existence of a dominant modality we can compare the performance of the text modality of Table 6.2 with trimodal models of Table 6.10. We observe that the two performances are comparable, while audio and video accuracies remain significantly lower, indicating that text is the primary modality guiding the learning process.

6.4 Investigating Dynamic Gradient Adjustment

This section presents the experimental evaluation of the proposed Dynamic Gradient Adjustment Methods, which aim to address the challenges of optimizing multimodal models by dynamically adapting gradients during training. Specifically, we analyze the performance of the On-the-fly Gradient Modulation with Generalization Enhancement (OGM-GE) [26] and Adaptive Gradient Modulation (AGM) [27] methods. To gain a better understanding of their mechanisms and applicability, we also conduct experiments with various alterations of the original methods, examining their sensitivity to key hyperparameters and experimental settings. The results are compared against baseline models to evaluate their effectiveness in sentiment classification tasks.

6.4.1 On-the-fly Gradient Modulation with Generalization Enhancement

Here, we evaluate the performance of the OGM bimodal models, OGM-GE bimodal models, and the variation of the OGM method referred to as ACC. Our experiments focus on analyzing the frequency of optimization updates, the choice of optimizer, the impact of batch size, and the benefits derived from using a development set. Additionally, we investigate the effects of modulation duration on the CMU-MOSEI dataset and the impact of prolonged training duration on the CMU-MOSI dataset to gain a more comprehensive understanding of these methods. For consistency, reported results represent the best-performing model for each configuration, with detailed hyperparameter tuning procedures provided in Appendix A.

Audio-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Standard Training	Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	OGM	53.30 ± 3.12	85.31 ± 2.62	32.67 ± 0.33	166.76 ± 0.86
	OGM-GE	52.48 ± 1.27	84.88 ± 1.23	32.40 ± 0.30	167.12 ± 0.80
Optimization Update Every 4 Iterations	Baseline	53.67 ± 1.87	85.48 ± 1.26	32.59 ± 0.22	167.46 ± 0.77
	OGM	52.95 ± 1.65	85.79 ± 0.97	32.72 ± 0.33	166.59 ± 0.76
	OGM-GE	47.84 ± 1.52	87.15 ± 0.87	32.48 ± 0.17	167.94 ± 0.46
SGD Optimizer	Baseline	49.97 ± 2.59	85.38 ± 1.40	32.51 ± 0.16	167.24 ± 0.41
	OGM	49.82 ± 2.38	85.04 ± 1.25	32.37 ± 0.30	167.45 ± 0.42
	OGM-GE	49.42 ± 1.43	85.55 ± 0.71	32.63 ± 0.20	167.15 ± 0.60
Use of Development Set	Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	OGM	49.13 ± 5.20	86.11 ± 1.24	32.68 ± 0.26	166.52 ± 0.73
	OGM-GE	50.79 ± 3.98	85.40 ± 1.40	32.31 ± 0.14	167.81 ± 0.56

Text-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Standard Training	Baseline	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	OGM	75.04 ± 0.96	63.54 ± 0.69	44.15 ± 0.43	133.15 ± 0.35
	OGM-GE	73.50 ± 0.79	64.44 ± 1.26	43.58 ± 0.40	133.74 ± 0.15
Optimization Update Every 4 Iterations	Baseline	73.32 ± 0.86	67.20 ± 0.83	43.66 ± 0.38	133.97 ± 0.41
	OGM	73.64 ± 1.19	66.65 ± 1.08	43.83 ± 0.40	133.70 ± 0.51
	OGM-GE	73.85 ± 1.04	66.41 ± 0.80	43.94 ± 0.39	133.91 ± 0.35
SGD Optimizer	Baseline	73.21 ± 2.81	63.91 ± 2.15	44.46 ± 0.54	131.12 ± 0.40
	OGM	73.53 ± 1.79	65.11 ± 3.86	44.39 ± 0.48	131.93 ± 0.86
	OGM-GE	74.17 ± 0.90	64.08 ± 1.73	44.33 ± 0.72	131.88 ± 0.71
Use of Development Set	Baseline	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	OGM	74.49 ± 0.86	64.37 ± 1.17	44.16 ± 0.24	132.02 ± 0.82
	OGM-GE	74.49 ± 0.79	63.54 ± 0.56	43.45 ± 0.52	132.99 ± 0.79

Table 6.3. *Performance of Audio-Video and Text-Video models on the CMU-MOSI and CMU-MOSEI datasets using OGM and OGM-GE methods under various training configurations: optimization updates every iteration, optimization updates every 4 iterations, SGD optimizer, and the use of a development set. Baseline model represents joint training with late concatenation fusion.*

Varying frequency of optimization updates: For this experiment, optimization updates were applied either every iteration or every four iterations. Results of Table 6.3 indicate that updating every iteration generally leads to better performance in both accuracy and loss across all models and datasets. However, in the Audio-Video model on the CMU-MOSEI dataset, OGM achieves slightly lower accuracy with frequent updates com-

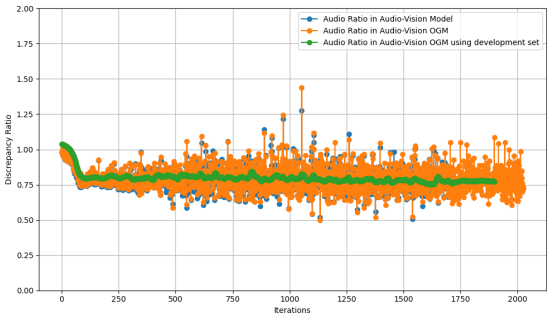
pared to updates every four iterations. This deviation may be attributed to the increased complexity of the CMU-MOSEI dataset, which might benefit from the stability provided by less frequent updates. This difference is marginal and may fall within the range of statistical variability. The findings suggest that more frequent updates allow the optimization process to better capture fine-grained patterns in the data and reduce the loss, improving model generalization. Frequent updates allow models to adapt more quickly to the nuances of the optimization landscape, reducing the risk of underfitting. As a result, updating every iteration will be considered as the preferred strategy in this thesis.

Choice of Optimizer: For the CMU-MOSI dataset, the Adam optimizer consistently achieves higher accuracy and lower standard deviation in vanilla models. While OGM and OGM-GE with Adam do not outperform the audio-vision vanilla model, they show notable improvements in the text-video scenario. Under SGD, OGM and OGM-GE perform worse than the audio-vision vanilla model but demonstrate improvements in accuracy and stability for text-vision models. For the CMU-MOSEI dataset, Adam enables OGM to surpass vanilla models, though OGM-GE underperforms across most scenarios. Conversely, with SGD, OGM-GE outperforms the audio-vision vanilla model but struggles to surpass the vanilla text-vision baseline. The Adam optimizer generally delivers better accuracy and stability across most cases, while SGD exhibits specific strengths in improving performance for text-vision models but often underperforms in comparison to vanilla baselines.

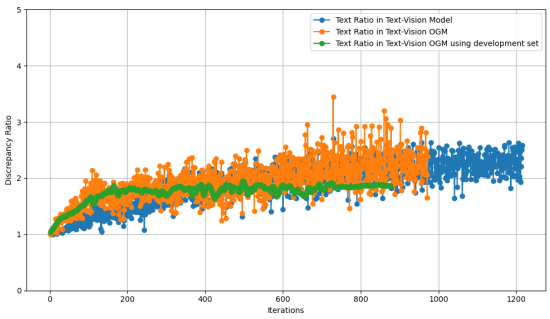
Use of Development Set: Next, we investigate the impact of batch size on the critical gradient coefficient calculations by incorporating the development set during training. To reduce the influence of constant recalculation due to batch size and better capture dynamic trends, discrepancy ratios (see Equation 5.1) and coefficients (see Equation 5.3) are updated every five iterations, while optimization updates are conducted every iteration using the Adam optimizer and the implemented gradient update method. For the CMU-MOSI dataset, the audio-vision OGM and OGM-GE models perform worse on the development set, indicating weaker generalization. The text-vision OGM model also degrades with the development set, while OGM-GE improves accuracy, reduces loss, and maintains moderate standard deviations. In the CMU-MOSEI dataset, the audio-vision OGM model shows similar performance with and without the development set but with lower variance. The text-vision OGM model benefits significantly, achieving higher accuracy, lower loss, and improved stability. It is worth observing some discrepancy ratio plots to further understand the impact of development set in stability of the algorithm.

In Figures 6.1a, 6.1b, we observe the audio discrepancy ratio and text discrepancy ratio for three different models on the CMU-MOSI dataset: the baseline model, the OGM model, and the OGM model on the development set. As shown, the trends become more stable, with reduced fluctuations, while still effectively representing the discrepancy between modalities without causing the model to favor the incorrect dominant modality. In the original OGM model, the audio discrepancy ratio remains predominantly below 1, indicating that audio is the weaker, underutilized modality in the model. When the development set is used, the OGM model successfully captures this trend while also smoothing the discrepancy ratio. This smoothness is crucial because the ratio is factored into the expression that multiplies the gradient weights, and the alpha parameter

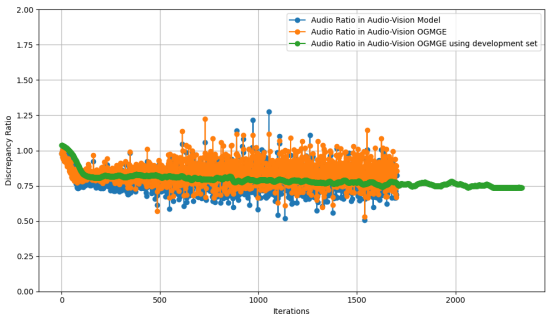
remains constant. Inconsistency in the coefficient causes the modulation of the gradients to vary unpredictably across iterations, potentially disrupting the optimization process. By smoothing the discrepancy ratio, the model ensures that the gradient multipliers do not deviate to misleading values, maintaining stable and effective optimization. The same observations can be made for the Text-Vision model, where the text discrepancy ratio remains above 1 when development set is used, indicating correctly that the dominant modality is text, but still managing to narrow down the fluctuations or large values of discrepancy ratio. Instability of discrepancy ratio may lead to erratic gradient modulation, potentially destabilizing the optimization process. A more stable discrepancy ratio ensures smooth and consistent gradient modulation, allowing for effective optimization and better convergence. The same trend is present in Figures 6.1c, 6.1d presenting the discrepancy ratios of the OGM-GE Audio-Vision and Text-Vision models. In the OGM-GE model, Gaussian noise is added to the gradients, which interacts with the modulation coefficient. If discrepancy ratio is stable, the noise interacts predictably with the gradients, enhancing exploration without disrupting convergence. If discrepancy ratios is unstable, the combination of noise and erratic modulation can amplify instability, further harming convergence. Thus, it is crucial to maintain a stable, yet representing trend for the discrepancy ratios.



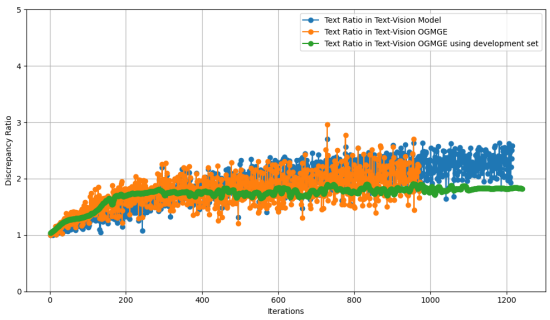
(a) *Audio Ratio in Audio-Vision OGM Model*



(b) *Text Ratio in Text-Vision OGM Model*



(c) *Audio Ratio in Audio-Vision OGM-GE Model*



(d) *Text Ratio in Text-Vision OGM-GE Model*

Figure 6.1. *Discrepancy Ratios in Audio-Vision and Text-Vision Models of CMU-MOSI. (a) Audio Ratio in Audio-Vision Baseline, OGM Model and OGM Model using development set, (b) Text Ratio in Baseline, OGM Model and OGM Model using development set, (c) Audio Ratio in Audio-Vision Baseline, OGM-GE Model and OGM-GE Model using development set, (d) Text Ratio in Text-VisionBaseline, OGM-GE Model and OGM-GE Model using development set. Ratios are no longer fluctuating while indicating the dominant modality.*

Accelerating the slow-learning modality: In Table 6.4, we compare the performance of the OGM method and its alteration, ACC. The ACC method focuses on enhancing the gradient magnitude of the weaker modality rather than penalizing the gradients of the dominant modality, as done in OGM. The results reveal that under the standard training configuration, ACC fails to outperform OGM across all models and datasets. Restricting the influence of the dominant modality, as implemented in OGM, is more effective for balancing multimodal learning than solely enhancing the weaker modality. When evaluated with the SGD optimizer, however, the ACC method demonstrates notable improvements. It manages to surpass OGM in several cases and delivers results that are comparable to the baseline models. In contrast, OGM struggles with SGD, failing to consistently outperform the baseline. This indicates that the ACC method may be better suited to optimization strategies like SGD, where adaptive mechanisms are absent, and enhancing the weaker modality provides stability to the training process. Overall, these findings highlight the complementary nature of OGM and ACC. While OGM performs better under standard configurations and adaptive optimizers like Adam, ACC shows promise in scenarios where simpler optimizers are employed.

Prolonging Training Duration: Previously, we noticed that calculations performed on the development set led to more stable ratio trends and enhanced the performance of our models compared to the standard on-the-fly calculations being applied after every iteration. Facilitating this stability, we experiment with the prolonging of the training process. Decreasing the learning rate to 8e-5 in Table 6.5a and increasing the patience of scheduler and early stopping did not manage to outperform results of Text-Video CMU-MOSI model of Table 6.3. However, experiments indicated more stable performance for the OGM-GE than the OGM model. This gave us the idea to prolong the training for a specified number of epochs without early stopping, but maintaining the use of development set.

Audio-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Standard Training	Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	OGM	53.30 ± 3.12	85.31 ± 2.62	32.67 ± 0.33	166.76 ± 0.86
	ACC	52.39 ± 2.42	85.98 ± 2.33	32.56 ± 0.37	166.70 ± 1.19
SGD Optimizer	Baseline	49.97 ± 2.59	85.38 ± 1.40	32.51 ± 0.16	167.24 ± 0.41
	OGM	49.82 ± 2.38	85.04 ± 1.25	32.37 ± 0.30	167.45 ± 0.42
	ACC	50.55 ± 1.80	84.86 ± 0.78	32.56 ± 0.31	167.56 ± 0.31

Text-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Standard Training	Baseline	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	OGM	75.04 ± 0.96	63.54 ± 0.69	44.15 ± 0.43	133.15 ± 0.35
	ACC	74.67 ± 1.68	64.36 ± 1.85	44.12 ± 0.21	133.13 ± 0.24
SGD Optimizer	Baseline	73.21 ± 2.81	63.91 ± 2.15	44.46 ± 0.54	131.12 ± 0.40
	OGM	73.53 ± 1.79	65.11 ± 3.86	44.39 ± 0.48	131.93 ± 0.86
	ACC	74.67 ± 1.02	62.96 ± 0.30	44.43 ± 0.54	131.46 ± 0.89

Table 6.4. *Performance of Audio-Video and Text-Video models on the CMU-MOSI and CMU-MOSEI datasets using OGM and ACC methods under various training configurations: optimization updates every iteration and SGD optimizer. Baseline model represents joint training with late concatenation fusion.*

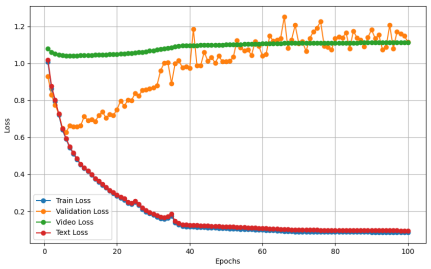
Method	Text-Video		Method	Audio-Video		Text-Video	
	Accuracy (%)	Loss (%)		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Prolonged Training with Early Stopping			**Prolonged Training for 100 Epochs**				
Baseline	74.35 ± 0.58	66.57 ± 0.49	Baseline	53.09 ± 2.22	85.54 ± 1.35	72.68 ± 2.08	68.54 ± 1.62
OGM	73.67 ± 1.19	66.92 ± 1.98	OGM	52.95 ± 2.25	85.61 ± 1.31	72.97 ± 1.99	68.44 ± 1.62
OGM-GE	73.64 ± 0.31	67.34 ± 0.42	OGM-GE	50.47 ± 2.22	86.82 ± 0.98	73.59 ± 0.64	67.56 ± 1.40

(a) *Decreased learning rate.* **(b)** *100 training epochs without early stopping.*

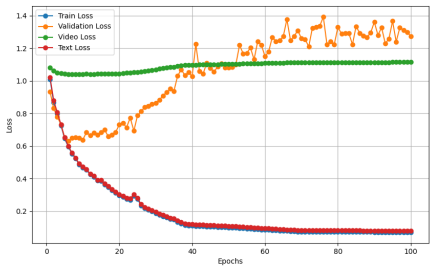
Table 6.5. *Prolonged training of OGM and OGM-GE bimodal models on CMU-MOSI dataset. Baseline model represents joint training with late concatenation fusion. Table (a) presents Text-Video models with decreased learning rate and increased early stopping patience and Table (b) Audio-Video and Text-Video models for 100 training epochs without early stopping.*

Experiments presented in Table 6.5b aim to examine the impact of the OGM-GE method during extended training periods, focusing on its potential implications in the later stages of training. To this end, we conducted experiments on the CMU-MOSI dataset using models with a development set for coefficient updates performed every 5 iterations. The learning rate was further decreased to 8e-5, modulation epochs were fixed at 50, and the total number of training epochs was set to 100 without early stopping. The audio-vision OGM or OGM-GE model fails to surpass the performance of the vanilla model for 100 epochs. Text-Vision OGM and OGM-GE model manage to surpass the vanilla regarding accuracy, while reducing loss and std values. Further increment of modulation epochs to 100 and training epochs to 200 resulted in the same performance for the Text-Vision models meaning they have already converged. Also, Text-Vision vanilla and OGM models indicated extreme overfitting behavior with training loss close to 0 and validation loss above 1. OGM-GE model limits the overfitting behavior as we will discuss and achieves the better performance among experiments of Table 6.5b.

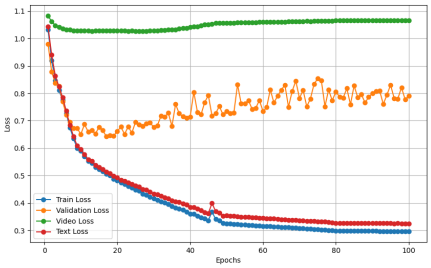
In Figure 6.2 we monitor the training loss of the fused representation, unimodal training losses and validation loss for the Baseline, OGM and OGM-GE Text-Vision models on CMU-MOSI for one run. The training loss of baseline model converges effectively, but there are fluctuations in validation loss. Vision modality learning indicates poor improvement, while text loss follows the trend of the joint training loss being the dominant modality. The OGM model-having the coefficients for the weight update calculated on the development set-indicate rapid decrease of the training loss, at a lower value compared to the baseline model. Validation loss also is lower, but fluctuates more prominently after epoch 40. Text loss also converges faster following the joint trend, while vision modality indicates similar behavior with baseline training. The training loss of the OGM-GE model (also using the development set for its coefficient updates) decreases rapidly, reaching the lower point compared to OGM and Baseline. Moreover the validation loss decreases steadily, indicating less fluctuations than the previous models. Vision learning process remains static, but the model indicates overall superiority in stability and performance compared to OGM and Baseline. The results highlight that the generalization enhancement in OGM-GE effectively reduces the training loss and mitigates oscillations in the validation loss.



(a) *Loss Trends of Baseline Text-Vision model.*



(b) *Loss Trends of OGM Text-Vision model.*



(c) *Loss Trends of OGM-GE Text-Vision model.*

Figure 6.2. *Training, Validation and Uni-modal Losses for Text-Vision models of CMU-MOSI: (a) Baseline, (b) OGM, and (c) OGM-GE.*

However, as illustrated in Figure 6.2, the application of early stopping proves critical. In the baseline and OGM models, the validation loss begins to rise significantly after epoch 20, diverging from the training loss and indicating overfitting. This underscores the importance of incorporating early stopping to maintain the model's generalization capability and prevent performance degradation in extended training scenarios. The OGM-GE model's superior performance in prolonged training stems from its ability to modulate gradients effectively while enhancing updates with Gaussian noise, but its behavior can vary depending on factors like early stopping and the frequency of coefficient updates. The use of Gaussian noise in optimization is well-known for helping escape sharp minima and guiding models toward flatter minima, which are associated with better generalization, as demonstrated in studies such as [44] [49]. However, this process requires sufficient training time for the noise-enhanced updates to refine the model's parameters, which early stopping may curtail. Early stopping, as explored by [51], is effective at preventing overfitting but may prematurely halt training, limiting the exploration of flatter minima. The frequency of coefficient updates also significantly influences model behavior. When coefficients are calculated on smaller batch sizes and applied at every iteration, their sensitivity to batch-specific noise can destabilize training, a phenomenon supported by Wilson et al. [50], which highlights the effects of small-batch stochastic gradient noise. In contrast, updating coefficients less frequently, as in the current setup (every five iterations), helps smooth out these fluctuations, ensuring more stable and consistent modulation of gradients. This aligns with research by Smith et al.[52], which demonstrates the advantages of periodic over per-iteration updates in optimization. Furthermore, this less frequent update strategy interacts synergistically with Gaussian noise, as the temporal consistency

of gradient modulation allows the noise to perturb gradients effectively, improving convergence and stability. Together, these factors explain why the combination of prolonged training, stable coefficient updates, and noise-enhanced exploration enables OGM-GE to achieve the best performance in the presence of dominant modality.

Method	Audio-Video		Text-Video	
	Accuracy (%)	Loss (%)	Acc (%)	Loss (%)
Modulation for First 5 Epochs				
Baseline	32.55 ± 0.44	166.74 ± 0.88	43.99 ± 0.39	133.11 ± 0.39
OGM	32.67 ± 0.33	166.76 ± 0.86	44.15 ± 0.34	133.11 ± 0.39
OGM-GE	32.40 ± 0.30	167.12 ± 0.80	43.58 ± 0.40	133.74 ± 0.15
Modulation During All Training Epochs				
OGM	32.59 ± 0.40	166.61 ± 0.93	44.02 ± 0.43	133.15 ± 0.47
OGM-GE	32.43 ± 0.29	167.30 ± 0.68	43.13 ± 0.53	134.83 ± 0.21

Table 6.6. *Performance of Audio-Video and Text-Video models on the CMU-MOSEI dataset with modulation applied for a specific number of epochs vs throughout all training epochs.*

Prolonging Modulation: Repeating the best experiments for the CMU-MOSEI audio-vision and text-vision, but now applying the modulation during the whole training did not indicate improvement in the performance, showing that the modulation application only for the first few epochs is a better choice. Prolonging the modulation phase introduces adjustments to gradients over a longer period. This could interfere with the model's ability to settle into optimal convergence paths, as the modulation modifies the gradient magnitudes or adds noise when using OGM-GE. Modulating the gradients early in training allows the model to benefit from stabilized updates during the high-variance initial phase of optimization, which is crucial for finding smoother minima and escaping sharp ones. During the early epochs, gradients tend to be noisy due to random initialization and high variance in parameter updates. Modulation at this stage can act as a stabilizer, helping to smooth gradient updates and control their magnitude. The modulation starts and ends define a window where the gradient updates are actively adjusted. A well-chosen modulation window (e.g., first 5 epochs) aligns with the period when gradients are most unstable, allowing the modulation mechanism to stabilize the training process, while the prolonged adjustment of gradients can interfere with the natural stabilization of the optimization process, particularly in later epochs when gradients are already small. Modulation mechanisms like OGM and OGM-GE are most effective when used strategically in the early stages of training. Extending their application into later epochs risks disrupting convergence, as gradients at this stage are already optimized for minimizing the loss function. This conclusion aligns with our previous observations: prolonging the training does not improve performance but highlights the presence of the effectiveness of Gaussian noise in the gradients during the later stages of training. However, by this point, our model has already converged, making the additional modulation unnecessary.

On-the-fly Gradient Modulation: Summary of findings

- **OGM:** Improves Text-Video performance but struggles to outperform the baseline Audio-Video models under standard training configurations.
- **OGM-GE:** Underperforms compared to both the baseline and OGM models in the standard setup. However, prolonged training with the development set allows OGM-GE to better leverage its potential, particularly in text-video models.
- **ACC:** Fails to surpass the OGM model when using the Adam optimizer but outperforms it when using SGD. When used with SGD improves model performance on CMU-MOSI.
- **Optimization Frequency:** More frequent parameter updates (every iteration) yield better results.
- **Choice of Optimizer:** The adaptive nature of Adam generally performs better than SGD in this setup. Notably, OGM-GE Text-Vision models benefit from the incorporation of SGD optimizer.
- **Development Set:** Enhances model stability by reducing discrepancy ratio fluctuations and preventing overfitting to previously seen data.
- **Prolonged Training:** Deploys the impact of Gaussian noise in the optimization process, improving stability.
- **Prolonged Modulation:** Does not improve performance, suggesting that early epochs are critical for effective gradient updates.

6.4.2 Adaptive Gradient Modulation

In this section, we evaluate the performance of the AGM bimodal models, building on insights gained from the previous evaluation of the OGM-GE method. Our analysis focuses on the effectiveness of AGM in handling both dominant and non-dominant modalities, the impact of optimizer selection, the influence of batch size, and the benefits derived from incorporating a development set. To ensure consistency, the reported results represent the best-performing model for each configuration. Detailed hyperparameter tuning procedures are provided in Appendix A for reproducibility.

Impact of Adaptive Gradient Modulation: Results in Table 6.7 indicate that AGM consistently demonstrates improved performance over the Baseline method across Text-Video Models of both datasets. For the Audio-Video Model of CMU-MOSEI, AGM exhibits better accuracy and a reduction in loss, indicating its capability to handle iterative updates more effectively. Audio-Video AGM Model on CMU-MOSI, however, fails to outperform baseline model, showing degraded accuracy and increased standard deviation. Results in Table 6.7 indicate that AGM consistently outperforms the Baseline method

across Text-Video Models for both datasets, demonstrating its effectiveness in optimizing multimodal interactions in the presence of strong, dominant modalities. For the Audio-Video Model, AGM shows mixed performance. On the CMU-MOSEI dataset, AGM achieves better accuracy and a noticeable reduction in loss compared to the Baseline, underscoring its ability to handle iterative updates effectively in scenarios where the audio and video modalities complement each other. However, on the CMU-MOSI dataset, AGM fails to surpass the Baseline, with degraded accuracy and increased standard deviation. This discrepancy suggests that AGM's performance may be influenced by dataset-specific characteristics, particularly in cases where the audio modality is less informative or dominant. It highlights the need for further tuning or adjustments to ensure consistent performance across different datasets and modality combinations.

Audio-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Standard Training	Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	AGM	53.73 ± 3.62	84.17 ± 1.35	32.71 ± 0.44	166.17 ± 0.64
SGD Optimizer	Baseline	49.97 ± 2.59	85.38 ± 1.40	32.51 ± 0.16	167.24 ± 0.41
	AGM	49.59 ± 1.17	85.15 ± 0.77	32.42 ± 0.23	166.94 ± 0.63
Use of Development Set	Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	AGM	52.25 ± 1.97	84.32 ± 1.24	32.71 ± 0.31	166.81 ± 1.01

Text-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Standard Training	Baseline	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	AGM	74.61 ± 0.86	63.83 ± 1.12	44.15 ± 0.39	132.58 ± 0.34
SGD Optimizer	Baseline	73.21 ± 2.81	63.91 ± 2.15	44.46 ± 0.54	131.12 ± 0.40
	AGM	73.76 ± 1.17	65.75 ± 1.31	44.10 ± 0.13	130.95 ± 0.81
Use of Development Set	Baseline	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	AGM	74.46 ± 1.93	64.26 ± 1.69	43.84 ± 0.11	133.26 ± 0.63

Table 6.7. *Performance of Audio-Video and Text-Video models on the CMU-MOSI and CMU-MOSEI datasets using AGM method under various training configurations: standard optimization updates every iteration, SGD optimizer, and the use of a development set. Baseline model represents joint training with late concatenation fusion.*

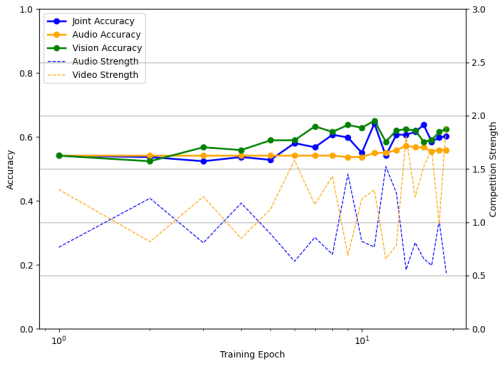
Choice of Optimizer: Experimentation with SGD optimizer evaluates the robustness and adaptability of the AGM method under a non-adaptive optimization strategy. While the Adam optimizer dynamically adjusts learning rates during training, SGD offers a more static approach, which may amplify the sensitivity of the model to hyperparameters and training dynamics. By testing AGM with SGD, as we did with OGM, OGM-GE and ACC models, we aim to investigate whether the method can effectively optimize multimodal interactions without relying on the adaptive capabilities of Adam. This experiment also provides insights into the method's behavior across different optimization strategies. For the Audio-Video Model, AGM under the SGD optimizer achieves comparable results to the Baseline with SGD, showing a slight improvement in loss but failing to surpass the Baseline in accuracy across both datasets. On the CMU-MOSI dataset, AGM under the SGD optimizer exhibits higher loss and a drop in accuracy compared to its performance with Adam optimizer. Similarly, on the CMU-MOSEI dataset, it still underperforms in terms

of accuracy relative to its performance under the Adam optimizer. For the Text-Video Model, AGM performs noticeably better than the Baseline under the SGD optimizer on both datasets, demonstrating its capability to effectively optimize text-video interactions under this setup. However, when compared to the results with Adam, AGM exhibits lower accuracy and higher loss, particularly on the CMU-MOSI dataset. On the CMU-MOSEI dataset, AGM performs reasonably well with SGD but does not achieve the same level of improvement seen with Adam. Overall, the results suggest that AGM is sensitive to the choice of optimizer, with its performance being more robust and effective particularly with Adam. AGM models with SGD struggle to replicate the same level of consistency and improvement seen with Adam-based updates

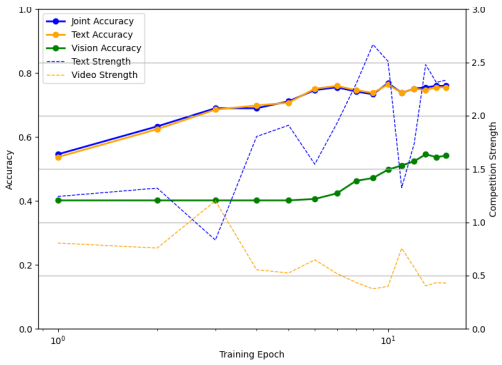
Use of Development Set: The purpose of incorporating a development set during training is to provide a more stable and consistent basis for calculating gradient coefficients, allowing the model to better capture the dynamic trends of both strong and weak modalities. By isolating the gradient calculations from the training batch, this setup ensures that the coefficients reflect broader trends rather than being overly influenced by batch-specific variations as illustrated in Figures 6.3, 6.4. Across the experiments, the inclusion of the development set leads to improved stability in optimization, which translates to more reliable performance and better generalization. This setup proves particularly effective in balancing the interactions between dominant and non-dominant modalities. For some configurations, the development set slightly improves accuracy or loss, particularly for setups involving weaker modalities, as it helps refine gradient coefficient calculations. However, in other cases, the results are comparable to or slightly worse than the baseline, indicating that the benefits of the development set may depend on the interplay between dataset characteristics and modality strength. This suggests that while the development set does not universally outperform the baseline, it contributes to a more consistent training process.

In Figure 6.3 Audio-Vision Baseline demonstrates steady joint accuracy but struggles with weak audio modality contributions. The CMU-MOSI Audio-Vision AGM Model improves joint accuracy compared to the Vanilla Model, leveraging dynamic adjustments in audio and video strengths during training. However, the strengths fluctuate significantly, indicating that the model is actively rebalancing modality contributions. This dynamic balancing capability highlights the AGM model's adaptability. The AGM Model with development set builds upon the strengths of the AGM model by introducing additional stability through the use of a development set. Audio strength steadily increases, while video strength stabilizes with minimal fluctuations. The model maintains high joint and aligned audio and video performances, showcasing a more balanced and consistent reliance on both modalities. The development set contributes to regularizing modality contributions, making this configuration the most robust and reliable between the three models.

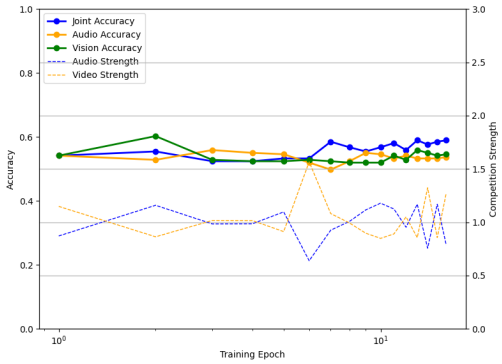
The Baseline Text-Vision Model in Figure 6.3 demonstrates steady improvement in joint accuracy but its reliance on the text modality is dominant, with text accuracy closely aligning with joint accuracy. However, it shows fluctuating text strength and a weak contribution from video, as video strength decreases steadily, reflecting the model's weakness



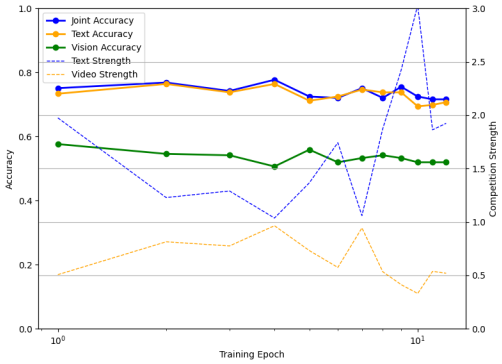
(a) *Baseline Audio-Vision Model.*



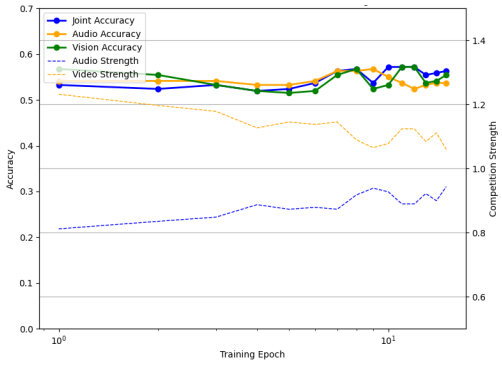
(b) *Baseline Text-Vision Model.*



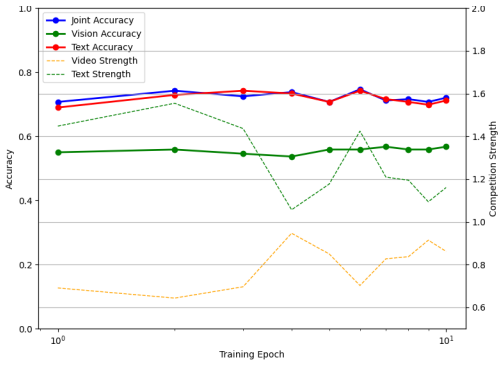
(c) *AGM Audio-Vision Model.*



(d) *AGM Text-Vision Model.*



(e) *AGM Audio-Vision Model using development set.*



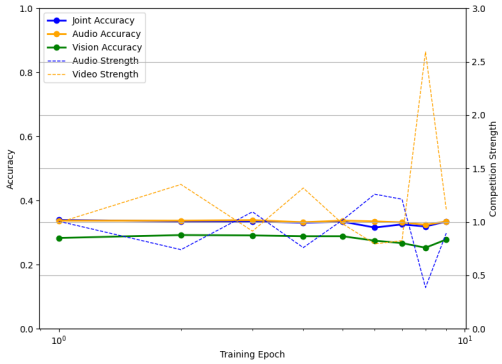
(f) *AGM Text-Vision Model using development set.*

Figure 6.3. *Comparison of Audio-Vision and Text-Vision Baseline, AGM and AGM with development set models on CMU-MOSI. The first row represents the baseline bimodal models, the second row the AGM model for the audio-vision and text-vision case and the third row the AGM models using development set for the update of modality strength and coefficients. Each figure includes the unimodal strengths and validation accuracies of the model for one run.*

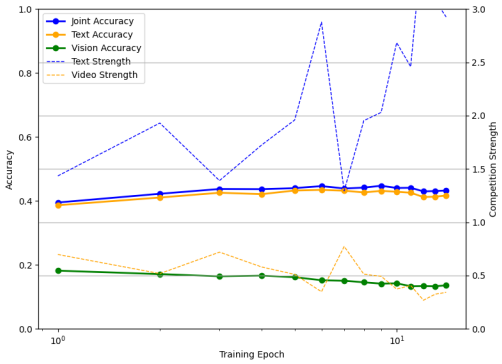
to dynamically balance modalities. In Text-Vision AGM Model text accuracy aligns with joint accuracy, but follow a decreasing trend indicating that the model struggles to learn effectively. This is supported by the unstable unimodal strengths that fluctuate having sharp drops and increases. Unimodal strengths fail to capture the relationship between the modalities leading to degrading trend performance compared to vanilla. AGM Model with a development set improves stability in unimodal strengths and manages to reduce the text strength, while increasing the video, as expected. The use of the development set helps achieve a more balanced reliance on video and text, stabilizing the contributions of both modalities and maintaining higher performance for the weak video modality, while increasing steadily the joint performance.

Starting with the Baseline Model in Figure 6.4, we observe that joint accuracy aligns closely with audio accuracy, both remaining steady throughout training. However, vision accuracy is consistently lower, indicating minimal contribution from the video modality. Interestingly, video strength exceeds audio strength, suggesting that while the model perceives video data as significant, it fails to extract meaningful insights. Accuracy refers to how well the model predicts using a particular modality (audio or video) when evaluated against ground truth labels. Strength reflects how much the model relies on a modality during the decision-making process. The mismatch between accuracy and strength arises because reliance (strength) does not directly correspond to predictive performance (accuracy). Audio strength remains low, reflecting the model's underutilization of the audio modality as well. In AGM Model the reliance on audio and video modalities is unstable, with significant fluctuations in both audio strength and video strength throughout training. Video strength occasionally spikes, but these do not lead to sustained accuracy improvements. As a result AGM Audio-Vision model struggles with instability in modality competition, leading to inconsistent reliance on video and less effective integration overall. It manages however to improve the video accuracy performance during training. In AGM with development set model, audio and video strengths stabilize significantly, with video strength remaining slightly higher than audio strength but without the instability observed in the AGM without a development set. This stability enables the model to integrate video information more effectively without overwhelming audio input, resulting in a better balance between the two modalities. As a results, the joint accuracy relies now more on the video modality. If the video modality contributes significantly to the model's accuracy depends on the quality and relevance of the input video data.

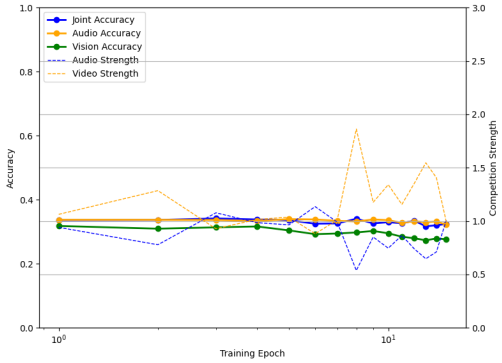
Baseline Text-Vision Model in Figure 6.4 shows a clear dependence on text data, as indicated by the close alignment between joint accuracy and text accuracy. Vision accuracy, however, remains consistently low, demonstrating the model's inability to effectively leverage video features. While text strength fluctuates significantly, video strength is consistently low, further highlighting the model's heavy reliance on textual input. Moving to the AGM Model, we observe marginal improvements in joint and text accuracies over the Baseline Model, but vision accuracy remains similarly low, reflecting the continued difficulty in integrating video features effectively. The AGM Model exhibits high instability in text strength, with frequent spikes and drops, while video strength improves slightly but remains much lower than text strength. This instability suggests that the AGM Model



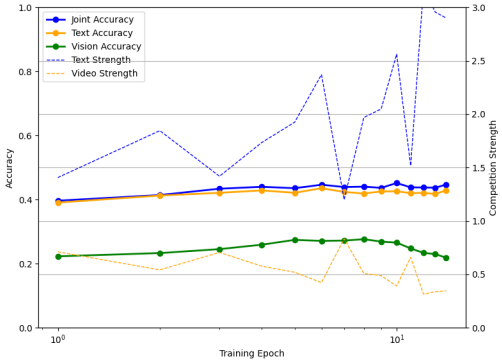
(a) *Baseline Audio-Vision Model on CMU-MOSEI.*



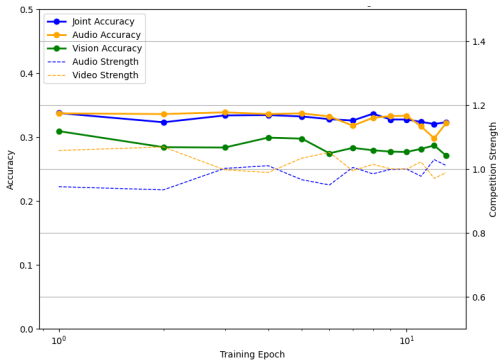
(b) *Baseline Text-Vision Model on CMU-MOSEI.*



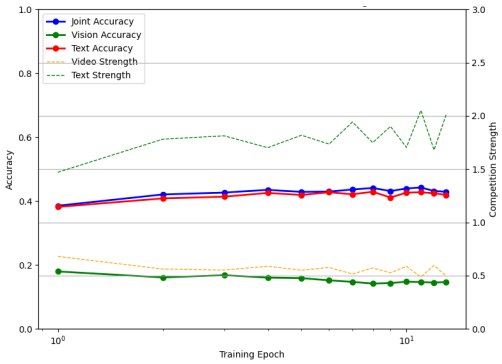
(c) *AGM Audio-Vision Model on CMU-MOSEI.*



(d) *AGM Text-Vision Model on CMU-MOSEI.*



(e) *AGM Audio-Vision Model using development set.*



(f) *AGM Text-Vision Model using development set.*

Figure 6.4. *Comparison of Audio-Vision and Text-Vision Baseline, AGM and AGM with development set models on CMU-MOSEI. The first row represents the baseline bimodal models, the second row the AGM model for the audio-vision and text-vision case and the third row the AGM models using development set for the update of modality strength and coefficients. Each figure includes the unimodal strengths and validation accuracies of the model for one run.*

struggles with balancing reliance on the two modalities and faces challenges in stabilizing the integration of video features. AGM Model with a Development Set improves stability in modality reliance compared to the other two models but still demonstrates a preference for textual input, with video playing a less critical role. The development set mitigates instability rather than fundamentally changing the model's modality integration, highlighting that further refinement is needed to fully leverage video features of CMu-MOSEI dataset.

Adaptive Gradient Modulation: Summary of findings

- **AGM:** Improves performance, but fails to surpass the baseline Audio-Video model on the CMU-MOSI dataset under standard training configurations.
- **Choice of Optimizer:** The adaptive nature of Adam outperforms SGD in this setup. AGM models trained with SGD do not exceed the performance of their corresponding baseline models.
- **Development Set:** This helps the model more accurately capture modality strength, improving the algorithm's accuracy as trends in modality strength indicate a reduction in the dominant modality. At the same time, it mitigates overfitting to previously seen data while maintaining performance comparable or better than the baselines.

6.5 Investigating Loss Based Optimization

Prototypical Modal Rebalance [20] and ReconBoost [21] are two approaches that address modality imbalance during training by utilizing the loss function through distinct mechanisms. This section examines the impact of various training configurations on their performance and evaluates their effectiveness against the baseline training approach, which employs a joint fusion loss to optimize multimodal interactions.

6.5.1 Prototypical Modal Rebalance

We apply the Prototypical Modal Rebalance optimization method to the audio-video and text-video models without incorporating the penalty through the entropy of the dominant modality as described in equation 5.16. We study the impact of the exponential moving average of the centroids, the choice of optimizer and the use of development set to overcome possible limitations by batch size. For detailed hyper-parameter tuning please refer to Appendix A.

Impact of PMR Method: The method fails to favor the audio-video models on either dataset, but improves accuracy of text-video models. This suggests that the method is highly affected by the relationship between modality features. The optimization guided by class prototypes appeared effective when a dominant modality is present, but failed otherwise.

Exponential Moving Average of Prototypes: Performance trends when the decay

rate of the Exponential Moving Average (EMA) for modality prototype centroids is greater than zero remain similar to previous observations. However, in this case, Text-Video models exhibit improved accuracy scores, while CMU-MOSEI shows a further reduction in loss, indicating that EMA contributes to better stability and enhanced modality representation in certain configurations of this method.

Audio-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Standard Training	Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	PMR	51.52 ± 2.85	85.24 ± 1.64	32.44 ± 0.44	166.65 ± 0.57
Exponential Moving Average of Prototypes	Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	PMR	51.55 ± 3.71	85.04 ± 1.85	32.21 ± 0.61	166.61 ± 0.75
SGD Optimizer	Baseline	49.97 ± 2.59	85.38 ± 1.40	32.51 ± 0.16	167.24 ± 0.41
	PMR	51.25 ± 1.85	85.74 ± 1.42	32.56 ± 0.31	167.56 ± 0.31
Use of Development Set	Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	PMR	53.76 ± 2.15	84.55 ± 1.14	32.44 ± 0.25	167.19 ± 0.92

Text-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Standard Training	Baseline	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	PMR	74.55 ± 0.79	63.95 ± 0.52	44.29 ± 0.26	132.11 ± 1.01
Exponential Moving Average of Prototypes	Baseline	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	PMR	75.51 ± 0.50	64.51 ± 1.42	44.29 ± 0.17	131.94 ± 0.56
SGD Optimizer	Baseline	73.21 ± 2.81	63.91 ± 2.15	44.46 ± 0.54	131.12 ± 0.40
	PMR	73.56 ± 1.83	63.79 ± 2.45	44.43 ± 0.54	131.46 ± 0.89
Use of Development Set	Baseline	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	PMR	74.72 ± 1.22	63.33 ± 1.95	44.60 ± 0.53	130.60 ± 0.36

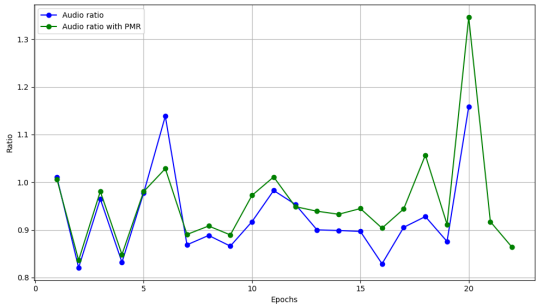
Table 6.8. *Performance of Audio-Video and Text-Video models on the CMU-MOSI and CMU-MOSEI datasets using PMR method under various training configurations: standard optimization updates every iteration, SGD optimizer, and the use of a development set. Baseline model represents joint training with late concatenation fusion.*

Choice of Optimizer: PMR models trained with Stochastic Gradient Descent (SGD) achieve performance comparable or better to their respective baselines, demonstrating their effectiveness in the given tasks. However, the Adam optimizer provides additional benefits for PMR models specifically on the CMU-MOSI dataset, leading to improved performance. On the other hand, for the CMU-MOSEI dataset, Adam yields results that are competitive with those obtained using SGD, suggesting that the choice of optimizer may have a dataset-dependent impact on model performance for this method.

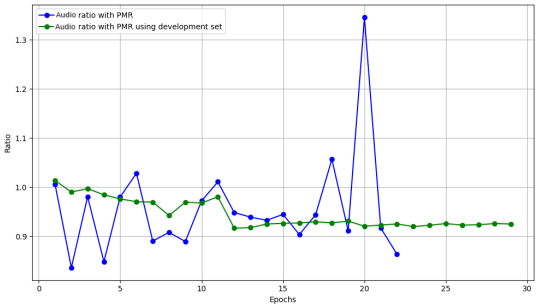
Use of Development Set: The use of the development set for calculating the imbalance ratio (see Equation 5.13) outperforms the Text-Video baselines, achieving lower loss and improved stability. Trends in the Figure 6.5 further confirm its positive impact on stabilizing metrics used to quantify modality contribution and dominance, reinforcing its role in enhancing model robustness. The comparison of imbalance ratios in PMR-based Audio-Vision and Text-Vision models of CMU-MOSI (see Figure 6.5) highlights the stabilizing effect of the development set. In the Audio-Vision model, the audio ratio fluctuates significantly without the development set, indicating instability in modality contribution. However, its inclusion leads to smoother trends, suggesting improved balance and opti-

mization stability, while showing that audio is the weaker modality in that case. Similarly, in the Text-Vision model, the text ratio exhibits a sharp increasing trend, reflecting its dominance over time. While PMR alone amplifies text contributions with noticeable fluctuations, the development set helps regulate this dominance, reducing extreme variations while maintaining its increasing trend. Figure 6.6 illustrates the impact of the development set on modality imbalance in Audio-Vision and Text-Vision PMR models of CMU-MOSEI. In the Audio-Vision model, the audio ratio exhibits significant fluctuations without the development set, indicating unstable modality contributions. However, its inclusion smooths the trends, demonstrating improved balance and reducing extreme variations. In the Text-Vision model, the text ratio increases over time, reflecting its dominance. While PMR alone amplifies text features with noticeable fluctuations, the development set stabilizes the trend, ensuring a more controlled and balanced modality interaction. These findings confirm that the development set mitigates instability, helping to regulate modality contributions and improving the robustness of multimodal learning.

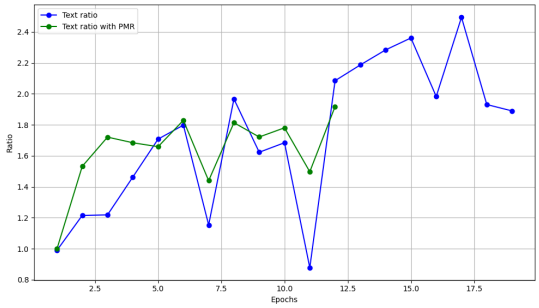
Additionally, the comparison between Audio-Video and Text-Video models highlights the dominant role of text in multimodal sentiment analysis. Text-Video models consistently achieve higher accuracy than Audio-Video models, reinforcing the idea that text carries the most informative features for sentiment classification. The performance gap between datasets further indicates that modality interactions vary based on sentiment distribution, emphasizing the need for dataset-aware optimization strategies.



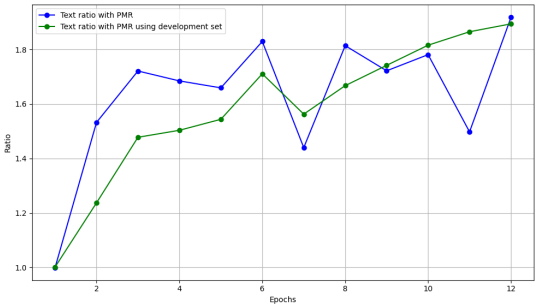
(a) *Imbalanced Audio Ratio in Audio-Vision PMR Model*



(b) *Imbalanced Audio Ratio in Audio-Vision PMR with Development Set*

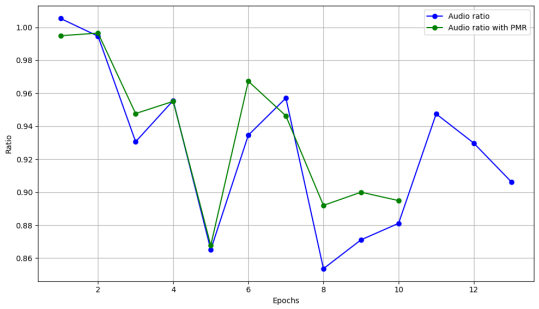


(c) *Imbalanced Text Ratio in Text-Vision PMR Model*

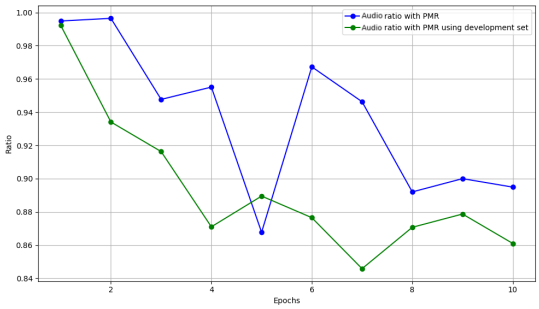


(d) *Imbalanced Text Ratio in Text-Vision PMR with Development Set*

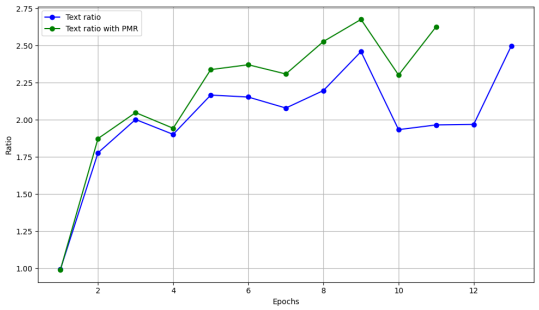
Figure 6.5. *Imbalanced Ratio Trends in PMR Models on CMU-MOSI dataset. (a) Audio Ratio in Audio-Vision PMR Model, (b) Audio Ratio in Audio-Vision PMR with Development Set, (c) Text Ratio in Text-Vision PMR Model, (d) Text Ratio in Text-Vision PMR with Development Set.*



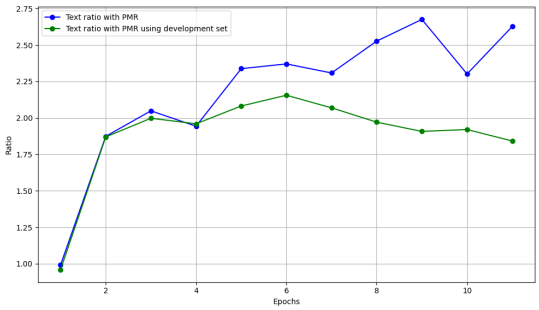
(a) *Imbalanced Audio Ratio in Audio-Vision PMR Model*



(b) *Imbalanced Audio Ratio in Audio-Vision PMR with Development Set*



(c) *Imbalanced Text Ratio in Text-Vision PMR Model*



(d) *Imbalanced Text Ratio in Text-Vision PMR with Development Set*

Figure 6.6. *Imbalanced Ratio Trends in PMR Models on CMU-MOSEI dataset. (a) Audio Ratio in Audio-Vision PMR Model, (b) Audio Ratio in Audio-Vision PMR with Development Set, (c) Text Ratio in Text-Vision PMR Model, (d) Text Ratio in Text-Vision PMR with Development Set.*

Prototypical Modal Rebalance: Summary of findings

- **PMR:** Enhances Text-Video model performance but does not outperform the baseline Audio-Vision model on either dataset under standard training configurations, suggesting that its effectiveness may be modality-dependent rather than universally applicable.
- **Exponential Moving Average:** Improves the performance of PMR Text-Video models, contributing to better stability and accuracy.
- **Choice of Optimizer:** PMR models trained with SGD achieve competitive with their corresponding baselines. Adam, however, benefits the PMR models on the CMU-MOSI dataset, while gives competitive results with the ones of SGD on CMU-MOSEI.
- **Development Set:** Consistently improves PMR model performance, leading to higher accuracy compared to the standard PMR setup. Although it does not surpass the Audio-Vision baseline on CMU-MOSI, it provides stabilized imbalanced ratio trends, effectively capturing modality dominance relationships, while providing with an unbiased quantification of the imbalance between modalities.

6.5.2 ReconBoost

We applied the ReconBoost method to the audio-vision and text-vision bimodal cases of the CMU-MOSI and CMU-MOSEI datasets, as well as the trimodal case involving audio, video, and text. Consistent with the original ReconBoost methodology, our experiments omitted learning rate scheduling, maintaining constant learning rates throughout training. These steady learning rates, applied equally to modality-specific and ensemble parameters, controlled the speed of parameter updates and impacted convergence. The experiments included 100 alternating stages, with each stage comprising one boosting epoch followed by one global rectification epoch. The boost rate, a critical parameter controlling the intensity of modality-specific corrections, was set to 1.0 to ensure balanced boosting across all modalities without additional scaling. This setup was designed to ensure robust optimization of modality contributions, providing a comprehensive evaluation of the ReconBoost method across different multimodal configurations. For detailed hyper-parameter tuning please refer to Appendix A.

Audio-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Adam Optimizer	Baseline	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	ReconBoost	47.29 ± 5.19	87.63 ± 2.73	33.14 ± 0.37	167.99 ± 1.18
SGD Optimizer	Baseline	49.97 ± 2.59	85.38 ± 1.40	32.51 ± 0.16	167.24 ± 0.41
	ReconBoost	49.88 ± 3.01	94.17 ± 3.60	32.42 ± 0.02	175.56 ± 0.80

Text-Video Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Adam Optimizer	Baseline	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	ReconBoost	74.79 ± 1.04	63.20 ± 1.26	44.78 ± 0.29	130.88 ± 0.75
SGD Optimizer	Baseline	73.21 ± 2.81	63.91 ± 2.15	44.46 ± 0.54	131.12 ± 0.40
	ReconBoost	73.45 ± 5.19	71.11 ± 7.90	44.37 ± 0.68	133.18 ± 1.09

Audio-Video-Text Model

Training Configuration	Method	CMU-MOSI		CMU-MOSEI	
		Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
Adam Optimizer	Baseline	74.87 ± 1.23	63.05 ± 1.65	44.46 ± 0.41	131.87± 0.60
	ReconBoost	75.09 ± 0.88	63.18 ± 1.30	44.42 ± 0.53	130.79 ± 0.76
SGD Optimizer	Baseline	73.24 ± 2.24	65.15 ± 2.35	44.80 ± 0.54	130.78 ± 0.48
	ReconBoost	73.07 ± 2.36	76.91 ± 5.95	44.30 ± 0.61	137.20 ± 4.35

Table 6.9. *Performance of Audio-Video, Text-Video and Audio-Video-Text models on the CMU-MOSI and CMU-MOSEI datasets using ReconBoost method under two different optimizers: Adam optimizer and SGD optimizer. Baseline model represents joint training with late concatenation fusion.*

The goal of the ReconBoost method is to address modality imbalance by leveraging a modality-alternating learning paradigm. In our experiments, we investigate the effectiveness of this method in managing unimodal contributions and its impact on multimodal fusion. Specifically, we aim to understand how ReconBoost balances modality-specific losses and ensemble corrections to optimize training dynamics and improve generalization. Additionally, we evaluate the role of the optimizer—comparing the static updates of

SGD to the adaptive learning of Adam—in shaping the training process and influencing the model's ability to converge effectively. Through this analysis, we aim to assess ReconBoost's potential as a robust solution for handling multimodal challenges and enhancing performance.

Regarding the CMU-MOSI dataset we observe that the Audio-Video models fail to surpass the vanilla performance. Specifically, the Audio-Video ReconBoost model with SGD optimizer increases its loss and std values leading to worse performance. The Audio-Video model with Adam demonstrates the same behavior, with major drop of its accuracy. The Text-Video model with SGD fails to improve performance, having increased loss and high stds for both accuracy and loss. The Text-Video model with Adam indicates improvement with increased accuracy and lower loss. The trimodal model with SGD also fails to improve the accuracy of baseline model, but the trimodal model with Adam indicates improvement with higher accuracy, lower loss and reduced std values.

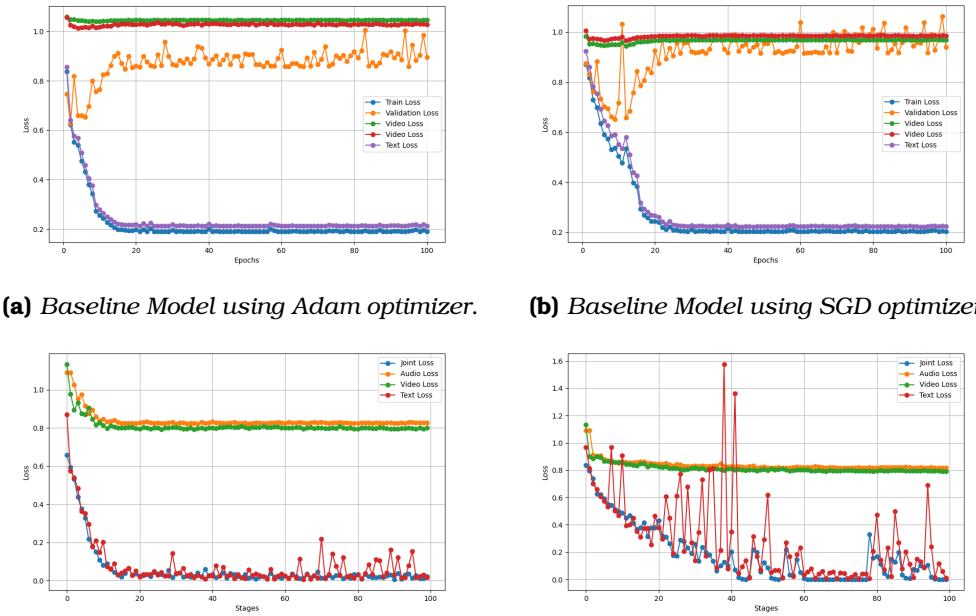
The trend of the losses (see Figure 6.7) during training of the trimodal model manage to achieve the goal of the method, which is to help weaker modalities follow the trend of the shared loss, without getting stuck at early epochs, aligning with the trend of losses during training of the original implementation. As we see, the baseline model with Adam maintain audio and video loss above 1, while training loss fluctuates between 0.1 and 0.2 after epoch 20. ReconBosot model with Adam on the other hand manages to drop the audio and video loss rapidly to 0.8 approximately, while the training loss approaches 0, suggesting a successful application of the code. It is also worth comparing the loss trends in our ReconBoost between the model using SGD and the model using Adam. ReconBoost model with SGD also achieves to drop the loss of each weak modality, without getting it stuck during early epochs. However, the training loss manages to approach 0 at a later epoch than the corresponding model with Adam. Training loss, also, indicates fluctuations during training struggling to converge.

Taking into account the model average performance results for 5 seeds and the trends of Figure 6.7, we see that the Adam optimizer for our ReconBoost setup is preferable on CMU-MOSI. The comparison of the training loss plots between Adam and SGD optimizers highlights key differences in model generalization. Adam optimizer exhibits smoother and more stable loss curves, indicating consistent convergence and effective learning. This stability suggests that the model trained with Adam is likely generalizing better. In contrast, the SGD optimizer shows more oscillations, particularly in the text loss, reflecting instability in training. These fluctuations indicate potential issues with generalization and less consistent learning, as the model appears to struggle more to converge smoothly across different seeds.

Although the stability of SGD compared to Adam is highly dependent on several factors, including the complexity of the model, the choice of hyperparameters such as the learning rate, data characteristics, and the presence of regularization, careful tuning can mitigate these issues. SGD is generally more prone to instability and oscillations, particularly in complex models and noisy environments. However, appropriate adjustment of the learning rate, use of momentum, and effective regularization strategies can significantly enhance its stability. Adam, with its adaptive learning rate and bias correction,

often provides more stable training with fewer fluctuations. Nevertheless, stability does not necessarily equate to better generalization performance. In our case, after tuning the hyperparameters for both optimizers—following the guidelines of the original implementation proposed by the authors and conducting a grid search on the validation set to determine the appropriate learning rate—we conclude that the Adam optimizer yields superior results in terms of both accuracy and generalization compared to SGD under these specific conditions.

On the CMU-MOSEI dataset, results with SGD show no improvement in accuracy and instead indicate an increase in loss values and standard deviations, leading to worse overall performance. In contrast, experiments with Adam on CMU-MOSEI demonstrate improvements compared to the vanilla models. Both the Audio-Vision and Text-Vision ReconBoost models achieve higher accuracies and lower standard deviations compared to the vanilla models. Notably, in the Text-Vision case, the models also achieve a reduction in loss. These findings support and validate our earlier conclusions. However, the trimodal model does not surpass the vanilla model in terms of accuracy but does achieve a lower loss, indicating improved optimization despite the accuracy plateau.



(a) *Baseline Model using Adam optimizer.* **(b)** *Baseline Model using SGD optimizer.*

(c) *ReconBoost Model using Adam optimizer.* **(d)** *ReconBoost Model using SGD optimizer.*

Figure 6.7. *Training, validation, and unimodal losses using Adam vs. SGD for a single run in Baseline and ReconBoost setup with three input modalities on the CMU-MOSI dataset.*

ReconBoost: Summary of findings

The method successfully reduces the loss of weaker modalities during the start of training and achieves improved performance in text-vision and trimodal settings when using the Adam optimizer. In contrast, SGD results in poorer performance with higher standard deviations, indicating unstable training dynamics.

6.6 Unified Comparative Analysis

In this section, we provide a conclusive summary of the best results achieved by the proposed methods under the standard training configurations outlined in Section 6.2.5, without additional alterations or experimental variations. Unlike previous sections where we investigate the behavior of the methods under different configurations, here we focus on presenting the peak performance of the algorithms. Our experimental setup included three unimodal encoders, with the baseline comparison conducted against a joint training approach using late concatenation. Additionally, we incorporate results from two other strong baselines—soft voting and uni-pre finetuned models—to provide a more comprehensive evaluation. The unified analysis in this section aims to highlight the peak performance achieved for each modality combination while identifying trends that demonstrate the potential of dynamic optimization methods to challenge established baselines in the literature.

Modality	**Method**	**CMU-MOSI**		**CMU-MOSEI**	
		Accuracy (%)	**Loss (%)**	**Accuracy (%)**	**Loss (%)**
Audio, Vision	Ensemble	47.96 ± 4.94	84.90 ± 1.50	32.63 ± 0.29	166.05 ± 0.75
	Uni-Pre Finetuned	51.46 ± 1.64	84.62 ± 1.07	32.55 ± 0.28	166.27 ± 0.49
	Late Concatenation	54.93 ± 1.19	84.10 ± 1.25	32.55 ± 0.44	166.74 ± 0.88
	OGM	53.30 ± 3.12	85.31 ± 2.62	32.67 ± 0.33	166.76 ± 0.86
	OGM-GE	52.48 ± 1.27	84.88 ± 1.23	32.40 ± 0.30	167.12 ± 0.80
	ACC	52.39 ± 2.42	85.98 ± 2.33	32.56 ± 0.37	166.70 ± 1.19
	AGM	53.73 ± 3.65	84.86 ± 1.81	32.71 ± 0.44	166.17 ± 0.64
	PMR	51.52 ± 2.85	85.24 ± 1.64	32.44 ± 0.44	166.65 ± 0.57
	ReconBoost	47.29 ± 5.19	87.63 ± 2.73	33.14 ± 0.37	167.99 ± 1.18
Text, Vision	Ensemble	73.35 ± 1.72	66.67 ± 1.82	43.94 ± 0.67	136.22 ± 1.17
	Uni-Pre Finetuned	75.72 ± 0.93	64.62 ± 1.19	45.22 ± 0.41	130.93 ± 0.64
	Late Concatenation	74.35 ± 0.58	66.57 ± 0.49	43.99 ± 0.39	133.11 ± 0.39
	OGM	75.04 ± 0.96	63.54 ± 0.69	44.15 ± 0.43	133.15 ± 0.35
	OGM-GE	73.50 ± 0.79	64.44 ± 1.26	43.58 ± 0.40	133.74 ± 0.15
	ACC	74.67 ± 1.68	64.36 ± 1.85	44.12 ± 0.21	133.13 ± 0.24
	AGM	74.61 ± 0.86	63.83 ± 1.12	44.15 ± 0.39	132.58 ± 0.34
	PMR	75.51 ± 0.50	64.51 ± 1.42	44.29 ± 0.17	131.94 ± 0.56
	ReconBoost	74.79 ± 1.04	63.20 ± 1.26	44.78 ± 0.29	130.88 ± 0.75
Audio, Vision, Text	Ensemble	72.62 ± 1.25	71.45 ± 1.00	40.40 ± 1.36	141.68 ± 1.32
	Uni-Pre Finetuned	75.65 ± 0.87	64.69 ± 1.10	44.65 ± 0.37	131.51 ± 0.63
	Late Concatenation	74.87 ± 1.23	63.05 ± 1.65	44.46 ± 0.41	131.87 ± 0.60
	ReconBoost	75.09 ± 0.88	63.18 ± 1.30	44.42 ± 0.53	130.79 ± 0.76

Table 6.10. *Performance comparison of proposed methods and baselines across different modality combinations (Audio-Vision, Text-Vision, and Audio-Vision-Text) on the CMU-MOSI and CMU-MOSEI datasets under standard training configurations. Methods include Ensemble, Uni-Pre Finetuned, Late Concatenation (Baseline), OGM, OGM-GE, ACC, AGM, PMR, and ReconBoost. Results are reported in terms of Accuracy and Loss. The table highlights the best-performing models for each metric (Accuracy and Loss) using a color-coded scheme. Dark shades represent the best performance, medium shades indicate the second-best performance, and light shades show the third-best performance.*

The experimental findings of Table 6.10 highlight Late Concatenation with joint optimization as the most effective audio-video model on the CMU-MOSI dataset. None of the examined methods surpass its performance in terms of accuracy or loss reduction. In contrast, dynamic gradient adjustment methods (OGM, AGM) achieve higher accuracy scores compared to other models; however, they introduce greater variability, leading to increased loss. The CMU-MOSEI dataset exhibits more stable learning dynamics, with

AGM achieving the best overall performance. While ReconBoost attains higher accuracy, it also experiences greater loss, suggesting a trade-off between these metrics. OGM remains competitive but does not significantly outperform standard baselines, including Ensemble, Uni-modal Pre-Finetuned models and Late Concatenation.

The Uni-Pre Finetuned Text-Video model achieves the best trade-off between accuracy and loss across both datasets. PMR successfully surpasses the Late Concatenation baseline across both datasets, while OGM on CMU-MOSI and ReconBoost on CMU-MOSEI demonstrate promising performance, achieving higher accuracy scores and reduced losses compared to the baseline concatenation model. AGM exhibits similar results for CMU-MOSEI, suggesting that the dynamic optimization methods examined in this thesis can be more effective when a dominant modality is present. However, Uni-Pre Finetuned remains the strongest model, raising important questions about the limitations of the proposed methods in surpassing its performance and the role of pre-training in sentiment classification. While our analysis confirms that the optimization techniques influence loss trends as expected and improve the performance of the text-video model, their inability to consistently improve overall performance suggests that they do not fully resolve the underlying challenges of modality imbalance.

Observing the trimodal models, we find that the Uni-Pre Finetuned model remains the strongest, with performance closely matching that of the Text-Video model. The results indicate that adding a third modality has minimal impact on performance, particularly in CMU-MOSEI, where bimodal text-video and trimodal accuracy remain nearly identical. This observation is consistent with previous findings [133], that also reported text dominates learning in CMU-MOSEI 7-class sentiment classification, with additional modalities contributing minimally. In CMU-MOSI, trimodal accuracy shows a slight improvement, but the negligible difference in loss suggests that the third modality (audio) contributes little to overall learning. These findings reinforce the idea that text is the dominant modality. The inclusion of additional modalities does not necessarily enhance performance, especially when they provide weakly informative features or fail to contribute effectively due to constraints imposed by the dominant modality.

Results by [39] confirm our observation that the examined methods fail to universally enhance performance across tasks. While the authors utilized transformer-based encoders and a sentiment regression framework instead of classification, their findings similarly highlight the limitations of these optimization techniques. Other studies [40] [41] question the universal effectiveness of the OGM-GE, AGM, and PMR methods under different levels of modality imbalance. This raises concerns regarding the generalization and robustness of these methods across diverse modeling paradigms in multimodal sentiment analysis, particularly when dealing with varying degrees of modality balance.

6.7 Summary of Findings

This section summarizes the key results obtained from the experiments and evaluates their implications for multimodal optimization. The Uni-Pre Finetuned model consistently achieves the best results in both the tri-modal and text-vision models, highlighting

its effectiveness in leveraging pre-trained knowledge for fine-tuned tasks. The strong performance of text-vision models suggests that the audio modality may not consistently contribute additional useful information, emphasizing the critical role of modality configurations in influencing the outcomes of multimodal learning and the dominant role of text in the datasets. This finding underscores the importance of carefully selecting and analyzing modality combinations in experiments.

The optimization methods examined in this research demonstrate potential in addressing unbalanced learning by improving weaker modalities. For instance, ReconBoost demonstrated their ability to amplify contributions from weaker modalities while controlling dominant ones. However, the inability of any single method to consistently outperform the baseline joint concatenation model across all configurations reveals a significant limitation. This suggests that while these methods influence the training process and benefit weaker modalities, they fail to achieve consistent, uniform improvements. Strong baselines such as Uni-Pre Finetuned and Late Concatenation with joint optimization not only achieve comparable results but often outperform the proposed methods.

A deeper analysis of these methods reveals certain trends and challenges. The inclusion of a development set for auxiliary calculations of modality discrepancy (OGM-GE), strength (AGM) and imbalanced (PMR) ratios proves beneficial. By using the development set for computing modality ratios and gradient coefficients, the methods become independent of batch size, preventing the models from being biased by sample distributions. This allows the models to rely entirely on learned features for their behavior, thereby examining their robustness.

Adam emerges as the more effective optimizer in terms of accuracy scores and loss trends, primarily due to its dynamic adaptation mechanisms, which enable more stable learning, faster convergence, and consistent performance across different training configurations. In contrast, SGD, with its static learning rate, exhibits slower convergence and greater sensitivity to hyperparameter selection, making it less robust in the multimodal training setups examined in our experiments.

However, many methods require prolonged training durations to exploit their full potential, which often leads to reduced performance compared to baselines. This suggests that extended training may not be sufficient or effective in bridging the performance gap. Extended modulation periods proved ineffective, as the models successfully converged within the initial epochs, rendering additional modulation unnecessary and redundant. Additionally, the sensitivity of these methods to hyperparameter tuning raises questions about their practicality and generalization in real-world scenarios.

In summary, the findings highlight the strengths and limitations of dynamic optimization methods in addressing multimodal learning challenges. While these methods show promise in improving weaker modalities and influencing training dynamics, their inconsistent performance across different configurations and their inability to surpass robust baselines underscore the complexities of multimodal optimization. These results reinforce the importance of dataset characteristics and modality-specific optimization, emphasizing the need for further refinement of multimodal learning techniques to balance accuracy, stability, and generalization.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Multimodal learning has emerged as a powerful approach in various machine learning applications, including sentiment analysis, where combining text, audio, and visual modalities can provide with nuanced, comprehensive information, enhancing accuracy. However, multimodal models often fail to outperform their unimodal counterparts, an issue primarily attributed to unbalanced multimodal learning.

This thesis investigated optimization techniques aimed at addressing this challenge, focusing on gradient-based methods (OGM-GE, AGM) and loss-based strategies (PMR, ReconBoost). These approaches were evaluated on the CMU-MOSI and CMU-MOSEI datasets to analyze their effectiveness in balancing modality learning dynamics and improving overall performance. In addition, we examined critical training configurations, including batch size selection, optimizer impact, and extended modulation periods, to assess how they influence model convergence and stability.

This research provides with an empirical analysis on multimodal optimization strategies. Gradient-based and multi-loss methods were systematically tested for their ability to address modality imbalance. Results indicated that while these methods improved learning stability, they did not universally outperform established baselines, such as Late Concatenation and Uni-Pre Finetuned models. The study also explored the role of batch size and proposed the incorporation of a development set to accurately quantify modality contributions in each method, minimizing bias and dependency on the selected batch size. This approach ensures that models can apply the proposed optimization techniques without requiring batch size adjustments to align with the capabilities of the algorithms, thereby enhancing flexibility in multimodal learning. Our study highlighted the impact of optimizer preference with Adam as the superior optimizer, facilitating faster and more stable convergence compared to SGD. Extending the modulation period beyond the initial training epochs proved ineffective, as models were capable of converging early. This suggests that dynamic optimization methods should examine whether they benefit from early-stage adjustments or prolonged intervention. The findings reinforced that text remains the dominant modality in multimodal sentiment analysis, with audio and video adding only marginal performance improvements even with the applied optimization techniques. This highlights the need for more modality-aware fusion techniques that better

leverage secondary modalities rather than treating them equally.

The findings of this research contribute to a deeper understanding of multimodal learning dynamics, particularly in sentiment analysis. While multimodal approaches are often assumed to be superior to unimodal models, our study confirms that modality imbalance can hinder performance. Furthermore, our analysis of training configurations like optimizer choice and incorporation of development set provides useful guidelines for future multimodal model training, particularly in balancing convergence speed and generalization performance.

This study provides valuable insights into multimodal optimization, yet certain limitations must be acknowledged. First, the experiments were conducted on CMU-MOSI and CMU-MOSEI, which, despite being well-established and widely used for sentiment analysis, may not fully generalize to other multimodal applications such as medical diagnosis, human activity recognition, or audiovisual speech processing. The effectiveness of the proposed methods should be further validated on datasets with different modality interactions. Second, our work primarily relied on late concatenation fusion with LSTM-based encoders, which, while effective, does not explore more advanced fusion strategies like transformer-based architectures. Third, the study found that extending modulation periods beyond the early training phases did not improve performance. Longer modulation durations may prove beneficial in settings where modality dominance shifts over time or when training on continually evolving multimodal data.

7.2 Future Work

Given the findings and limitations discussed, several directions for future research emerge.

- Future research should focus on developing effective methods for quantifying modality contributions to establish rules for optimizing each modality separately. A structured optimization approach integrating real-time modality contribution assessments and ensuring that each modality is adjusted based on its actual impact on learning has proved already a promising concept. Meta-learning techniques could further enhance this by dynamically determining when and how to adjust modality-specific learning rates.

- Alongside with optimization methods, future work should explore how to extract richer information from secondary modalities, potentially through cross-modal attention mechanisms or modality-specific feature transformations that amplify their unique contributions.

- Additionally, further research should explore whether alternative fusion methods yield greater benefits when combined with adaptive optimization techniques, assessing their effectiveness across different fusion strategies.

- Future research should evaluate whether a one-size-fits-all optimization strategy is feasible or if tailored approaches are necessary for different datasets and tasks. Ex-

panding the scope to include a broader range of sentiment analysis datasets would provide deeper insights into generalization and scalability of adaptive optimization methods, as training configurations that perform well for example on CMU-MOSI may not generalize effectively to CMU-MOSEI or other multimodal benchmarks.

- Future research could explore hybrid approaches that integrate unimodal pretraining with adaptive multimodal optimization, as the consistently strong performance of the Uni-Pre Finetuned model supports the effectiveness of this strategy.
- Future work should explore optimization techniques that minimize the need for extensive hyperparameter tuning, ensuring more practical and scalable multimodal learning frameworks.

The goal is to develop methods that adapt dynamically to modality-specific learning dynamics to reduce the dependency on dataset-specific tuning while maintaining robust performance across different architectures. This thesis contributes to the ongoing effort of improving multimodal neural network optimization by analyzing gradient-based and loss-based methods under various training configurations for sentiment analysis. While no single approach fully resolves modality imbalance, our findings provide a strong foundation for future research. The insights presented in this thesis aim to inspire continued exploration of the multimodal optimization challenge.

Bibliography

[1] J. Ebner, "Regression vs Classification, Explained - Sharp Sight," Apr. 2021. [Online]. Available: https://www.sharpsightlabs.com/blog/regression-vs-classification/

[2] Ralph, "Neural Network Building Blocks," Jul. 2019. [Online]. Available: https://medium.com/@rdugue1/neural-network-building-blocks-7ea6f8c790bf

[3] J. P. Davim, Ed., *Machining of Hard Materials*. London: Springer London, 2011. [Online]. Available: http://link.springer.com/10.1007/978-1-84996-450-0

[4] M. Vakalopoulou, S. Christodoulidis, N. Burgos, O. Colliot, and V. Lepetit, "Deep Learning: Basics and Convolutional Neural Networks (CNNs)," in *Machine Learning for Brain Disorders*, O. Colliot, Ed. New York, NY: Springer US, 2023, vol. 197, pp. 77–115. [Online]. Available: https://link.springer.com/10.1007/978-1-0716-3195-9_3

[5] D. Kalita, "What is Recurrent Neural Networks (RNN)?" Mar. 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/

[6] I. R. Jenkins, L. O. Gee, A. Knauss, H. Yin, and J. Schroeder, "Accident Scenario Generation with Recurrent Neural Networks," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 3340–3345, iSSN: 2153-0017. [Online]. Available: https://ieeexplore.ieee.org/document/8569661

[7] A. Graves, "Long Short-Term Memory," in *Supervised Sequence Labelling with Recurrent Neural Networks*, A. Graves, Ed. Berlin, Heidelberg: Springer, 2012, pp. 37–45. [Online]. Available: https://doi.org/10.1007/978-3-642-24797-2_4

[8] "Feed-forward vs feedback neural networks | DigitalOcean." [Online]. Available: https://www.digitalocean.com/community/tutorials/feed-forward-vs-feedback-neural-networks

[9] "3.6. Generalization — Dive into Deep Learning 1.0.3 documentation." [Online]. Available: https://d2l.ai/chapter_linear-regression/generalization.html#generalization

[10] P. Huilgol, "Bias and Variance in Machine Learning - A Fantastic Guide for Beginners!" Aug. 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/08/bias-and-variance-tradeoff-machine-learning/

[11] M. S. Minhas, "Techniques for handling underfitting and overfitting in Machine Learning," Jun. 2021. [Online]. Available: https://towardsdatascience.com/techniques-for-handling-underfitting-and-overfitting-in-machine-learning-348daa2380b9/

[12] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-Aware Minimization for Efficiently Improving Generalization," Apr. 2021, arXiv:2010.01412. [Online]. Available: http://arxiv.org/abs/2010.01412

[13] N. V. Otten, "Multimodal Natural Language Processing (NLP): The Next Powerful Shift In AI," Dec. 2023. [Online]. Available: https://spotintelligence.com/2023/12/19/multimodal-nlp-ai/

[14] P. P. Liang, A. Zadeh, and L.-P. Morency, "Foundations and Trends in Multimodal Machine Learning: Principles, Challenges, and Open Questions," Feb. 2023, arXiv:2209.03430. [Online]. Available: http://arxiv.org/abs/2209.03430

[15] M. Tavakoli, R. Chandra, F. Tian, and C. Bravo, "Multi-Modal Deep Learning for Credit Rating Prediction Using Text and Numerical Data Streams," Nov. 2024, arXiv:2304.10740. [Online]. Available: http://arxiv.org/abs/2304.10740

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Aug. 2023, arXiv:1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762

[17] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal Transformer for Unaligned Multimodal Language Sequences," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 6558–6569. [Online]. Available: https://aclanthology.org/P19-1656/

[18] E. Georgiou, C. Papaioannou, and A. Potamianos, "Deep Hierarchical Fusion with Application in Sentiment Analysis," in *Interspeech 2019*. ISCA, Sep. 2019, pp. 1646–1650. [Online]. Available: https://www.isca-archive.org/interspeech_2019/georgiou19_interspeech.html

[19] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations," Jun. 2019, arXiv:1810.02508. [Online]. Available: http://arxiv.org/abs/1810.02508

[20] Y. Fan, W. Xu, H. Wang, J. Wang, and S. Guo, "PMR: Prototypical Modal Rebalance for Multimodal Learning," Nov. 2022, arXiv:2211.07089. [Online]. Available: http://arxiv.org/abs/2211.07089

[21] C. Hua, Q. Xu, S. Bao, Z. Yang, and Q. Huang, "ReconBoost: Boosting Can Achieve Modality Reconcilement," May 2024, arXiv:2405.09321. [Online]. Available: http://arxiv.org/abs/2405.09321

[22] W. Wang, D. Tran, and M. Feiszli, "What Makes Training Multi-Modal Classification Networks Hard?" Apr. 2020, arXiv:1905.12681. [Online]. Available: http://arxiv.org/abs/1905.12681

[23] N. Wu, S. Jastrzębski, K. Cho, and K. J. Geras, "Characterizing and overcoming the greedy nature of learning in multi-modal deep neural networks," Sep. 2022, arXiv:2202.05306. [Online]. Available: http://arxiv.org/abs/2202.05306

[24] R. Panda, C.-F. Chen, Q. Fan, X. Sun, K. Saenko, A. Oliva, and R. Feris, "AdaMML: Adaptive Multi-Modal Learning for Efficient Video Recognition," May 2021, arXiv:2105.05165. [Online]. Available: http://arxiv.org/abs/2105.05165

[25] X. Zhang, J. Yoon, M. Bansal, and H. Yao, "Multimodal Representation Learning by Alternating Unimodal Adaptation," Apr. 2024, arXiv:2311.10707. [Online]. Available: http://arxiv.org/abs/2311.10707

[26] X. Peng, Y. Wei, A. Deng, D. Wang, and D. Hu, "Balanced Multimodal Learning via On-the-fly Gradient Modulation," Mar. 2022, arXiv:2203.15332. [Online]. Available: http://arxiv.org/abs/2203.15332

[27] H. Li, X. Li, P. Hu, Y. Lei, C. Li, and Y. Zhou, "Boosting Multi-modal Model Performance with Adaptive Gradient Modulation," Aug. 2023, arXiv:2308.07686. [Online]. Available: http://arxiv.org/abs/2308.07686

[28] R. Rojas, "The Backpropagation Algorithm," in *Neural Networks: A Systematic Introduction*, R. Rojas, Ed. Berlin, Heidelberg: Springer, 1996, pp. 149–182. [Online]. Available: https://doi.org/10.1007/978-3-642-61068-4_7

[29] Y. Huang, J. Lin, C. Zhou, H. Yang, and L. Huang, "Modality Competition: What Makes Joint Training of Multi-modal Network Fail in Deep Learning? (Provably)," Mar. 2022, arXiv:2203.12221. [Online]. Available: http://arxiv.org/abs/2203.12221

[30] J. Wu, Y. Liang, F. Han, H. Akbari, Z. Wang, and C. Yu, "Scaling Multimodal Pre-Training via Cross-Modality Gradient Harmonization," Nov. 2022, arXiv:2211.02077. [Online]. Available: http://arxiv.org/abs/2211.02077

[31] Z. Guo, T. Jin, J. Chen, and Z. Zhao, "Classifier-guided Gradient Modulation for Enhanced Multimodal Learning," Nov. 2024, arXiv:2411.01409. [Online]. Available: http://arxiv.org/abs/2411.01409

[32] Y. Wei and D. Hu, "MMPareto: Boosting Multimodal Learning with Innocent Unimodal Assistance," May 2024, arXiv:2405.17730. [Online]. Available: http://arxiv.org/abs/2405.17730

[33] M. R. Zhang, J. Lucas, G. Hinton, and J. Ba, "Lookahead Optimizer: k steps forward, 1 step back," Dec. 2019, arXiv:1907.08610. [Online]. Available: http://arxiv.org/abs/1907.08610

[34] S. Alfasly, J. Lu, C. Xu, and Y. Zou, "Learnable Irrelevant Modality Dropout for Multimodal Action Recognition on Modality-Specific Annotated Videos," Mar. 2022, arXiv:2203.03014. [Online]. Available: http://arxiv.org/abs/2203.03014

[35] S. Liu, P. Gao, Y. Li, W. Fu, and W. Ding, "Multi-modal fusion network with complementarity and importance for emotion recognition," *Information Sciences*, vol. 619, pp. 679–694, Jan. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025522013652

[36] Y. He, R. Cheng, G. Balasubramaniam, Y.-H. H. Tsai, and H. Zhao, "Efficient Modality Selection in Multimodal Learning," *Journal of Machine Learning Research*, vol. 25, no. 47, pp. 1–39, 2024. [Online]. Available: http://jmlr.org/papers/v25/23-0439.html

[37] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency, "MOSI: Multimodal Corpus of Sentiment Intensity and Subjectivity Analysis in Online Opinion Videos," *arXiv preprint*, 2016. [Online]. Available: https://arxiv.org/abs/1606.06259

[38] A. Bagher Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency, "Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2236–2246. [Online]. Available: https://aclanthology.org/P18-1208/

[39] K. Kontras, T. Strypsteen, C. Chatzichristos, P. P. Liang, M. Blaschko, and M. D. Vos, "Multimodal Fusion Balancing Through Game-Theoretic Regularization," Dec. 2024, arXiv:2411.07335. [Online]. Available: http://arxiv.org/abs/2411.07335

[40] Y. Wei, R. Feng, Z. Wang, and D. Hu, "Enhancing Multimodal Cooperation via Sample-Level Modality Valuation," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2024, pp. 27 328–27 337. [Online]. Available: https://ieeexplore.ieee.org/document/10656817/

[41] Y. Wei, S. Li, R. Feng, and D. Hu, "Diagnosing and Re-learning for Balanced Multimodal Learning," Jul. 2024, arXiv:2407.09705. [Online]. Available: http://arxiv.org/abs/2407.09705

[42] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017, arXiv:1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," May 2019, arXiv:1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805

[44] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap

and Sharp Minima," Feb. 2017, arXiv:1609.04836. [Online]. Available: http://arxiv.org/abs/1609.04836

[45] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic Gradient Descent as Approximate Bayesian Inference," Jan. 2018, arXiv:1704.04289. [Online]. Available: http://arxiv.org/abs/1704.04289

[46] S. L. Smith, E. Elsen, and S. De, "On the Generalization Benefit of Noise in Stochastic Gradient Descent," Jun. 2020, arXiv:2006.15081. [Online]. Available: http://arxiv.org/abs/2006.15081

[47] W. Yu, H. Xu, Z. Yuan, and J. Wu, "Learning Modality-Specific Representations with Self-Supervised Multi-Task Learning for Multimodal Sentiment Analysis," Feb. 2021, arXiv:2102.04830. [Online]. Available: http://arxiv.org/abs/2102.04830

[48] Z. Wu, Z. Gong, J. Koo, and J. Hirschberg, "Multimodal Multi-loss Fusion Network for Sentiment Analysis," Jun. 2024, arXiv:2308.00264. [Online]. Available: http://arxiv.org/abs/2308.00264

[49] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, "Entropy-SGD: Biasing Gradient Descent Into Wide Valleys," Apr. 2017, arXiv:1611.01838. [Online]. Available: http://arxiv.org/abs/1611.01838

[50] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The Marginal Value of Adaptive Gradient Methods in Machine Learning," May 2018, arXiv:1705.08292. [Online]. Available: http://arxiv.org/abs/1705.08292

[51] L. Prechelt, "Early Stopping - But When?" in *Neural Networks: Tricks of the Trade*, G. Goos, J. Hartmanis, J. Van Leeuwen, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, vol. 1524, pp. 55–69. [Online]. Available: http://link.springer.com/10.1007/3-540-49430-8_3

[52] L. N. Smith and N. Topin, "Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates," May 2018, arXiv:1708.07120. [Online]. Available: http://arxiv.org/abs/1708.07120

[53] T. M. Mitchell, "Machine Learning textbook." [Online]. Available: https://www.cs.cmu.edu/~tom/mlbook.html

[54] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[55] P. Cunningham, M. Cord, and S. J. Delany, "Supervised Learning," in *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, M. Cord and P. Cunningham, Eds. Berlin, Heidelberg: Springer, 2008, pp. 21–49. [Online]. Available: https://doi.org/10.1007/978-3-540-75171-7_2

[56] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, "Unsupervised Learning," in *An Introduction to Statistical Learning: with Applications in Python*, G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, Eds. Cham: Springer International Publishing, 2023, pp. 503–556. [Online]. Available: https://doi.org/10.1007/978-3-031-38747-0_12

[57] Y. Li, "Deep Reinforcement Learning: An Overview," Nov. 2018, arXiv:1701.07274. [Online]. Available: http://arxiv.org/abs/1701.07274

[58] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992. [Online]. Available: https://doi.org/10.1007/BF00992698

[59] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986. [Online]. Available: https://www.nature.com/articles/323533a0

[60] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, J. Fürnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 807–814. [Online]. Available: https://icml.cc/Conferences/2010/papers/432.pdf

[61] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, Eds. Berlin, Heidelberg: Springer, 1998, pp. 9–50. [Online]. Available: https://doi.org/10.1007/3-540-49430-8_2

[62] J. S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," in *NATO Neurocomputing*, 1989. [Online]. Available: https://api.semanticscholar.org/CorpusID:59636530

[63] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. [Online]. Available: https://doi.apa.org/doi/10.1037/h0042519

[64] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: https://www.nature.com/articles/nature14539

[65] R. M. Schmidt, "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview," Nov. 2019, arXiv:1912.05911. [Online]. Available: http://arxiv.org/abs/1912.05911

[66] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.

[67] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, "A Comprehensive Survey of Loss Functions in Machine Learning," *Annals of Data Science*, vol. 9, no. 2, pp. 187–212, Apr. 2022. [Online]. Available: https://doi.org/10.1007/s40745-020-00253-5

[68] D. R. Cox, "The Regression Analysis of Binary Sequences," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 21, no. 1, pp. 238–238, Jan. 1959. [Online]. Available: https://academic.oup.com/jrsssb/article/21/1/238/7035243

[69] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul. 1997. [Online]. Available: https://doi.org/10.1023/A:1007379606734

[70] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994. [Online]. Available: https://ieeexplore.ieee.org/document/279181

[71] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks," Feb. 2013, arXiv:1211.5063. [Online]. Available: http://arxiv.org/abs/1211.5063

[72] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html

[73] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1026–1034, iSSN: 2380-7504. [Online]. Available: https://ieeexplore.ieee.org/document/7410480

[74] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Mar. 2015, arXiv:1502.03167. [Online]. Available: http://arxiv.org/abs/1502.03167

[75] S. Hochreiter, "Recurrent neural net learning and vanishing gradient," *International Journal Of Uncertainity, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107–116, 1998.

[76] T. L. Lai, "Stochastic approximation: invited paper," *The Annals of Statistics*, vol. 31, no. 2, Apr. 2003. [Online]. Available: https://projecteuclid.org/journals/annals-of-statistics/volume-31/issue-2/Stochastic-approximation-invited-paper/10.1214/aos/1051027873.full

[77] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp.

1–17, Jan. 1964. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0041555364901375

[78] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011. [Online]. Available: https://dl.acm.org/doi/10.5555/1953048.2021068

[79] S. Geman, E. Bienenstock, and R. Doursat, "Neural Networks and the Bias/Variance Dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, Jan. 1992. [Online]. Available: https://direct.mit.edu/neco/article/4/1/1-58/5624

[80] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, no. none, Jan. 2010. [Online]. Available: https://projecteuclid.org/journals/statistics-surveys/volume-4/issue-none/A-survey-of-cross-validation-procedures-for-model-selection/10.1214/09-SS054.full

[81] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*, ser. ICML '04. New York, NY, USA: Association for Computing Machinery, Jul. 2004, p. 78. [Online]. Available: https://dl.acm.org/doi/10.1145/1015330.1015435

[82] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jun. 2014.

[83] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, Jul. 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0

[84] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the Loss Landscape of Neural Nets," Nov. 2018, arXiv:1712.09913. [Online]. Available: http://arxiv.org/abs/1712.09913

[85] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," Aug. 2017, arXiv:1705.09406. [Online]. Available: http://arxiv.org/abs/1705.09406

[86] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA: IEEE, Jun. 2014, pp. 1725–1732. [Online]. Available: https://ieeexplore.ieee.org/document/6909619

[87] T. Siebert, K. N. Clasen, M. Ravanbakhsh, and B. Demir, "Multi-Modal Fusion Transformer for Visual Question Answering in Remote Sensing," Oct. 2022, arXiv:2210.04510. [Online]. Available: http://arxiv.org/abs/2210.04510

[88] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: A Neural Image Caption Generator," Apr. 2015, arXiv:1411.4555. [Online]. Available: http://arxiv.org/abs/1411.4555

[89] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding," Sep. 2016, arXiv:1606.01847. [Online]. Available: http://arxiv.org/abs/1606.01847

[90] M. S. Saeed, M. H. Khan, S. Nawaz, M. H. Yousaf, and A. D. Bue, "Fusion and Orthogonal Projection for Improved Face-Voice Association," Dec. 2021, arXiv:2112.10483. [Online]. Available: http://arxiv.org/abs/2112.10483

[91] M. S. Saeed, S. Nawaz, M. H. Khan, M. Z. Zaheer, K. Nandakumar, M. H. Yousaf, and A. Mahmood, "Single-branch Network for Multimodal Training," Mar. 2023, arXiv:2303.06129. [Online]. Available: http://arxiv.org/abs/2303.06129

[92] M. S. Saeed, S. Nawaz, M. H. Khan, S. Javed, M. H. Yousaf, and A. D. Bue, "Learning Branched Fusion and Orthogonal Projection for Face-Voice Association," Aug. 2022, arXiv:2208.10238. [Online]. Available: http://arxiv.org/abs/2208.10238

[93] X. Wang, L. Zhu, Z. Zheng, M. Xu, and Y. Yang, "Align and Tell: Boosting Text-Video Retrieval With Local Alignment and Fine-Grained Supervision," *IEEE Transactions on Multimedia*, vol. 25, pp. 6079–6089, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9878037/

[94] Z. Feng, Z. Zeng, C. Guo, and Z. Li, "Temporal Multimodal Graph Transformer With Global-Local Alignment for Video-Text Retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 3, pp. 1438–1453, Mar. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9895256/

[95] C. Arnold and A. Küpfer, "Alignment Helps Make the Most of Multimodal Data," Jul. 2024, arXiv:2405.08454. [Online]. Available: http://arxiv.org/abs/2405.08454

[96] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Batra, and D. Parikh, "VQA: Visual Question Answering," Oct. 2016, arXiv:1505.00468. [Online]. Available: http://arxiv.org/abs/1505.00468

[97] J. Li, R. R. Selvaraju, A. D. Gotmare, S. Joty, C. Xiong, and S. Hoi, "Align before Fuse: Vision and Language Representation Learning with Momentum Distillation," Oct. 2021, arXiv:2107.07651. [Online]. Available: http://arxiv.org/abs/2107.07651

[98] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal Deep Learning," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, L. Getoor and T. Scheffer, Eds. Omnipress, 2011, pp. 689–696. [Online]. Available: https://icml.cc/2011/papers/399_icmlpaper.pdf

[99] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in neural information processing systems*, 2012, pp. 2222–2230.

[100] M. Wöllmer, M. Kaiser, F. Eyben, B. Schuller, and G. Rigoll, "LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework," *Image and Vision Computing*, vol. 31, no. 2, pp. 153–163, Feb. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0262885612000285

[101] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia Systems*, vol. 16, no. 6, pp. 345–379, Nov. 2010. [Online]. Available: http://link.springer.com/10.1007/s00530-010-0182-0

[102] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "A review of affective computing: From unimodal analysis to multimodal fusion," *Information Fusion*, vol. 37, pp. 98–125, Sep. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1566253517300738

[103] W. Guo, J. Wang, and S. Wang, "Deep Multimodal Representation Learning: A Survey," *IEEE Access*, vol. 7, pp. 63 373–63 394, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8715409/

[104] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical Question-Image Co-Attention for Visual Question Answering," Jan. 2017, arXiv:1606.00061. [Online]. Available: http://arxiv.org/abs/1606.00061

[105] M. Wöllmer, A. Metallinou, F. Eyben, B. Schuller, and S. S. Narayanan, "Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional LSTM modeling," in *Interspeech 2010*. ISCA, Sep. 2010, pp. 2362–2365. [Online]. Available: https://www.isca-archive.org/interspeech_2010/wollmer10c_interspeech.html

[106] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: a Visual Language Model for Few-Shot Learning," Nov. 2022, arXiv:2204.14198. [Online]. Available: http://arxiv.org/abs/2204.14198

[107] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks," Aug. 2019, arXiv:1908.02265. [Online]. Available: http://arxiv.org/abs/1908.02265

[108] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "UNITER: UNiversal Image-TExt Representation Learning," Jul. 2020, arXiv:1909.11740. [Online]. Available: http://arxiv.org/abs/1909.11740

[109] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," Jul. 2017, arXiv:1703.03400. [Online]. Available: http://arxiv.org/abs/1703.03400

[110] Y.-H. H. Tsai, P. P. Liang, A. Zadeh, L.-P. Morency, and R. Salakhutdinov, "Learning Factorized Multimodal Representations," May 2019, arXiv:1806.06176. [Online]. Available: http://arxiv.org/abs/1806.06176

[111] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008. [Online]. Available: http://www.nowpublishers.com/article/Details/INR-011

[112] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. Association for Computational Linguistics, Jul. 2002, pp. 79–86. [Online]. Available: https://aclanthology.org/W02-1011/

[113] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," Oct. 2013, arXiv:1310.4546. [Online]. Available: http://arxiv.org/abs/1310.4546

[114] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162/

[115] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. [Online]. Available: https://aclanthology.org/D14-1181/

[116] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: https://direct.mit.edu/neco/article/9/8/1735-1780/6109

[117] P. Ekman, "An argument for basic emotions," *Cognition and Emotion*, vol. 6, no. 3-4, pp. 169–200, May 1992. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/02699939208411068

[118] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," Nov. 2018, arXiv:1708.02709. [Online]. Available: http://arxiv.org/abs/1708.02709

[119] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[120] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4. Montreal, Que., Canada: IEEE, 2005, pp. 2047–2052. [Online]. Available: http://ieeexplore.ieee.org/document/1556215/

[121] A. Zadeh, P. P. Liang, N. Mazumder, S. Poria, E. Cambria, and L.-P. Morency, "Memory Fusion Network for Multi-view Sequential Learning," Feb. 2018, arXiv:1802.00927. [Online]. Available: http://arxiv.org/abs/1802.00927

[122] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency, "Tensor Fusion Network for Multimodal Sentiment Analysis," Jul. 2017, arXiv:1707.07250. [Online]. Available: http://arxiv.org/abs/1707.07250

[123] A. Bagher Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency, "Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2236–2246. [Online]. Available: https://aclanthology.org/P18-1208

[124] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient Surgery for Multi-Task Learning," Dec. 2020, arXiv:2001.06782. [Online]. Available: http://arxiv.org/abs/2001.06782

[125] Y. Zhang, P. E. Latham, and A. Saxe, "Understanding Unimodal Bias in Multimodal Deep Linear Networks," Jun. 2024, arXiv:2312.00935. [Online]. Available: http://arxiv.org/abs/2312.00935

[126] R. Xu, R. Feng, S.-X. Zhang, and D. Hu, "MMCosine: Multi-Modal Cosine Loss Towards Balanced Audio-Visual Fine-Grained Learning," Mar. 2023, arXiv:2303.05338. [Online]. Available: http://arxiv.org/abs/2303.05338

[127] P. Ngatchou, A. Zarei, and A. El-Sharkawi, "Pareto Multi Objective Optimization," in *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*. Arlington, Virginia, USA: IEEE, 2005, pp. 84–91. [Online]. Available: http://ieeexplore.ieee.org/document/1599245/

[128] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951. [Online]. Available: http://projecteuclid.org/euclid.aoms/1177729694

[129] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *International Conference on Machine Learning*, 1996. [Online]. Available: https://api.semanticscholar.org/CorpusID:1836349

[130] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *The Annals of Statistics*, vol. 29, no. 5, Oct.

2001. [Online]. Available: https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full

[131] G. Degottex, J. Kane, T. Drugman, T. Raitio, and S. Scherer, "COVAREP — A collaborative voice analysis repository for speech technologies," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Florence, Italy: IEEE, May 2014, pp. 960–964. [Online]. Available: http://ieeexplore.ieee.org/document/6853739/

[132] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "OpenFace 2.0: Facial Behavior Analysis Toolkit," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. Xi'an: IEEE, May 2018, pp. 59–66. [Online]. Available: https://ieeexplore.ieee.org/document/8373812/

[133] J.-B. Delbrouck, N. Tits, M. Brousmiche, and S. Dupont, "A Transformer-based joint-encoding for Emotion Recognition and Sentiment Analysis," in *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*, A. Zadeh, L.-P. Morency, P. P. Liang, and S. Poria, Eds. Seattle, USA: Association for Computational Linguistics, Jul. 2020, pp. 1–7. [Online]. Available: https://aclanthology.org/2020.challengehml-1.1/

[134] P. Zhou, J. Feng, C. Ma, C. Xiong, S. Hoi, and W. E, "Towards Theoretically Understanding Why SGD Generalizes Better Than ADAM in Deep Learning," Nov. 2021, arXiv:2010.05627. [Online]. Available: http://arxiv.org/abs/2010.05627

List of Abbreviations

AdaMML	Adaptive Multi-Modal Learning
Adam	Adaptive Moment Estimation
AGM	Adaptive Gradient Modulation
AI	Artificial Intelligence
ANN	Artificial Neural Network
CLS	Conditional Learning Speed
CMU-MOSI	Carnegie Mellon University Multimodal Opinion Sentiment Intensity
CMU-MOSEI	Carnegie Mellon University Multimodal Opinion Sentiment and Emotion Intensity
CNN	Convolutional Neural Network
CUR	Conditional Utilization Rate
DHF	Deep Hierarchical Fusion
DFG	Dynamic Fusion Graph
DNN	Deep Neural Network
EMA	Exponential Moving Average
FNN	Feed-forward Neural Network
GRS	Global Rectification Scheme
IMD	Irrelevant Modality Dropout
KL-divergence	Kullback–Leibler divergence
LSTM	Long Short-Term Memory
MFCC	Mel-Frequency Cepstral Coefficient
MAE	Mean Absolute Error
MCR	Memory Consolidation Regularization
ML	Machine Learning
MLA	Multimodal Learning with Alternating Unimodal Adaptation
MLP	Multilayer Perceptron
MSE	Mean Squared Error
MulT	Multimodal Transformer
OGM	On-the-fly Gradient Modulation
OGM-GE	On-the-fly Gradient Modulation with Generalization Enhancement
PMR	Prototypical Modal Rebalance
RNN	Recurrent Neural Network
SAM	Sharpness-Aware Minimization
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TFN	Tensor Fusion Network

VQA	Visual Question Answering

Appendix

Appendix A

Hyperparameter Tuning Details

This appendix outlines the hyperparameter tuning strategies employed in the experiments discussed in the main body of this thesis. The general training configurations, including batch sizes, optimizers, and other fundamental parameters, are detailed in the main text. Here, we focus on the method-specific hyperparameter tuning, which was systematically refined to ensure fair comparisons and optimal performance across different approaches. These tuning ranges serve as a structured reference for future work, maintaining consistency in hyperparameter selection while allowing flexibility for further optimization.

A.0.1 On-the-Fly Gradient Modulation, On-the-fly Gradient Modulation with Generalization Enhancement, Acceleration of slow learning modality

- **Exploration Range for** a: The parameter a was varied within the range $[0.2, 0.7]$ in increments of 0.1, with the best performance typically observed around 0.5.
- **Modulation Duration**: For short-term experiments, modulation was applied for the first 5 epochs, while for extended evaluations, the first 50 epochs yielded optimal results.
- **Learning Rate Selection**: The learning rate was chosen from the set $\{5 \times 10^{-4}, 1 \times 10^{-4}, 8 \times 10^{-5}\}$ based on empirical performance for Adam optimizer. For the SGD optimizer, it was set to either 1×10^{-2} or 5×10^{-2} for CMU-MOSI and 5×10^{-3} for CMU-MOSEI.
- **Discrepancy Ratio Update Frequency**: This parameter was only considered in experiments where discrepancy ratio calculations were based on the development set. The update frequency was set to 1, 2, or 5, with 5 being chosen in our experiments.

A.0.2 Adaptive Gradient Modulation

- **Exploration Range for** a: The parameter a was set to 1 or 2, with the best performance typically observed when set to 2.
- **Modulation Duration**: Modulation was applied for the first 10 epochs.

- **Learning Rate Selection**: The learning rate was chosen based on empirical performance. For the Adam optimizer, it ranged from 5×10^{-3} to 1×10^{-4}. For the SGD optimizer, it was set to either 1×10^{-2} or 5×10^{-2} for CMU-MOSI and 5×10^{-3} for CMU-MOSEI.
- **Discrepancy Ratio Update Frequency**: This parameter was only considered in experiments where discrepancy ratio calculations were based on the development set. The update frequency was set to 1, 2, or 5, with 2 being chosen in our experiments.

A.0.3 Prototypical Modal Rebalance

- **Exploration Range for** a: The parameter a was varied within the range $[0.3, 1.0]$ in increments of 0.1, with the best performance typically observed around 0.3 for audio-video models and 0.5 for text-video models.
- **Modulation Duration**: We experimented with modulation for the first 5 or 10 epochs, while for extended evaluations, the first 50 epochs yielded optimal results.
- **Learning Rate Selection**:The learning rate was chosen from the set $\{5 \times 10^{-4}$, 1×10^{-4}, $8 \times 10^{-5}\}$ based on empirical performance for Adam optimizer. For the SGD optimizer, it was set to either 1×10^{-2} or 5×10^{-2} for CMU-MOSI and 5×10^{-3} for CMU-MOSEI.
- **Discrepancy Ratio Update Frequency**: This parameter was only considered in experiments where discrepancy ratio calculations were based on the development set. The update frequency was set to 1 or 5, with 5 being chosen in our experiments.
- **Exponential Moving Average (EMA)**: The smoothing factor ϵ was set to 0.1 or 0.3 to control the degree of weighting applied to past prototypes.

A.0.4 ReconBoost

- **Weight Parameter** a: Fixed at 0.5 for all experiments.
- **Alternating Stages**: Set to 100 in all cases.
- **Boosting Rate**: Fixed at 1 across all experimental settings.
- **Weight Factors**: Controlling the relative importance of the direct loss and residual loss, respectively, with $w_1 = 1$ and $w_2 = 0.25$.
- **One alternating-boosting stage** lasts for 1 epoch.
- **Global rectification stage** lasts for 1 epoch.
- **Learning Rates**: The learning rate for both the boosting scheme and the GRS scheme was set to 5×10^{-4} for the Adam optimizer and 1×10^{-2} for the SGD optimizer.