NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF SIGNALS, CONTROL AND ROBOTICS
SPEECH AND LANGUAGE PROCESSING GROUP

# Efficient Incomplete Multimodal-Diffused Emotion Recognition

## Diploma Thesis

of

**IOANNIS ASPROGERAKAS**

**Supervisor:** Alexandros Potamianos
Associate Professor, NTUA

Athens, October 2025

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF SIGNALS, CONTROL AND ROBOTICS
SPEECH AND LANGUAGE PROCESSING GROUP

# Efficient Incomplete Multimodal-Diffused Emotion Recognition

## Diploma Thesis

of

**IOANNIS ASPROGERAKAS**

**Supervisor:** Alexandros Potamianos
Associate Professor, NTUA

Approved by the examination committee on 24 October 2025.

| *(Signature)* | *(Signature)* | *(Signature)* |
|---|---|---|
| Alexandros Potamianos | Athanasios Rontogiannis | Athanasios Voulodimos |
| Associate Professor, NTUA | Associate Professor, NTUA | Assistant Professor, NTUA |

Athens, October 2025

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF SIGNALS, CONTROL AND ROBOTICS
SPEECH AND LANGUAGE PROCESSING GROUP



## 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)*

. . . . . .
Ioannis Asprogerakas,
Graduate of Electrical and
Computer Engineering,
NTUA
24 October 2025

# Περίληψη

Η Πολυτροπική Αναγνώριση Συναισθημάτων (Multimodal Emotion Recognition – MER) στοχεύει στη μοντελοποίηση των ανθρώπινων συναισθημάτων μέσω της ενοποίησης σημάτων από γλώσσα, όραση και ήχο. Οι μέθοδοι βαθιάς μάθησης έχουν επιτύχει εντυπωσιακά αποτελέσματα μέσω του cross-modal fusion, αλλά οι περισσότερες υποθέτουν πλήρη διαθεσιμότητα των τροπικοτήτων κατά την εκπαίδευση και inference, κάτι που δεν είναι συνήθης στην πράξη λόγω αποκλεισμών, θορύβου ή βλαβών αισθητήρων. Η αντιμετώπιση αυτού του προβλήματος απαιτεί εύρωστες στρατηγικές αναπλήρωσης που να ανακτούν τα ελλείποντα σήματα χωρίς να θυσιάσουν αποδοτικότητα.

Σε αυτή την εργασία εξερευνούμε τον σχεδιασμό μοντέλων διάχυσης για την αναπλήρωση ελλειπουσών τροπικοτήτων επεκτείνοντας την αρχιτεκτονική του IMDER [1]. Προτείνουμε ένα σχήμα εκπαίδευσης δύο σταδίων, όπου τα modality-specific μοντέλα διάχυσης προεκπαιδεύονται ανεξάρτητα και στη συνέχεια ενσωματώνονται στη διαδικασία MER. Επιπλέον, συγκρίνουμε διατυπώσεις στοχαστικών διαφορικών εξισώσεων (ΣΔΕ), συγκεκριμένα τις Variance Preserving (VP) και Variance Exploding (VE), αξιολογούμε εναλλακτικούς μηχανισμούς υπο συνθήκης παραγωγής με αρχιτεκτονικές transformer, και διερευνούμε αλγορίθμους δειγματοληψίας για να ισορροπήσουμε αποδοτικότητα και ακρίβεια.

Εκτενή πειράματα στα CMU-MOSI και CMU-MOSEI αποδεικνύουν συνεπείς βελτιώσεις τόσο σε σταθερά όσο και σε τυχαία πρωτόκολλα ελλείψεων. Η διαμόρφωσή μας που επικεντρώνεται στην ποιότητα επιτυγχάνει ανώτερη ακρίβεια, με κέρδη έως +2% F1 και +1.5% $ACC_2$ σε σχέση με το IMDER, παρέχοντας παράλληλα 5× ταχύτερο inference. Ταυτόχρονα, η διαμόρφωσή μας που εστιάζει στην ταχύτητα διατηρεί ανταγωνιστική απόδοση +1% $ACC_2$, +0.5% F1 αλλά επιτυγχάνει αξιοσημείωτη αποδοτικότητα με 15× ταχύτερο inference, καθιστώντας την ανταγωνιστική για εφαρμογές MER πραγματικού χρόνου.

## Λέξεις Κλειδιά

Μοντέλα Διάχυσης, Πολυτροπική Αναγνώριση Συναισθήματος, Στοχαστικές Διαφορικές Εξισώσεις, Πολυτροπική Βαθιά Μάθηση, Βαθιά Γεννετική Μοντελοποίηση

# Abstract

Multimodal Emotion Recognition (MER) aims to model human affect by integrating complementary signals from language, vision, and audio. While deep learning methods have achieved impressive results through cross-modal fusion, most assume complete modality availability during training and inference, a condition rarely met in real world deployments where occlusions, noise, or sensor failures frequently cause missing modalities. Addressing this problem requires robust imputation strategies that can recover missing signals without sacrificing efficiency.

In this work, we explore the design space of diffusion models for missing modality imputation, building upon and extending the IMDER [1] framework. We propose a decoupled two-stage training scheme where modality-specific diffusion models are pre-trained independently and then integrated into the MER pipeline. This design avoids the instability of end-to-end IMDER training, where untrained diffusion models initially degrade classifier performance. In addition, we systematically compare stochastic differential equation (SDE) formulations, specifically Variance Preserving (VP) and Variance Exploding (VE) processes, evaluate alternative conditioning mechanisms with transformer-based backbones, and finally investigate multiple sampling strategies to balance efficiency and accuracy.

Extensive experiments on CMU-MOSI and CMU-MOSEI demonstrate consistent improvements across both fixed and random missing protocols. Our quality-focused configuration achieves superior accuracy, with up to +2% F1 and +1.5% $ACC_2$ gains over IMDER, while delivering 5× faster inference. Meanwhile, our speed-optimized configuration maintains competitive performance, +1% $ACC_2$, +0.5% F1, but achieves remarkable efficiency with 15× faster inference, making it competitive for real-time MER applications.

## Keywords

Diffusion Models, Multimodal Emotion Recognition, Stochastic Differential Equations, Multimodal Deep Learning, Deep Generative Modeling

*to my parents*

# Acknowledgements

I would like to express my sincere gratitude to my parents Evangelos and Aggeliki for their constant love and support during my academic journey. Since I was a little kid remember my father constantly nurturing my curiosity and inspiring my passion for engineering with such dedication that it shaped the very course of my life. I would also want to extend my gratitude to my friends for their presence and encouragement during the difficulties I encountered and were there when I needed them. Another big thank you I would like to give to my collaborating researcher Efthymi Georgiou for his valuable guidance, fruitful ideas, excellent communication and great patience with me. Finally, I want to sincerely thank my supervisor Mr. Alexandros Potamianos for his invaluable guidance and insightful feedback throughout the preparation of this thesis.

Athens, October 2025

Ioannis Asprogerakas

# Contents

# List of Figures

# List of Tables

Κεφάλαιο 0

# Εκτεταμένη Ελληνική Περίληψη

## 0.1 Εισαγωγή

### 0.1.1 Κίνητρο

Η ενοποίηση πληροφορίας από πολλαπλές πηγές παρέχει συμπληρωματικά στοιχεία και αυξάνει την επίδοσης και την ανθεκτικότητα των συστημάτων σε σχέση με μονοτροπικές προσεγγίσεις [31], ιδιαίτερα σε απαιτητικές εφαρμογές όπως η αναγνώριση συναισθήματος. Ωστόσο, τα πολυτροπικά δεδομένα συχνά περιέχουν ελλιπείς ή κατεστραμμένες εγγραφές, καθιστώντας την εκπαίδευση και την ανάπτυξη σε πραγματικά σενάρια πιο περίπλοκη.

Η Αναγνώριση Συναισθήματος από πολλαπλές τροπικότητες σήματος (γλώσσα, οπτικά χαρακτηριστικά, ακουστική τροπικότητα) έχει σημειώσει σημαντική πρόοδο χάρη στις μεθόδους βαθιάς μάθησης και ιδιαίτερα τα μοντέλα διασταυρούμενης προσοχής cross-attention όπως τα MISA [32] και MulT [26]. Παρόλα αυτά, τα περισσότερα προϋποθέτουν πλήρη διαθεσιμότητα όλων των τροπικοτήτων (modalities), κάτι που σπανίζει σε πραγματικές συνθήκες. Για το πρόβλημα της απουσίας modalities έχουν προταθεί δύο βασικές κατευθύνσεις: (α) *μέθοδοι αναπλήρωσης* με autoencoders [33], GANs [34], normalizing flows [35] ή diffusion models όπως το IMDER [1], και (β) *μη-αναπληρωτικές μέθοδοι* (π.χ. canonical correlation analysis [27], knowledge distillation [36], subset grouping [37]), που αν και πιο αποδοτικές συχνά αγνοούν κρίσιμες συσχετίσεις.

### 0.1.2 Συνεισφορές

Η παρούσα εργασία εισάγει ένα **αποσυνδεδεμένο (decoupled) σχήμα εκπαίδευσης** για diffusion-based πολυτροπικής αναγνώρισης συναισθήματος (ΠΑΣ), το οποίο βελτιώνει την απόδοση και την αποδοτικότητα έναντι του IMDER [1]. Συγκεκριμένα:

1. Προτείνεται δισταδιακή εκπαίδευση, αρχικά των score networks και στη συνέχεια του

MER μοντέλου, αποφεύγοντας την εκκίνηση με 'θορυβώδη' ανακατασκευασμένα σήματα.

2. Εξετάζονται διαφορετικές διατυπώσεις diffusion διαδικασιών όπως οι Variance Exploding (VE) και Variance Preserving (VP) στοχαστικές διαφορικές εξισώσεις.

3. Συγκρίνονται διαφορετικές πολυτροπικές αρχιτεκτονικές για μοντέλα διάχυσης(diffusion backbone), με εναλλακτικά conditioning σχήματα.

4. Δοκιμάζονται αποδοτικές μέθοδοι δειγματοληψίας (samplers) για την αντίστροφη επίλυση των στοχαστικών διαφορικών εξισώσεων (ΣΔΕ).

5. Παρουσιάζονται δύο βελτιστοποιημένες παραμετροποιήσεις: μία για μέγιστη ποιότητα ανακατασκευής και μία για ταχεία εξαγωγή, και οι δύο με ανταγωνιστικά αποτελέσματα έναντι της σχετικής βιβλιογραφίας.

Με τις παραπάνω αναζητήσεις επιτύχαμε: πιο σταθερή και αποδοτική εκπαίδευση χωρίς παραγωγή τροπικοτήτων από μη-εκπαιδευμένα diffusion models, βελτιωμένη ποιότητα ανακατασκευής ελλειπών τροπικοτήτων, σημαντική μείωση χρόνου δειγματοληψίας μέσω βελτιστοποιημένων δειγματολειπτών (samplers), καθώς και την ανάπτυξη ενός νέου πολυτροπικού δικτύου που αξιοποιεί καλύτερα τις διασταυρούμενες εξαρτήσεις των δεδομένων το οποίο συγκρίναμε και αξιολογίσαμε με άλλες συγχρόνες εναλλακλτικές αρχιτεκτονικές για υπό συνθήκη (conditional generation) παραγωγή σε μοντέλα διάχυσης. Συνολικά, η μεθοδολογία μας ανταγωνίζεται και σε πολλές περιπτώσεις ξεπερνά τα βασικά μοντέλα της βιβλιογραφίας σε απόδοση και ταχύτητα, φέρνοντας τα μοντέλα διάχυσης πιο κοντά σε λύσεις για πραγματικές εφαρμογές ελλειπών πολυτροπικών εργασιών αναγνώρισης συναισθήματος που χρειάζονται συστήματα πραγματικού χρόνου.

## 0.2 Μοντέλα Διάχυσης και Γενετική Μοντελοποίηση μέσω Στοχαστικών Διαφορικών Εξισώσεων

Τα πιθανοτικά μοντέλα διάχυσης [38, 11] ή η προσέγγιση Score Matching [39] ορίζουν μια γενετική διαδικασία εισάγοντας σταδιακά θόρυβο στα δεδομένα και στη συνέχεια μαθαίνοντας πώς να τον αναιρούν. Δεδομένης μιας κατανομής δεδομένων $p_0(\mathbf{x})$, η εμπρόσθια διαδικασία (forward process) προσθέτει Γκαουσιανό θόρυβο σε $T$ βήματα σχηματίζοντας μια αλυσίδα Markov:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}), \tag{1}$$

όπου η $\alpha_t$ είναι ένα πρόγραμμα διακύμανσης που ελέγχει το μέγεθος του θορύβου. Στην συνεχή χρονική διατύπωση [2], η εμπρόσθια διαδικασία περιγράφεται από μια Στοχαστική Διαφορική Εξίσωση (ΣΔΕ):

$$d\mathbf{x} = f(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \tag{2}$$

όπου $\mathbf{w}$ είναι η τυπική κίνηση Brown. Η αντίστροφη ΣΔΕ που χρησιμοποιείται για παραγωγή δειγμάτων είναι:

$$d\mathbf{x} = \left[f(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + g(t)d\bar{\mathbf{w}}. \tag{3}$$

Η συνάρτηση βαθμίδας πιθανότητας (score function)$\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ προσεγγίζεται από ένα νευρωνικό δίκτυο $s_\theta(\mathbf{x}, t)$, το οποίο εκπαιδεύεται με denoising score matching:

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{\mathbf{x}_0,t,\boldsymbol{\epsilon}} \left[ \left\| \sqrt{\lambda(t)} s_\theta(\mathbf{x}_t, t) + \boldsymbol{\epsilon} \right\|^2 \right], \tag{4}$$

όπου $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ και η $\lambda(t)$ είναι συνήθως η διακύμανση του θορύβου. Αυτή η διατύπωση επιτρέπει εκφραστική γενετική μοντελοποίηση σε πλήθος πεδίων [11, 12]. Στο Σχήμα 1 απεικονίζεται πώς ένα δείγμα αλλοιώνεται μέσω της εμπρόσθιας ΣΔΕ σε θόρυβο $x(T)$ και πώς ανακτάται μέσω της αντίστροφης ΣΔΕ αξιοποιώντας τη μαθημένη συνάρτηση βαθμίδας πιθανότητας (score function) .

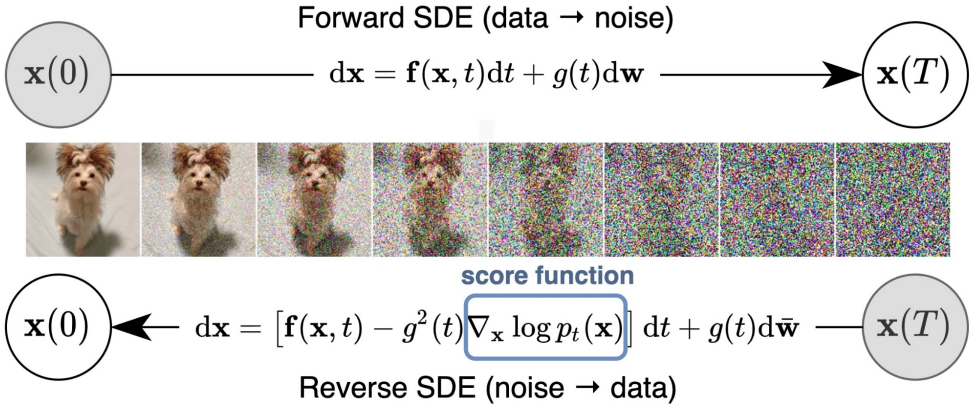


**Figure 1.** *Απεικόνιση των εμπρόσθιων και αντίστροφων ΣΔΕ διαδικασιών στη μοντελοποίηση διάχυσης. Η εμπρόσθια ΣΔΕ προσθέτει σταδιακά θόρυβο σε καθαρό δείγμα* $\mathbf{x}(0)$ *μέχρι να γίνει καθαρός θόρυβος* $\mathbf{x}(T)$*, ενώ η αντίστροφη ΣΔΕ χρησιμοποιεί τη μαθημένη συνάρτηση βαθμίδας πιθανότητας για να αποθορυβοποιήσει το* $\mathbf{x}(T)$ *και να ανακτήσει ένα δείγμα. Προσαρμογή από [2].*

Παρά τις εξαιρετικές δυνατότητές τους, τα μοντέλα διάχυσης υποφέρουν από ένα κρίσιμο υπολογιστικό εμπόδιο: την απαίτηση για εκατοντάδες ή και χιλιάδες διαδοχικά βήματα αποθορυβοποίησης κατά τη δειγματοληψία. Αυτό καθιστά την εξαγωγή δειγμάτων σημαντικά πιο αργή σε σχέση με άλλα γενετικά μοντέλα, όπως τα GANs ή τα VAEs. Η υψηλή αυτή υπολογιστική δαπάνη έχει οδηγήσει σε εκτενή έρευνα για μεθόδους επιτάχυνσης, όπως:

- Ντετερμινιστικές προσεγγίσεις όπως το DDIM [40],
- Βελτιστοποίηση θορύβου και χρονοπρογραμματισμού όπως στο iDDPM [41],
- Προηγμένες μέθοδοι δειγματοληψίας όπως το EDM [42], που σχεδιάζουν καλύτερους αριθμητικούς επιλυτές χωρίς αλλαγή στα εκπαιδευμένα μοντέλα,
- Τεχνικές απόσταξης όπως η προοδευτική απόσταξη [13], τα consistency models [43], και η απόσταξη γνώσης [44], που επιτρέπουν στα μοντέλα να προσομοιώνουν τη διαδικασία πολλών βημάτων με λίγα βήματα ή ακόμα και με ένα μόνο βήμα.
- Αριθμητικοί επιταχυντές (fast ODE solvers) όπως ο DPM-Solver [45],

## 0.3 Πολυτροπική μάθηση με ελλειπείς τροπικότητες

Η Αναγνώριση Συναισθήματος από πολλαπλές τροπικότητες (πολυτροπική αναγώριση συναισθήματος ΠΑΣ) στοχεύει στην εκτίμηση ανθρώπινου συναισθήματος από συγχρονισμένα σήματα όπως κείμενο, ήχος και εικόνα. Βενςημαρκ δατασετς όπως τα CMU-MOSI [46] και CMU-MOSEI [47] παρέχουν ευθυγραμμισμένες φράσεις με κατηγορικές ή συνεχείς ετικέτες συναισθήματος. Παρά τις δυνατότητες της πολυτροπικής μάθησης, η πολυτροπική αναγνώρισης συναισθήματος (ΠΑΣ) παραμένει πρόκληση λόγω ετερογένειας σημασιολογικής, χρονικής ασυγχρονίας και θορύβου μεταξύ των τροπικοτήτων. Ένα κρίσιμο πρακτικό πρόβλημα είναι η συχνή εμφάνιση ελλειπόντων τροπικοτήτων κατά τη διάρκεια της δειγματοληψίας ή ανάπτυξης, όπως απουσία ήχου ή κατεστραμμένο βίντεο.

Οι παραδοσιακές προσεγγίσεις ΠΑΣ όπως τα MISA [32] και MulT [26] επιτυγχάνουν ισχυρή απόδοση μοντελοποιώντας διασταυρούμενες αλληλεπιδράσεις και μαθαίνοντας ευθυγραμμισμένες αναπαραστάσεις. Ωστόσο, αυτές υποθέτουν πλήρη διαθεσιμότητα όλων των τροπικοτήτων κατά τη δοκιμή, περιορίζοντας την ανθεκτικότητα σε πραγματικά σενάρια.

Για να αντιμετωπιστεί αυτό, η βιβλιογραφία έχει εξελιχθεί σε δύο κατευθύνσεις: *μη-αναπληρωτικές μέθοδοι*, που μαθαίνουν να συνενώνουν μόνο τις διαθέσιμες εισόδους, και *μέθοδοι αναπλήρωσης*, που ανακατασκευάζουν ρητά τις ελλείπουσες τροπικότητες πριν από τη συνένωση. Στις τελευταίες, σύγχρονες μέθοδοι αξιοποιούν γραφο-βασισμένα μοντέλα

όπως το GCNet [30] και τεχνικές ανακατασκευής με δικτύα φαντασίας (imagination-driven reconstruction) [6]. Πιο πρόσφατα, γενετικά μοντέλα όπως τα DiCMoR [35] και IMDER [1] μοντελοποιούν υπό συνθήκη κατανομές $p(x_i \mid x_j, x_k)$ μέσω κανονικοποιημένων ροών ή διαδικασιών διάχυσης για δειγματοληψία κατά την εκτέλεση downstream, αυξάνοντας την ανθεκτικότητα.

## 0.4 Προτεινόμενη μεθοδολογία

### 0.4.1 Γενική Διατύπωση

Έστω ότι $\mathcal{X} = \{x_l, x_v, x_a\}$ δηλώνει τον χώρο εισόδων τριών τροπικοτήτων—language ($x_l$), vision ($x_v$), και audio ($x_a$)—που αντιστοιχούν σε μια μοναδική φράση. Στην προβλήμα πολυτροπικής αναγνώρισης συναισθήματος (Multimodal Emotion Recognition) (MER), ο στόχος είναι να μάθουμε μια συνάρτηση $\mathcal{F} : \mathcal{X} \to y$ που αντιστοιχεί τις παρατηρούμενες τροπικότητες σε μια διακριτή ή συνεχής ετικέτα συναισθήματος $y$.

Στις εφαρμογές πραγματικού κόσμου, όμως, δεν είναι πάντα διαθέσιμες όλες οι τροπικότητες κατά τη φάση της αναγνώρησης συναισθήματος (inference). Έστω ότι $\mathcal{M} \subseteq \{l, v, a\}$ δηλώνει το σύνολο των διαθέσιμων τροπικοτήτων σε ένα δείγμα, και $\mathcal{M}^c$ το συμπλήρωμά του—το σύνολο των ελλειπόντων τροπικοτήτων. Η κεντρική πρόκληση είναι να εκτιμήσουμε ή να προσεγγίσουμε τα ελλείποντα στοιχεία $\{x_m : m \in \mathcal{M}^c\}$ υπό συνθήκη των διαθέσιμων $\{x_o : o \in \mathcal{M}\}$, έτσι ώστε η τελική ταξινόμηση συναισθήματος να παραμένει ανθεκτική.

Τυπικά, στοχεύουμε στη μοντελοποίηση των κατανομών υπό συνθήκη:

$$p_\theta(x_m(0) \mid x_o(0)), \quad \text{για όλα τα } m \in \mathcal{M}^c, \tag{5}$$

όπου $x(0)$ αναφέρεται σε καθαρά δεδομένα (δηλαδή στο χρόνο $t = 0$ στη διαδικασία διάχυσης. Αφού δειγματοληφθούν ή αναπληρωθούν οι ελλείπουσες τροπικότητες, περνάμε τόσο τις διαθέσιμες όσο και τις ανακατασκευασμένες τροπικότητες σε ένα (fusion model) $\mathcal{T}_k$ για την τελική πρόβλεψη:

$$\hat{y} = \mathcal{T}_k(x_l^*, x_v^*, x_a^*), \tag{6}$$

όπου $x_m^* = x_m$ αν $m \in \mathcal{M}$ και $x_m^* = \hat{x}_m$ (δειγματοληφθέν) αν $m \in \mathcal{M}^c$.

Για να μοντελοποιήσουμε τις κατανομές υπό συνθήκη $p_\theta(x_m(0) \mid x_o(0))$, χρησιμοποιούμε ένα πλαίσιο εκπαίδευσης για μοντέλα διάχυσης. Σε κάθε τροπικότητα ανατίθεται ένα ξεχωριστό score network $s_m(\cdot, t)$, εκπαιδευμένο μέσω score matching για να προσεγγίσει

την κλίση της κατανομής των δεδομένων στον χρόνο $t$:

$$s_m(x_m(t), t \mid x_o(t)) \approx \nabla_{x_m} \log p_t(x_m \mid x_o). \tag{7}$$

Μετά την αναπλήρωση των ελλειπόντων δεδομένων μέσω δειγματοληψίας χρησιμοποιώντας τα παραπάνω δίκτυα προσέγγισης βαθμίδας πιθανότητας (score function), χρησιμοποιείται ένας modality-specific alignment decoder $\mathcal{D}_m$ για περαιτέρω βελτίωση των παραγόμενων χαρακτηριστικών, και τελικά εφαρμόζεται ένας τελικός fusion classifier $\mathcal{T}$ με modality-specific transformers. Το πλήρες δίκτυο φαίνεται στο Σχήμα 3, και θα αναλύσουμε κάθε συστατικό του.

### 0.4.2 Διατύπωση ΣΔΕ και Αντίστροφη Διαδικασία

Λαμβάνουμε υπόψη τις δύο πιο δημοφιλείς διαδικασίες διάχυσης, τις **Variance Exploding (VE)** και **Variance Preserving (VP)** ΣΔΕ, όπως περιγράφονται στο θεμελιώδες έργο των Song et al. [2]. Και οι δύο διατυπώσεις παρέχουν συνεχή χρονικά πλαίσια για σταδιακή αλλοίωση των δεδομένων και μετέπειτα δημιουργία μέσω αντίστροφων διαδικασιών.

#### Variance Exploding (VE) SDE

Η διατύπωση **Variance Exploding (VE)** ορίζει μια συνεχή χρονικά εμπρόσθια διαδικασία διάχυσης (forward diffusion process)όπου προστίθεται σταδιακά θόρυβος στο δείγμα χωρίς να μεταβάλλεται το μέγεθος του σήματος:

$$d\mathbf{x} = \sigma(t) d\mathbf{w}, \quad \text{με} \quad \sigma(t) = \sigma^t, \tag{8}$$

όπου $\mathbf{w}$ δηλώνει τυπική Brownian motion, και $\sigma(t)$ είναι ένας χρονικά εξαρτώμενος συντελεστής διάχυσης που αυξάνεται εκθετικά για $t \in [0, 1]$. Στα πειράματά μας, ορίζουμε $\sigma = 25$ σύμφωνα με [1].

Η εμπρόσθια ΣΔΕ (forward SDE) ορίζει μια οικογένεια αλλοιωμένων κατανομών $p_t(\mathbf{x})$, όπου η οριακή κατανομή στον χρόνο $t$ έχει τυπική απόκλιση:

$$\text{std}_t = \sqrt{\frac{\sigma^{2t} - 1}{2 \ln \sigma}}, \tag{9}$$

τέτοια ώστε $\mathbf{x}(t) = \mathbf{x}(0) + \mathbf{z} \cdot \text{στδ}_t$ με $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, και το $\mathbf{x}(t)$ γίνεται όλο και πιο θορυβώδες καθώς $t \to 1$.

Η γενετική διαδικασία αντιστοιχεί στην προσομοίωση της αντίστροφου χρόνου ΣΔΕ

(reverse-time SDE):

$$d\mathbf{x} = -\sigma(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) dt + \sigma(t) d\bar{\mathbf{w}}, \tag{10}$$

όπου $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ είναι η score function και $\bar{\mathbf{w}}$ δηλώνει αντίστροφη Brownian motion.

### Variance Preserving (VP) SDE

Η διατύπωση **Variance Preserving (VP)** διατηρεί περίπου σταθερή διασπορά κατά τη διάρκεια της διαδικασίας διάχυσης προσθέτοντας ταυτόχρονα θόρυβο και κλιμακώνοντας το σήμα:

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}, \tag{11}$$

όπου $\beta(t)$ είναι χρονικά εξαρτώμενο πρόγραμμα θορύβου. Χρησιμοποιούμε γραμμικό πρόγραμμα $\beta(t) = \beta_0 + t(\beta_1 - \beta_0)$ με $\beta_0 = 0.1$ και $\beta_1 = 20.0$ σύμφωνα με την κοινή πρακτική.

Η οριακή κατανομή στον χρόνο $t$ για την VP ΣΔΕ έχει τυπική απόκλιση:

$$\text{std}_t = \sqrt{1 - \exp(2 \log \alpha(t))}, \tag{12}$$

όπου $\log \alpha(t) = -0.25t^2(\beta_1 - \beta_0) - 0.5t\beta_0$ είναι το λογάριθμο του συντελεστή κλιμάκωσης σήματος, ώστε $\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \mathbf{z} \cdot \sigma\text{τδ}_t$ με $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$.

Η αντίστοιχη reverse-time SDE για γενετική διαδικασία είναι:

$$d\mathbf{x} = \left[-\frac{1}{2}\beta(t)\mathbf{x} - \beta(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + \sqrt{\beta(t)}d\bar{\mathbf{w}}. \tag{13}$$

### Προσέγγιση της Score Function

Για και τις δύο διατυπώσεις ΣΔΕ, προσεγγίζουμε την μη υπολογίσιμη score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ χρησιμοποιώντας ένα νευρωνικό δίκτυο $s_\theta(\mathbf{x}, t)$, εκπαιδευμένο με denoising score matching. Ο στόχος εκπαίδευσης ελαχιστοποιεί:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{z}}\left[\|s_\theta(\mathbf{x}_t, t) \cdot \sigma\text{τδ}_t + \mathbf{z}\|^2\right], \tag{14}$$

όπου $t \sim \mathcal{U}[\epsilon, 1-\epsilon]$ με $\epsilon = 10^{-5}$, $\mathbf{x}_0$ είναι καθαρά δεδομένα, $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ είναι θόρυβος, και $\mathbf{x}_t = \mathbf{x}_0 + \mathbf{z} \cdot \sigma\text{τδ}_t$ είναι το δείγμα μετά την προσθήκη θορύβου στον χρόνο $t$. Για παραγωγή ύπο συνθήκη (conditional generation), εφαρμόζουμε την ίδια παραμόρφωση τόσο στις στοχευμένες όσο και στις τροπικότητες υπό συνθήκη, όπως περιγράφεται στην προσέγγιση κοινής παραμόρφωσης που χρησιμοποιούμε.

Η επιλογή μεταξύ των διατυπώσεων VE και VP εξαρτάται από την εφαρμογή και τα χαρακτηριστικά των δεδομένων. Οι VE SDEs προτιμώνται συχνά για παραγωγή χωρίς συνθήκη (unconditional generation), ενώ οι VP SDEs παρέχουν πιο σταθερή εκπαίδευση σε σενάρια παραγωγής ύπο συνθήκη όπως το δικό μας.

### 0.4.3 Αποσυνδεδεμένη Εκπαίδευση

Στο προτεινόμενο πλαίσιο μας, κάθε modality-specific diffusion model εκπαιδεύεται ανεξάρτητα για να εκτιμήσει την κατανομή υπό συνθήκη δεδομένων από τις υπόλοιπες τροπικότητες. Σε αντίθεση με τις ολοκληρωμένες end-to-end διαδικασίες όπως το IMDER [1], αποσυνδέουμε την γενετική εκπαίδευση κάθε score network από τον τελικό κατηγοριοποιητή, όπως φαίνεται στο Στάδιο 1 του Σχήματος 2. Αυτό επιτρέπει τη χρήση ενός πλήρως εκπαιδευμένου score model για αναπλήρωση τιμών στη downstream εργασία. Αντίθετα με την προηγούμενη προσέγγιση του IMDER, όπου η εκπαίδευση end-to-end οδηγούσε σε κακής ποιότητας αναπληρωμένες τροπικότητες και μη σταθερά gradients, η αποσυνδεδεμένη εκπαίδευση προσφέρει αποτελεσματική και αρθρωτή εκπαίδευση, διευκολύνοντας καλύτερο έλεγχο της αρχιτεκτονικής, του μηχανισμού conditioning και της στρατηγικής δειγματοληψίας.

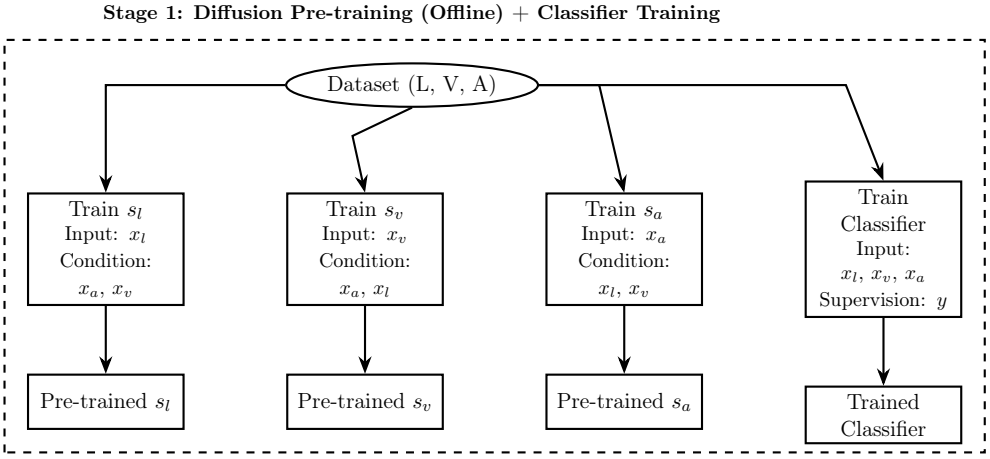


**Figure 2.** *Διάγραμμα που αναπαριστά το πρώτο Stage της εκπαίδευσης του αποσυνδεδεμένου (decoupled) modality diffusion MER network. Τα modality-specific score networks εκπαιδεύονται χρησιμοποιώντας το πλήρες σύνολο δεδομένων με τυχαία καθοδήγηση (randomized conditioning), ενώ ο downstream classifier εκπαιδεύεται επίσης στο πλήρες σύνολο δεδομένων για το MER task, με πρόσβαση σε πλήρως παρατηρούμενες τροπικότητες (fully observable modalities).*

Έστω $x = \{x_l, x_v, x_a\}$ το πλήρες multimodal feature tuple για ένα δείγμα. Για να

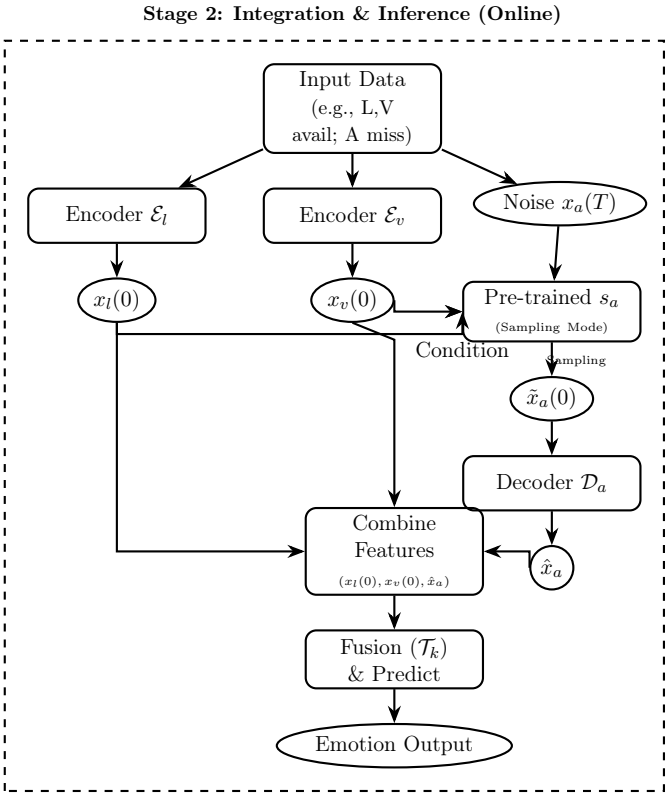


**Figure 3.** *Διάγραμμα που απεικονίζει τη προτεινόμενη εκπαίδευση του 2ου σταδίου της μεθοδολογίας μας, δείχνοντας ένα παράδειγμα όπου η ακουστική τροπικότητα (acoustic modality) λείπει. Αρχικά, δειγματοληπτούμε θόρυβο με διακύμανση βάσει της διαδικασίας διάχυσης. Έπειτα εκτελείται η αντίστροφη διαδικασία diffusion sampling μέσω του εκπαιδευμένου audio score network $s_\alpha$ και παράγεται μια αδρή ανακατασκευασμένη τροπικότητα $\tilde{x}_\alpha$. Στη συνέχεια, αυτή βελτιώνεται περαιτέρω περνώντας από τον alignment decoder $D_\alpha$, πριν χρησιμοποιηθεί στην ανίχνευση συναισθήματος (emotion inference) μέσω του fusion classifier $T_k$.*

αναπληρώσουμε μια ελλείπουσα τροπικότητα $x_m \in x$, στοχεύουμε στη μοντελοποίηση της κατανομής υπό συνθήκη $p(x_m(0) \mid x_o(0))$, όπου $x_o(0) \subset x \setminus x_m$ είναι οι διαθέσιμες τροπικότητες, και $x_m(0)$ τα καθαρά δεδομένα της ελλείπουσας τροπικότητας. Ωστόσο, όπως δείχνει το [2], μπορεί να εκπαιδευτεί ένα μόνο score network για να προσεγγίσει τις κλίσεις της κοινής κατανομής $\nabla_x \log p_t(x)$ μέσω παραμόρφωσης όλων των μεταβλητών..

**Παραμόρφωση τόσο των στοχευμένων όσο και των υπό συνθήκη τροπικοτήτων.** Αντί να παγώνουμε τις τροπικότητες υπό συνθήκη και να παραμορφώνουμε μόνο τη στοχευμένη τροπικότητα, εισάγουμε θόρυβο τόσο στο $x_m$ όσο και στις $x_o$ χρησιμοποιώντας την ίδια forward SDE (όπως και στο IMDER):

$$d\mathbf{x} = f(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \quad \mathbf{x}(0) \sim p_0(\mathbf{x}). \tag{15}$$

Αυτό οδηγεί σε θορυβώδεις εκδόσεις $x_m(t) \sim q(x_m(t) \mid x_m(0))$ και $x_o(t) \sim q(x_o(t) \mid x_o(0))$. Κατά την εκπαίδευση, παρέχουμε τις θορυβώδεις τροπικότητες υπό συνθήκη $x_o(t)$ αντί για τις καθαρές $x_o(0)$, ενθαρρύνοντας το score network να μάθει την αληθινή κοινή κατανομή $(x_m, x_o)$ αντί να βασίζεται σε συντομεύσεις από καθαρές εισόδους.

Θεωρητικά, αυτό προκύπτει από τη διατύπωση στο [2], όπου η ελαχιστοποίηση της απώλειας denoising score matching (DSM) σε όλα τα συστατικά που παραμορφώνονται από την SDE παρέχει έναν εκτιμητή της score function $\nabla_x \log p_t(x)$. Στην περίπτωσή μας, αυτό επιτρέπει τη μοντελοποίηση της conditional score function:

$$\nabla_{x_m} \log p_t(x_m \mid x_o) \approx s_\theta(x_m(t), t, x_o(t)), \tag{16}$$

όπου $s_\theta$ είναι νευρωνικό δίκτυο που προσεγγίζει τη conditional score function. Η χρήση των θορυβωδών $x_o(t)$ κανονικοποιεί την εκπαίδευση και επιτρέπει στο μοντέλο να γενικεύει σε ποικίλα μοτίβα απουσίας κατά την inference.

**Στόχος Εκπαίδευσης.** Δεδομένης αυτής της διατύπωσης, κάθε score network $s_m(\cdot)$ για τη τροπικότητα $m \in \{l, v, a\}$ εκπαιδεύεται με στόχο DSM:

$$\mathcal{L}_m = \mathbb{E}_{x_m(0), x_o(0), t, \epsilon} \left[ \left\| \sqrt{\sigma(t)} s_m(x_m(t), t, x_o(t), \theta_m) + \epsilon \right\|^2 \right], \tag{17}$$

όπου $x_m(t) = \alpha(t)x_m(0) + \sigma(t)\epsilon$, και $x_o(t)$ κατασκευάζεται αναλογικά. Εδώ, το $\sigma(t)$ είναι συνάρτηση βάρους για την εκτίμηση της score function από τον θόρυβο όπως περιγράφεται στο [2]. Προσομοιώνουμε διαφορετικές ρυθμίσεις απουσίας σε κάθε batch, διασφαλίζοντας ότι το μοντέλο μαθαίνει να βασίζεται σε μεταβλητά υποσύνολα των τροπικοτήτων. Αυτή η

προσέγγιση μαθαίνει αποτελεσματικά μια οικογένεια conditional generative models $p_\theta(\mathbf{x}_m \mid \mathbf{x}_o)$ για κάθε τροπικότητα $m$.

Μετά την εκπαίδευση, τα score networks παγώνουν και χρησιμοποιούνται σε λειτουργία inference για δειγματοληψία ελλειπόντων δεδομένων (π.χ. μέσω αντίστροφης SDE ή probability-flow ODE samplers). Αυτές οι δειγματοληφθείσες αναπαραστάσεις περνούν στη συνέχεια από έναν decoder και συγχωνεύονται μέσω ενός downstream transformer για πρόβλεψη συναισθήματος (βλ. Σχήμα 5.2, Στάδιο 2).

### 0.4.4 Αρχιτεκτονικές δικτύων για μοντέλα διάχυσης

Η αρχιτεκτονική των score networks $s_m$ και ο μηχανισμός conditioning αποτελούν κρίσιμες σχεδιαστικές επιλογές που καθορίζουν τόσο την ποιότητα των παραγόμενων ελλειπόντων δεδομένων όσο και την υπολογιστική αποτελεσματικότητα. Αξιολογούμε συστηματικά διαφορετικές αρχιτεκτονικές backbone σε συνδυασμό με διάφορες τεχνικές conditioning για να προσδιορίσουμε την ιδανική διαμόρφωση για την πολυτροπική αναγνώριση συναισθημάτων.

#### Αρχιτεκτονικές Backbone

Συγκρίνουμε τέσσερις διαφορετικές αρχιτεκτονικές score network, η καθεμία αντιπροσωπεύει διαφορετικές προσεγγίσεις για μοντελοποίηση χρονικών εξαρτήσεων και cross-modal αλληλεπιδράσεων:

- **U-Net with Cross-Attention (Baseline IMDER):** Ακολουθώντας τη τυπική προσέγγιση στα diffusion models [48, 17], αυτή η αρχιτεκτονική χρησιμοποιεί convolutional layers για εξαγωγή χαρακτηριστικών με skip connections. Οι χρονικές ενσωματώσεις (time embeddings) προβάλλονται σε ενδιάμεσα επίπεδα, ενώ η conditioning στις διαθέσιμες τροπικότητες επιτυγχάνεται μέσω μηχανισμών cross-attention, όπως αναπτύχθηκε στο Stable Diffusion [17] (Η πλήρης αρχιτεκτονική φαίνεται στο Σχήμα 4).

- **Multimodal Diffusion Transformer:** Βασιζόμενοι στη Feature-wise Linear Modulation (FiLM) [49], υιοθετούμε την αρχιτεκτονική που προτάθηκε από τους Esser et al. [4]. Οι τροπικότητες υπό conditioning επεξεργάζονται μέσω ενός modulation flow σε κάθε transformer block για συγχώνευση της πληροφορίας χρονικού βήματος, παράγοντας διαφορετικούς συντελεστές κλίμακας ($\gamma$) και μετατόπισης ($\beta$) για τις εισόδους και τις τροπικότητες υπό conditioning. Στη συνέχεια, οι είσοδοι και οι τροπικότητες υπό conditioning συνενώνονται στο self-attention για εξαγωγή πληροφοριών (βλ. Σχήμα 5).

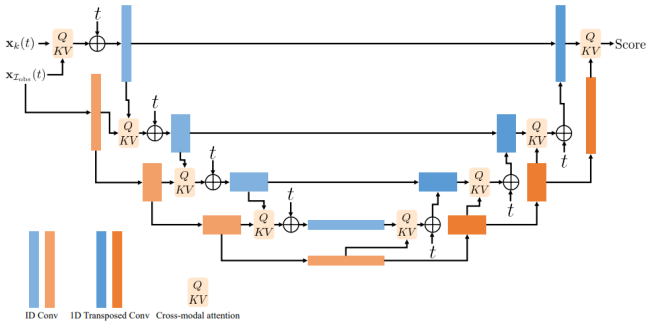


**Figure 4.** *Απεικόνιση του network που χρησιμοποιήθηκε στα πειράματά μας. Χρησιμοποιήθηκε U-Net 4 επιπέδων encoder-decoder με residual connections και μηχανισμούς cross-attention στις παρατηρούμενες τροπικότητες. Σχήμα τροποποιήθηκε άπο εδώ [3]*

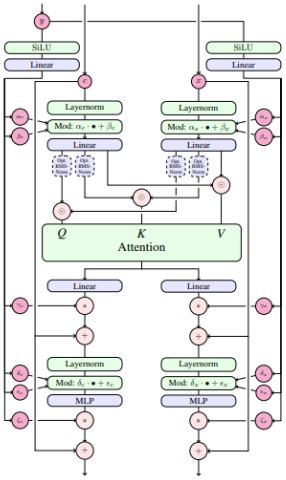


**Figure 5.** *Block του Multimodal Diffusion Transformer που επεξεργάζεται ξεχωριστά τις σειρές εισόδου και conditioning, συνενώνει και τις δύο για self-attention και προσθέτει επιπλέον modulation για καλύτερο conditioning. Σχήμα τροποποιήθηκε άπο εδώ [4]*

- **Diffusion-Transformer:** Αυτή η παραλλαγή προτάθηκε από τους Peebles et al. [5] και χρησιμοποιεί μια ειδική μορφή Adaptive Layer Normalization (AdaLN) που ονομάζεται AdaLN-zero. Το AdaLN τροποποιεί τους συντελεστές κλίμακας και μετατόπισης της layer normalization βάσει των διαθέσιμων τροπικοτήτων, παρέχοντας λεπτομερή έλεγχο της ροής πληροφορίας υπό conditioning χωρίς το υπολογιστικό κόστος επιπλέον μηχανισμών attention. Στο Di-Transformer, ένας επιπλέον συντελεστής κλίμακας $\alpha$ εισάγεται πριν από κάθε residual connection για επιπλέον έλεγχο conditioning (βλ. Σχήμα 6).

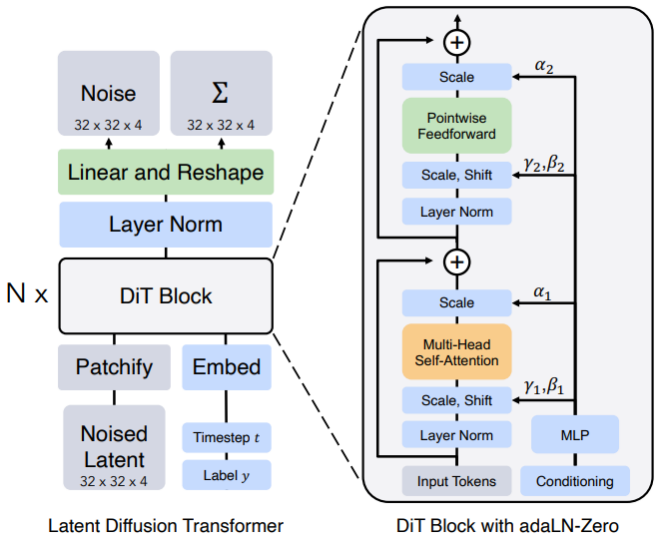


**Figure 6.** *Αρχιτεκτονική ενός single block του Diffusion Transformer. Πρόκειται για βελτιωμένο FiLM conditioning με κλιμάκωση και μετατόπιση μετά από κάθε layer normalization και επιπλέον παράγοντες κλίμακας $\alpha$. Σχήμα τροποποιήθηκε άπο εδώ [5].*

- **ScoreTransformer1D(Νέα Προσέγγιση):** Προτείνουμε μια lightweight αρχιτεκτονική βασισμένη σε transformer για 1Δ ακολουθίες χαρακτηριστικών. Η αρχιτεκτονική εμπνέεται από το UniDiffuser [18], όπου οι τροπικότητες συνενώνονται ως είσοδος του transformer, κάθε μια με τη δική της timestep embedding. Εμείς παραλείπουμε τις επιπλέον ενσωματώσεις χρονικού βήματος και τις εισάγουμε μόνο στην είσοδο, καθώς είναι κοινές με τη χρονική πληροφορία των δεδομένων υπό conditioning. Αντί των convolutional U-Net αρχιτεκτονικών, το ScoreTransformer1D αντικαθιστά τις χωρικές συνελίξεις με token-mixing transformer blocks που μπορούν να μοντελοποιούν μακροχρόνιες χρονικές εξαρτήσεις. Με τη concatenation-based con-

ditioning, όλες οι τροπικότητες αντιμετωπίζονται ομοιόμορφα εντός ενός transformer framework, επιτρέποντας τη μάθηση croos-modal εξαρτήσεων μέσω self-attention.

Η είσοδος αποτελείται από τρία στοιχεία:

- $x \in \mathbb{R}^{B \times C \times T}$: θορυβώδες feature tensor για τη στοχευμένη τροπικότητα[1]
- $\gamma(t) \in \mathbb{R}^{B \times 1 \times D}$: timestep embedding που προστίθεται σε κάθε κανάλι του $x$
- $c \in \mathbb{R}^{B \times C \times T^c}$: είσοδος υπό conditioning που συντίθεται από τις θορυβώδεις τροπικότητες $x_{\setminus m}(t)$

Η μονάδα υπολογίζει πρώτα την time-conditioned latent representation:

$$\tilde{x} = x + \gamma(t) \tag{18}$$

Στη συνέχεια συνενώνει την είσοδο υπό conditioning:

$$z = [\tilde{x}; c] \in \mathbb{R}^{B \times C \times (T + T^c)} \tag{19}$$

Αυτή η αρχικτεκτονική επεξεργάζεται την είσοδο μέσω ενός transformer encoder που εφαρμόζει self-attention σε κάθε χρονικό βήμα. Η έξοδος φιλτράρεται για να εξαχθεί μόνο ο προβλεπόμενος θόρυβος για την τροπικότητα $x_m$.

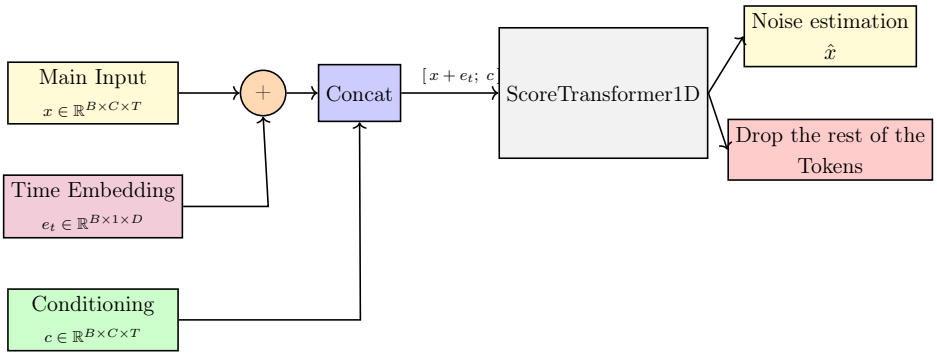


**Figure 7.** *Διάγραμμα για τη προτεινόμενη αρχιτεκτονική του* `ScoreTransformer1D`.

### Ενσωμάτωση χρονικού βήματος (Timestep Embeddings)

Ακολουθώντας προηγούμενες εργασίες [50], χρησιμοποιούμε Gaussian Fourier time embeddings για όλες τις ραχοκοκαλιές ώστε να αναπαραστήσουμε το συνεχές χρονικό βήμα

[1] Όλες οι τροπικότητες προβάλλονται σε κοινό feature space μέσω ενός απλού feature encoder.

της διαδικασίας διάχυσης ως υψηλής διάστασης περιοδικό σήμα:

$$\gamma(t) = [\sin(2\pi W t), \cos(2\pi W t)], \quad W \in \mathbb{R}^{D/2} \tag{20}$$

όπου $W$ είναι σταθερός Gaussian matrix και $D$ η συνολική διάσταση της embedding.

### Σχήμα Εκπαίδευσης μοντέλων διάχυσης

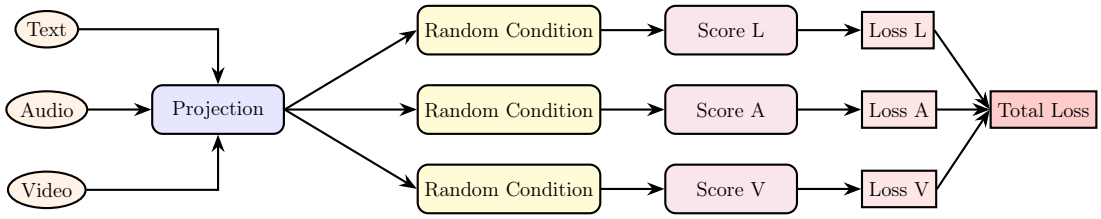


**Figure 8.** *Υψηλού επιπέδου διάγραμμα της διαδικασίας εκπαίδευσης του diffusion model, όπου όλα τα score networks εκπαιδεύονται σε μία μόνο προώθηση (forward pass).*

Στο Σχήμα 8 παρουσιάζεται η προσέγγισή μας για την εκπαίδευση όλων των score networks. Η διαδικασία εκπαίδευσης ακολουθεί ένα multi-target scheme, όπου όλα τα τρία modality-specific score networks εκπαιδεύονται ταυτόχρονα σε ένα μόνο forward pass:

1. **Επεξεργασία Εισόδου**: Οι ακατέργαστες είσοδοι κειμένου, ήχου και βίντεο περνούν μέσα από modality-specific projection layers για να χαρτογραφηθούν σε έναν κοινό χώρο χαρακτηριστικών.

2. **Τυχαία Επιλογή τροπικοτήτων υπό Conditioning**: Για κάθε δίκτυο, παραλείπονται τυχαία 1-2 τροπικότητες και οι υπόλοιπες χρησιμοποιούνται ως είσοδος υπό συνθήκη. Αυτή η στρατηγική εξασφαλίζει ότι το μοντέλο μαθαίνει να χειρίζεται διάφορα μοτίβα ελλιπών δεδομένων κατά την εκπαίδευση, βελτιώνοντας τη γενίκευση σε διαφορετικά σενάρια κατά την inference.

3. **Παράλληλος Υπολογισμός Score**: Τρία ξεχωριστά score networks (Score L, Score A, Score V) προβλέπουν ταυτόχρονα τις score functions για τις τροπικότητες κειμένου, ήχου και βίντεο. Κάθε δίκτυο λαμβάνει τη θορυβώδη στοχευμένη τροπικότητα μαζί με τις τυχαία επιλεγμένες τροπικότητες υπό conditioning.

4. **Συνδυασμός Απωλειών**: Υπολογίζονται οι ατομικές απώλειες για κάθε modality-specific score network χρησιμοποιώντας την συνάρτηση κόστους denoising score matching loss, και αυτές οι απώλειες συνενώνονται σε μία συνολική απώλεια.

Ο μηχανισμός τυχαίας επιλογής τροπικότητων (random modality conditioning) εξασφαλίζει ότι κάθε score network μαθαίνει να αξιοποιεί διαφορετικούς συνδυασμούς διαθέσιμων τροπικοτήτων, καθιστώντας το σύστημα ανθεκτικό σε διάφορα μοτίβα ελλιπών δεδομένων κατά την inference.

### 0.4.5 Αλγόριθμοι Δειγματοληψίας

Μετά την εκπαίδευση, κάθε modality-specific score network $s_\theta^m$ χρησιμοποιείται ως προεκπαιδευμένο generative module για την ανακατασκευή των ελλιπών τροπικοτήτων μέσω της εκμαθημένης αντίστροφης ΣΔΕ. Κατά την inference, παραλείπουμε την εμπρόσθια διαδικασία διάχυσης της εκπαίδευσης και εφαρμόζουμε αριθμητικούς δειγματολείπτες για την επίλυση της αντίστροφης ΣΔΕ.

Η επιλογή του **αλγορίθμου δειγματοληψίας** (sampling algorithm) επηρεάζει σημαντικά τόσο την ποιότητα των παραγόμενων αποτελεσμάτων όσο και την υπολογιστική αποδοτικότητα. Εξερευνούμε τέσσερις διαφορετικούς δειγματολείπτες που αντιπροσωπεύουν διάφορες ισορροπίες μεταξύ ακρίβειας και ταχύτητας.

Οι αλγόριθμοι δειγματοληψίας που θα χρησιμοποιήσουμε και η εκάστοτε πολυπλοκότητα του κάθε ενώς είναι οι εξής:

- **Euler-Maruyama**: $O(N)$ υπολογισμοί της score function, όπου $N \approx 100 - 1000$
- **Predictor-Corrector**: $O(N \cdot (1 + J))$ υπολογισμοί, με $J = 1 - 2$ βήματα corrector
- **Heun**: $O(2N)$ υπολογισμοί με $N \approx 30 - 50$, οδηγώντας σε συνολικά $\sim 60 - 100$ υπολογισμούς NFE
- **DDIM**: $O(N)$ υπολογισμοί με $N \approx 10 - 50$, επιτυγχάνοντας την ταχύτερη inference

Η ευελιξία στις μεθόδους δειγματοληψίας είναι ιδιαίτερα χρήσιμη σε εργασίες multimodal imputation, όπου διαφορετικές εφαρμογές μπορεί να δίνουν προτεραιότητα είτε στην ποιότητα της παραγόμενης εξόδου (π.χ. ιατρική απεικόνιση) είτε στην ταχύτητα εκτίμησης (real-time inference).

### 0.4.6 Αποκωδικοποιητής Ευθυγράμμισης

Αφού το δείγμα της λείπουσας τροπικότητας $\tilde{\mathbf{x}}_m(0)$ παραχθεί από το μοντέλο διάχυσης, συχνά δεν ταιριάζει με την ίδια κατανομή χαρακτηριστικών όπως τα αρχικά δεδομένα εκπαίδευσης λόγω ατελούς αποθορυβοποίησης, ειδικά όταν χρησιμοποιούνται λίγοι υπολογισμοί δικτύου (NFEs). Για να γεφυρώσουμε αυτό το χάσμα, εισάγουμε έναν αποκωδικοποιητή

ευθυγράμμισης $\mathcal{D}_m$, ο οποίος εκπαιδεύεται ώστε να αντιστοιχεί τα θορυβώδη/παραγόμενα χαρακτηριστικά στον αρχικό χώρο κατανομής χαρακτηριστικών.

Χρησιμοποιούμε μια ελαφριά έκδοση της αρχιτεκτονικής **Residual Channel Attention Block (RCAB)** [25], αρχικά σχεδιασμένη για υπερ-ανάλυση εικόνων. Στην 1Δ προσαρμογή μας, το RCAB αποτελείται από ένα υπόλειμμα convolutional block με channel attention:

- **Residual Convolution:** Εξάγει τοπικά χρονικά μοτίβα από $\tilde{\mathbf{x}}_m$.
- **Channel Attention:** Επαναβαθμονομεί τα κανάλια χαρακτηριστικών για ενίσχυση των διακριτικών ενδείξεων χρησιμοποιώντας squeeze-and-excitation [51].

Αυτός ο αποκωδικοποιητής εκπαιδεύεται ανεξάρτητα με ρεσονστρυστιον λοσς ενισχυμένο με περσεπτυαλ λοσς πάνω στις ενεργοποιήσεις του downstream μοντέλου $\mathcal{L}_{task}$:

$$\mathcal{L}_{\text{αλιγν}} = \|\mathcal{D}_m(\tilde{\mathbf{x}}_m(0)) - \mathbf{x}_m(0)\|_1, \tag{21}$$

### 0.4.7 Πολυτροπικοί Ταξινομητές συνένωσης

Για την αναγνώριση συναισθημάτων από τα συνενωμένα χαρακτηριστικά των διαφορετικών τροπικότητων, υιοθετούμε έναν μηχανισμό συγχώνευσης(fusion mechanism) εμπνευσμένο από την αρχιτεκτονική Multimodal Transformer (MulT) [26]. Συγκεκριμένα, χρησιμοποιούμε αποκωδικοποιητές ειδικούς για κάθε τροπικότητα $\mathcal{E}_m$ για να προβάλουμε τα χαρακτηριστικά σε έναν κοινό λανθάνων χώρο, και εφαρμόζουμε μια σειρά από pairwise crossmodal transformers για να μοντελοποιήσουμε τις κατευθυνόμενες εξαρτήσεις μεταξύ των τροπικοτήτων. Κάθε transformer μαθαίνει πώς να ενισχύει μια τροπικότητα χρησιμοποιώντας πληροφορίες από μια άλλη μέσω crossmodal attention, αιχμαλωτίζοντας αποτελεσματικά τις μακροχρόνιες αλληλεπιδράσεις ανάμεσα σε ροές διαφορετικού μήκους και ρυθμών δειγματοληψίας.

Οι συνενωμένες αναπαραστάσεις συγκεντρώνονται μέσω ενός memory transformer $\mathcal{T}_k$, και το τελευταίο token κάθε τροπικότητας που αναπαριστά όλα τα χαρακτηριστικά της ακολουθίας συναισθήματος συνενώνεται με τα υπόλοιπα και περνάει σε ένα multi-layer perceptron (MLP) για την τελική πρόβλεψη. Οι fusion transformers εκπαιδεύονται από κοινού με το classification head χρησιμοποιώντας cross-entropy loss για ταξινόμηση ή L1 loss για παλινδρόμηση[2]. Κατά τη διάρκεια αυτού του σταδίου, οι λείπουσες τροπικότητες είτε μηδενίζονται (zero-masked) είτε αντικαθίστανται με ανακατασκευές από το μοντέλο διάχυσης.

[2] Στις δοκιμές μας χρησιμοποιούμε μόνο την L1 παλινδρόμηση.

Ας θεωρήσουμε ότι $x_m$ αναπαριστά τα εισερχόμενα χαρακτηριστικά για την τούπλα τροπικότητας $m \in \{1, \ldots, M\}$. Κάθε τροπικότητα αρχικά κωδικοποιείται μέσω ενός ειδικού, ριχού κωδικοποιητή $\mathcal{E}_m$:

$$h_m = \mathcal{E}_m(x_m), \tag{22}$$

Στη συνέχεια, για κάθε τροπικότητα $m$, εφαρμόζουμε ζεύγη crossmodal transformers $\mathcal{T}_{m \leftarrow j}$ για να ενσωματώσουμε πληροφορίες από όλες τις άλλες τροπικότητες $j \neq m$. Οι αναπαραστάσεις αυτές συγκεντρώνονται μέσω ενός memory transformer $\mathcal{T}_m^{\text{mem}}$:

$$\tilde{h}_m = \mathcal{T}_m^{\text{mem}} \left( [\mathcal{T}_{m \leftarrow j}(h_m, h_j)]_{j \neq m} \right), \tag{23}$$

Η τελική συνενωμένη αναπαράσταση είναι η συνένωση όλων των διανυσμάτων $\tilde{h}_m$, που περνάει σε ένα prediction head:

$$\hat{y} = \text{MLP} \left( \left[ \tilde{h}_1, \ldots, \tilde{h}_M \right] \right), \tag{24}$$

και ο στόχος μάθησης δίνεται από το Λ1 λοσς:

$$\mathcal{L}_{\text{task}} = \|\hat{y} - y\|_1, \tag{25}$$

όπου οι λείπουσες τροπικότητες είτε αντικαθίστανται με ανακατασκευασμένες εξόδους από τον αποκωδικοποιητή είτε μηδενίζονται κατά την εκπαίδευση.

**Συνολική απώλεια (Loss) του Σταδίου 2:** Στο στάδιο 2 της εκπαίδευσης (βλέπε Σχήμα 3) η συνολική απώλεια αποτελεί σταθμισμένο άθροισμα του loss του αποκωδικοποιητή ευθυγράμμισης και του loss της εργασίας:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{align}} \tag{26}$$

Στις δοκιμές μας επιλέξαμε $\lambda = 0.2$.

## 0.5 Πειραματική Διάταξη

### 0.5.1 Σύνολα Δεδομένων Πολυτροπικής Αναγνώρισης Συναισθημάτων

Αξιολογούμε την προσέγγισή μας σε δύο ευρέως χρησιμοποιούμενα σύνολα δεδομένων πολυτροπικής αναγνώρισης συναισθημάτων (MER):

- **CMU-MOSI** [46]: Αποτελείται από 2.199 βιντεοκλίπ με reviews από το YouTube. Το σύνολο δεδομένων χωρίζεται σε 1.284 δείγματα εκπαίδευσης, 229 δείγματα επικύρωσης και 686 δείγματα δοκιμής.
- **CMU-MOSEI** [47]: Περιέχει 22.856 βιντεοκλίπ σε επίπεδο εκφωνήματος με σχολιασμό ετικετών συναισθήματος και συναισθηματικότητας. Η επίσημη διαίρεση περιλαμβάνει 16.326 δείγματα εκπαίδευσης, 1.871 δείγματα επικύρωσης και 4.659 δείγματα δοκιμής.

### 0.5.2 Εξαγωγή Χαρακτηριστικών

Χρησιμοποιούμε εργαλεία προ-επεξεργασίας ειδικά για κάθε τροπικότητα ως εξής:

- **Κείμενο**: Χρησιμοποιούμε την τελική κρυφή κατάσταση ενός προ-εκπαιδευμένου μοντέλου BERT [52] για την εξαγωγή ενσωματώσεων λέξεων 768 διαστάσεων.
- **Ακουστική**: Εξάγουμε ακουστικά χαρακτηριστικά 74 διαστάσεων χρησιμοποιώντας το εργαλειοθήκη COVAREP [53], καταγράφοντας τον τόνο, παραμέτρους γλωττιδικής πηγής και άλλα προσωδιακά χαρακτηριστικά.
- **Όραση**: Εξάγουμε 35 χαρακτηριστικά έκφρασης προσώπου από κάθε καρέ χρησιμοποιώντας το εργαλειοθήκη Facet [54].

### 0.5.3 Μετρικές Αξιολόγησης

Ακολουθούμε προηγούμενες εργασίες [1] και αναφέρουμε τρεις τυπικές μετρικές:

- **Ακρίβεια 2-κλάσεων ($\mathbf{ACC_2}$)**: Κατηγοριοποίηση 2 κλάσεων συναισθήματος (θετικό/αρνητικό).
- **Ακρίβεια 7-κλάσεων ($\mathbf{ACC_7}$)**: Λεπτομερής κατηγοριοποίηση σε 7 διατεταγμένες κατηγορίες συναισθήματος.

- **Βαθμολογία F1**: Μακρο-μέση βαθμολογία F1 στη ταξινόμιση 2 κλάσεων.

.

### 0.5.4 Πρωτόκολλο Ελλιπών και Σταθερών Δεδομένων

Στο πλαίσιο του χειρισμού ελλιπών πολυτροπικών δεδομένων, χρησιμοποιούνται δύο κύρια πρωτόκολλα:

- **Πρωτόκολλο Σταθερών Ελλιπών Δεδομένων (Fixed Missing Protocol)**: Υπό αυτό το πρωτόκολλο, ένα συνεπές σύνολο μίας ή δύο τροπικοτήτων απορρίπτεται σκόπιμα σε όλα τα δείγματα του συνόλου δεδομένων. Για παράδειγμα, τα πειράματα μπορεί να διεξάγονται όπου:
  - Μία τροπικότητα λείπει (π.χ., μόνο γλώσσα, μόνο όραση, ή μόνο ακουστικά δεδομένα είναι διαθέσιμα).
  - Δύο τροπικότητες λείπουν (π.χ., μόνο γλώσσα και όραση, γλώσσα και ακουστικά, ή όραση και ακουστικά δεδομένα είναι διαθέσιμα).

  Αυτό εξασφαλίζει ένα προκαθορισμένο μοτίβο ελλείψεων σε όλη την πειραματική διάταξη.

- **Πρωτόκολλο Τυχαίων Ελλιπών Δεδομένων (Random Missing Protocol)**: Αυτό το πρωτόκολλο εισάγει μεταβλητότητα τυχαιοποιώντας τα μοτίβα ελλείψεων για κάθε μεμονωμένο δείγμα. Συνεπώς, για οποιοδήποτε δεδομένο δείγμα, είτε μία είτε δύο τροπικότητες μπορεί να απουσιάζουν. Ο βαθμός ελλείψεων σε αυτό το πρωτόκολλο ποσοτικοποιείται από τον Ρυθμό Ελλείψεων (MR), που ορίζεται ως:

$$MR = 1 - \frac{\sum_{i=1}^{N} m_i}{N \times M}$$

  όπου το $m_i$ αντιπροσωπεύει τον αριθμό των διαθέσιμων τροπικοτήτων για το $i$-οστό δείγμα, το $N$ είναι ο συνολικός αριθμός δειγμάτων, και το $M$ είναι ο συνολικός αριθμός τροπικοτήτων (σε αυτή την περίπτωση, $M = 3$). Είναι κρίσιμος περιορισμός ότι τουλάχιστον μία τροπικότητα πρέπει να είναι πάντα διαθέσιμη για κάθε δείγμα ($m_i \geq 1$), που σημαίνει ότι ο μέγιστος δυνατός Ρυθμός Ελλείψεων είναι $\frac{M-1}{M}$. Για πειράματα με τρεις τροπικότητες, οι τιμές MR επιλέχθηκαν από το σύνολο $\{0.0, 0.3, 0.5, 0.7\}$. Ο επιλεγμένος MR διατηρείται συνεπώς στις φάσεις εκπαίδευσης, επικύρωσης και δοκιμής για να εξασφαλιστεί δίκαιη αξιολόγηση.

### 0.5.5 Λεπτομέρειες Υλοποίησης

Για όλα τα πειράματα χρησιμοποιήσαμε τον Adam optimizer [55] με ρυθμό μάθησης $\lambda = 0.002$ και παρακμή βάρους $\beta = 0.005$. Επιπλέον, χρησιμοποιήθηκε πρόωρη διακοπή (early stopping) με $patience = 10$ και σημεία ελέγχου μοντέλου (model checkpoints) για τα αποτελέσματα της εκπαίδευσης σταδίου 2. Όλα τα εξαγόμενα χαρακτηριστικά τροπικοτήτων προβάλλονται σε έναν κοινό χώρο χαρακτηριστικών μέσω ενός ρηχού κωδικοποιητή[3] με διάσταση καναλιού $d = 32$ και μήκος ακολουθίας $T = 48$. Για το μέρος της διάχυσης (diffusion) επιλέξαμε να σχεδιάσουμε όλα τα score networks να έχουν το ίδιο μέγεθος εκπαιδεύσιμων παραμέτρων για δίκαιη σύγκριση:

- **UNet με Διασταυρούμενη Προσοχή (Cross Attention)**: Αρχιτεκτονική U-Net με διασταυρούμενη προσοχή υποβιβασμού, με κωδικοποιητή-αποκωδικοποιητή 4 επιπέδων με κανάλια $[32, 64, 128, 256]$, διάσταση ενσωμάτωσης χρόνου $d_{emb} = 256$ με στρωματικά πυκνά στρώματα για συγχώνευση, και μπλοκ Διασταυρούμενης Προσοχής Κωδικοποιητή Μετασχηματιστή (Transformer Encoder Cross Attention blocks) με 2 στρώματα και 8 κεφάλια προσοχής ανά επίπεδο.

- **Πολυτροπικός Μετασχηματιστής Διάχυσης (Multimodal Diffusion Transformer)**: Αρχιτεκτονική Μετασχηματιστή με στοιχειακή Γραμμική Διαμόρφωση (FiLM) υποβιβασμού, διάσταση μοντέλου $d_{model} = 256$, βάθος 6 στρωμάτων μετασχηματιστή, 8 κεφάλια προσοχής, διάσταση MLP $d_{mlp} = 512$, και διάσταση υποβιβασμού $d_{cond} = 256$ και διάσταση ενσωμάτωσης χρόνου $d_{time} = 128$.

- **Μετασχηματιστής Διάχυσης (Diffusion Transformer)**: Μετασχηματιστής Διάχυσης με διάσταση μοντέλου $d_{model} = 256$, βάθος 6 στρωμάτων μετασχηματιστή, 8 κεφάλια προσοχής, διάσταση MLP $d_{mlp} = 512$, και διάσταση υποβιβασμού $d_{cond} = 256$ που περιέχει επίσης τις πληροφορίες χρονικού βήματος (timestep information).

- **ScoreTransformer1D**: 1D Μετασχηματιστής με διάσταση μοντέλου $d_{model} = 256$, βάθος 6 στρωμάτων, 8 κεφάλια προσοχής, διάσταση MLP $d_{mlp} = 512$, και διάσταση ενσωμάτωσης χρόνου $d_{time} = 256$.

Επιπρόσθετα, τα δίκτυα βαθμολογίας εκπαιδεύτηκαν για 50 εποχές μέγιστο[4]. Στη συνέχεια, ο αποκωδικοποιητής ευθυγράμμισης $D_m$ που χρησιμοποιείται για περαιτέρω βελτίωση των διαχεόμενων τροπικοτήτων έχει 20 μπλοκ RCAB με μείωση 16. Όλοι οι κατάντη ταξινομητές συγχώνευσης $T_k$ είναι προεπιλεγμένοι Κωδικοποιητές Μετασχηματιστή PyTorch (PyTorch Transformer Encoders) με 4 στρώματα, 8 κεφάλια και *attention dropout* $= 0.2$.

[3] Ένας μονός στρώμα 1D συνελικτικός πυρήνας με μέγεθος 3.

[4] εάν δεν συνέβη πρόωρη διακοπή (early stopping)

## 0.6 Πειραματικά Αποτελέσματα

Υιοθετούμε μια στρατηγική διαδοχικής βελτιστοποίησης για να εντοπίσουμε τη βέλτιστη διαμόρφωση. Αξιολογούμε συστηματικά: (1) διατυπώσεις ΣΔΕ (VP έναντι VE) με τον προεπιλεγμένο δειγματολήπτη διασταυρούμενης προσοχής U-Net και Euler-Maruyama, (2) αρχιτεκτονικές οπισθίου άκρου ξανά με τον ίδιο προεπιλεγμένο δειγματολήπτη βάσης, και (3) μεθόδους δειγματοληψίας, επιλέγοντας την καλύτερα αποδίδουσα επιλογή σε κάθε στάδιο πριν προχωρήσουμε στην επόμενη αξιολόγηση. Σε κάθε στάδιο θα συγκρίνουμε την απόδοση του μοντέλου με το μοντέλο βάσης IMDER στο οποίο βασίζεται η εργασία μας.

### 0.6.1 Στάδιο 1: Σύγκριση Διατύπωσης ΣΔΕ

Πρώτα συγκρίνουμε τις δύο διατυπώσεις ΣΔΕ χρησιμοποιώντας μια διαμόρφωση βάσης (base configuration) για να καθορίσουμε ποια παρέχει καλύτερη απόδοση για εργασίες πολυτροπικής αναγνώρισης συναισθημάτων.

#### Αρχική Διαμόρφωση

Για αυτή τη σύγκριση, χρησιμοποιούμε:

- **Αρχιτεκτονική ραχοκοκαλιάς**: U-Net with Cross Attention
- **Δειγματολήπτης**: Euler-Maruyama με 100 NFEs
- **Πρωτόκολλα Ελλείψεων**: Σταθερά και τυχαία μοτίβα ελλείψεων
- **Σύνολα Δεδομένων**: CMU-MOSI

**Table 1.** *Σύγκριση Διατυπώσεων ΣΔΕ στο Σύνολο Δεδομένων CMU-MOSI*

| Available Modalities | Variance Exploding (VE) | | | Variance Preserving (VP) | | | Vanilla IMDER | | |
|---|---|---|---|---|---|---|---|---|---|
| | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ |
| Language | 84.9 | 84.9 | 45.3 | **85.3** | **85.3** | **45.6** | 84.8 | 84.7 | 44.8 |
| Acoustic | 60.4 | 60.3 | **18.7** | **62.0** | **62.1** | 17.7 | 61.3 | 60.8 | 20.5 |
| Vision | 58.1 | 58.3 | **19.1** | **58.6** | **58.8** | 18.3 | 61.0 | 61.2 | 21.0 |
| Language + Acoustic | 85.6 | 85.4 | **46.7** | **86.4** | **86.3** | 45.4 | 85.4 | 85.3 | 45.0 |
| Language + Vision | 85.5 | 85.4 | **45.6** | **86.1** | **85.9** | 45.0 | 85.5 | 85.4 | 45.3 |
| Acoustic + Vision | **60.6** | 59.3 | 19.5 | 59.6 | **59.7** | **21.8** | 62.0 | 62.1 | 20.2 |
| **Average** | 72.5 | 72.2 | **32.4** | **73.0** | **73.0** | 32.3 | 73.3 | 73.2 | 32.8 |

**Table 2.** *Διατυπώσεις SDE υπό Πρωτόκολλο Τυχαίων Ελλείψεων (CMU-MOSI). Για κάθε πείραμα που αναφέρεται στον παρακάτω πίνακα, εκτελέσαμε το μοντέλο με 5 διαφορετικούς τυχαίους σπόρους στο σύνολο δοκιμής και υπολογίσαμε τον μέσο όρο των αποτελεσμάτων για πιο ισχυρές μετρικές.*

| Missing Rate | Variance Exploding (VE) | | | Variance Preserving (VP) | | | Vanilla IMDER | | |
|---|---|---|---|---|---|---|---|---|---|
| | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ |
| MR = 0.3 | 80.6 | 80.7 | **41.8** | **81.0** | **81.0** | 41.5 | 79.9 | 79.6 | 39.1 |
| MR = 0.5 | 73.6 | 72.5 | **33.7** | **74.6** | **73.2** | 33.5 | 74.0 | 73.8 | 34.2 |
| MR = 0.7 | 69.5 | 69.7 | 29.5 | **70.2** | **69.9** | **30.2** | 70.8 | 70.3 | 31.6 |
| **Average** | 74.5 | 74.3 | **35.0** | **75.2** | **74.7** | 35.0 | 74.9 | 74.6 | 34.9 |

### Ανάλυση Διατυπώσεων SDE στο CMU-MOSI

Οι Πίνακες 1,2 παρουσιάζει μια σύγκριση μεταξύ των διατυπώσεων ΣΔΕ (VE) και (VP) σε διάφορα σενάρια ελλιπών τροπικοτήτων στο σύνολο δεδομένων CMU-MOSI. Οι μετρικές που αναφέρονται περιλαμβάνουν την ακρίβεια 2-κατηγοριών ($ACC_2$), τη βαθμολογία F1, και την ακρίβεια 7-κατηγοριών ($ACC_7$).

**Η VP Γενικά Υπερτερεί της VE.** Σε όλα τα μοτίβα ελλιπών τροπικοτήτων, η διατύπωση VP παράγει συστηματικά υψηλότερες βαθμολογίες $ACC_2$ και F1 σε σύγκριση με την VE. Για παράδειγμα, όταν οι διαθέσιμες τροπικότητες είναι η *Γλώσσα + Ακουστική*, η VP επιτυγχάνει 86.4 $ACC_2$ και 86.3 F1, υπερτερώντας της VE που σημειώνει 85.6 και 85.4, αντίστοιχα. Παρόμοια τάση ισχύει στη διαμόρφωση *Γλώσσα + Όραση*.

**Απόδοση όταν Ακουστικά και Όραση είναι Διαθέσιμα.** Και οι δύο διατυπώσεις δείχνουν σημαντικά χαμηλότερη απόδοση όταν είναι διαθέσιμη μόνο η ακουστική ή η οπτική τροπικότητα. Αυτό αντικατοπτρίζει την κυριαρχία της γλωσσικής τροπικότητας σε εργασίες πρόβλεψης συναισθήματος στο CMU-MOSI, όπως έχει παρατηρηθεί σε προηγούμενες εργασίες. Επίσης, αξιοσημείωτο είναι ότι όταν είναι διαθεσιμές οι τροπικότητες ακουστικής και οράσεως, το μοντέλο δεν αποδίδει πολύ καλύτερα και στην περίπτωση της VP αποδίδει ελαφρώς χειρότερα από το να έχει μόνο την ακουστική τροπικότητα σε $ACC_2$ και F1.

**Τα Αποτελέσματα $ACC_7$** Η ακρίβεια 7-κατηγοριών ($ACC_7$) δείχνει περισσότερη μεταβλητότητα. Η VE υπερτερεί ελαφρώς της VP σε ορισμένες περιπτώσεις (π.χ., *Γλώσσα + Ακουστική* και *Μόνο-Όραση*), ενώ η VP αποδίδει καλύτερες βαθμολογίες σε άλλες (π.χ., *Ακουστική + Όραση*). Αυτά τα αποτελέσματα υποδεικνύουν ότι η VE μπορεί περιστασιακά να διατηρεί ικανότητες πρόβλεψης πιο λεπτομερών κατηγοριών, αν και οι διαφορές είναι

μικρές.

**Η Πολυτροπική Συγχώνευση Οδηγεί στην Ισχυρότερη Απόδοση.** Τα καλύτερα συνολικά αποτελέσματα παρατηρούνται όταν η γλώσσα συγχωνεύεται με μια άλλη τροπικότητα, επιδεικνύοντας το όφελος της πολυτροπικής μάθησης. Σε αυτές τις περιπτώσεις, η VP παραμένει η πιο αξιόπιστη διατύπωση.

**Μέση Απόδοση.** Κατά μέσο όρο, η VP επιτυγχάνει ελαφρώς καλύτερη διχοτομική ακρίβεια και βαθμολογία F1 (73.0 και για τις δύο) σε σύγκριση με την VE (72.5 $ACC_2$, 72.2 F1), ενώ η VE υπερτερεί ελαφρώς της VP στην $ACC_7$ (32.4 έναντι 32.3). Οι διαφορές, ωστόσο, είναι **οριακές**.

**Συμπέρασμα.** Συνολικά, η VP προσφέρει πιο συνεπή και ισχυρή απόδοση σε σενάρια ελλιπών τροπικοτήτων, ιδιαίτερα σε μετρικές ταξινόμισης και F1. Ενώ η VE μπορεί να διατηρεί ελαφρά πλεονεκτήματα στη λεπτομερή κατηγοριοποίηση σε επιλεγμένες περιπτώσεις, η VP είναι γενικά πιο αποτελεσματική για πολυτροπική ανάλυση συναισθήματος στο CMU-MOSI.

### 0.6.2 Στάδιο 2: Σύγκριση Αρχιτεκτονικών για ύπο συνθήκη παραγωγή

Χρησιμοποιώντας τη βέλτιστη ΣΔΕ από το Στάδιο 1, συγκρίνουμε διαφορετικές αρχιτεκτονικές ραχοκοκαλιάς και μηχανισμούς ύπο συνθήκη παραγωγής.

#### Πειραματική Διαμόρφωση

Για αυτή τη σύγκριση, χρησιμοποιούμε:

- **Επιλεγμένη ΣΔΕ**: Variance Preserving (VP) ΣΔΕ
- **Δειγματολήπτης**: Euler-Maruyama με 100 NFEs
- **Αρχιτεκτονικές ραχοκοκαλιάς**: U-Net με Cross-Attention, Multimodal Diffusion Transformer με συνένωση και στρώματα τύπου FiLM, Diffusion Transformer με AdaLN και ScoreTransformer1D με απλή συνένωση
- **Πρωτόκολλα Ελλείψεων**: Σταθερό μοτίβο ελλείψεων

### Αναλύση Αποτελέσματων για αρχιτεκτονικές ύπο συνθήκη παραγωγής

Από τον Πίνακα 3, παρατηρούμε ότι οι διαφορές απόδοσης μεταξύ των conditioning backbones είναι μικρές, αλλά αναδύονται συνεπείς τάσεις. Για μονοτροπικές περιπτώσεις, ο Di-Transformer αποδίδει ελαφρώς καλύτερα σε εργασίες μόνο-γλώσσας, ενώ το U-Net παραμένει ισχυρό για ακουστικές εισόδους. Αξίως προσοχής, ο MMDi-Transformer επιτυγχάνει την υψηλότερη $ACC_7$ όταν ανακτά ακουστικές ή οπτικές τροπικότητες, υποδηλώνοντας ότι ο υποβιβασμός τύπου FiLM παρέχει πλεονεκτήματα για λεπτομερή κατηγοριοποίηση. Ο ScoreTransformer1D, ωστόσο, ισοφαρίζει ή υπερτερεί των ανταγωνιστών σε αρκετές ρυθμίσεις και αποκτά τη βέλτιστη μέση $ACC_2$ και F1 σε μοτίβα ελλείψεων.

Ο Πίνακας 4 δείχνει πως η πολυπλοκότητα των πιο βαριών αρχιτεκτονικών δεν αποτυπώνεται κατάνάγκη και στην στην αποδοτικότητα τους. Παρά το γεγονός ότι έχει τις λιγότερες παραμέτρους, το U-Net backbone είναι πάνω από δύο φορές πιο αργό στην εμπρόσθια τροφοδότηση σε σύγκριση με τις εναλλακτικές βασισμένες σε μετασχηματιστές.

Αντίθετα, ο προτεινόμενος ScoreTransformer1D δεν είναι μόνο το πιο αποδοτικό μοντέλο σε παραμέτρους (3.2M παράμετροι) αλλά επίσης επιτυγχάνει τον ταχύτερο χρόνο εξαγωγής συμπερασμάτων (13.1 ms), πάνω από 5× ταχύτερος από το U-Net, διατηρώντας παράλληλα ανταγωνιστική ακρίβεια. Αυτό τον καθιστά ιδιαίτερα ελκυστικό για ανάπτυξη σε πολυτροπικές εφαρμογές ευαίσθητες στο χρόνο.

**Table 3.** *Συγκρίση αρχιτεκτονικών για υποσυνθήκη παραγωγή στο Σύνολο Δεδομένων CMU-MOSI*

| Available Modalities | U-Net Cross-Attn | | | Di-Transformer | | | MMDi-Transformer | | | ScoreTransformer1D | | | Vanilla IMDER | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ |
| Language | 85.3 | 85.3 | 45.6 | **86.1** | **86.0** | 45.4 | 84.4 | 84.4 | **46.6** | 85.6 | 85.5 | 45.3 | 84.8 | 84.7 | 44.8 |
| Acoustic | **62.0** | **62.1** | 17.7 | 61.2 | 61.1 | 18.8 | 61.8 | 60.9 | **20.9** | 61.0 | 60.0 | 20.0 | 61.3 | 60.8 | 20.5 |
| Vision | 58.6 | 58.8 | 18.3 | 59.6 | 58.5 | 17.4 | 60.2 | 59.8 | **18.6** | **61.1** | **60.8** | 17.6 | 61.0 | 61.2 | 21.0 |
| Language + Acoustic | **86.4** | **86.3** | 45.4 | 86.0 | 85.9 | **46.2** | 85.6 | 85.5 | 45.0 | 85.5 | 85.4 | 45.0 | 85.4 | 85.3 | 45.0 |
| Language + Vision | 86.1 | 85.9 | 45.0 | 86.0 | 85.9 | **47.8** | 85.3 | 85.3 | 46.0 | **86.4** | **86.3** | 46.3 | 85.5 | 85.4 | 45.3 |
| Acoustic + Vision | 59.6 | 59.7 | **21.8** | 61.0 | 60.4 | 19.7 | 61.2 | **61.3** | 19.5 | **61.3** | 60.4 | 19.5 | 62.0 | 62.1 | 20.2 |
| **Average** | 73.0 | 73.0 | 32.3 | 73.3 | 72.9 | 32.6 | 73.1 | 72.9 | **32.8** | **73.5** | **73.1** | 32.3 | 75.1 | 75.0 | 34.5 |

**Table 4.** *Ανάλυση Αποδοτικότητας Μοντέλου: Ο πίνακας αυτός παρουσιάζει τα μεγέθη και τον χρόνο προώθησης και αριθμό FLOPs (Floating point operations) για μια εμπρόσθια τροφοδότηση μέσα από ένα δίκτυο βαθμίδας score network μίας τροπικότητας. Αξιοσημείωτο είναι ότι, παρόλο που το Unet διαθέτει πολύ λιγότερες παραμέτρους σε σύγκριση με τους Transformer ανταγωνιστές του, ο χρόνος προώθησης είναι υπερδιπλάσιος σε σχέση με τον χειρότερο από αυτούς.*

| Architecture | Parameters (M) | Training Time (hrs) | Inference Time (ms) | FLOPs (G) | Memory (MB) |
|---|---|---|---|---|---|
| U-Net Cross-Attention | 3.5 | 0.55 | 72.1 | 0.66 | 13.1 |
| Multimodal Di-Transformer | 8.8 | 0.52 | 31.2 | 1.21 | 33.7 |
| Di-Transformer | 9.3 | 0.45 | 24.6 | **0.53** | 35 |
| ScoreTransformer1D (Ours) | **3.2** | **0.30** | **13.1** | 1.04 | **12.5** |

### 0.6.3 Στάδιο 3: Σύγκριση Αλγορίθμων Δειγματοληψίας

Χρησιμοποιώντας τον βέλτιστο συνδυασμό ΣΔΕ-ραχοκοκαλιάς από τα προηγούμενα στάδια, αξιολογούμε διαφορετικούς αλγόριθμους δειγματοληψίας για να βρούμε την καλύτερη ισορροπία μεταξύ ταχύτητας και ποιότητας.

#### Πειραματική Διαμόρφωση

Για αυτήν τη σύγκριση, χρησιμοποιούμε:

- **Επιλεγμένη Διαμόρφωση**: VP ΣΔΕ + Ραχοκοκαλιά Score Transformer
- **Δειγματολήπτες**: Euler-Maruyama, Predictor-Corrector, Heun, DDIM
- **Εύρος NFE**: 10-100 αξιολογήσεις συναρτήσεων
- **Αξιολόγηση**: Ισορροπία απόδοσης έναντι ταχύτητας

#### Ανάλυση Ισορροπίας Ταχύτητας-Ποιότητας

**Table 5.** *Σύγκριση Αλγορίθμων Δειγματοληψίας στο Σύνολο Δεδομένων CMU-MOSI, η απόδοση είναι ο μέσος όρος του πρωτοκόλλου σταθερής έλλειψης για κάθε διαμόρφωση δειγματολήπτη.*

| Sampler | NFEs | $ACC_2$ | F1 | $ACC_7$ | Sampling Time (s) |
|---|---|---|---|---|---|
| Vanilla IMDER | 100 | 73.3 | 73.2 | 32.8 | 1.17 |
| Euler-Maruyama | 100 | 73.5 | 73.1 | 32.3 | 1.17 |
| Euler-Maruyama | 80 | 72.9 | 72.6 | 32.9 | 0.95 |
| Euler-Maruyama | 50 | 73.3 | 73.1 | 33.3 | 0.61 |
| Predictor-Corrector | 100 | 72.9 | 72.7 | 33.2 | 1.17 |
| Predictor-Corrector | 80 | 72.9 | 72.8 | 32.9 | 0.95 |
| Predictor-Corrector | 60 | 72.8 | 72.9 | **33.4** | 0.71 |
| Heun | 80 | 73.4 | 73.1 | 32.9 | 0.95 |
| Heun | 60 | 73.4 | **73.3** | **33.4** | 0.71 |
| Heun | 40 | 72.8 | 72.5 | 32.9 | 0.49 |
| DDIM | 30 | **73.8** | 72.5 | 33.3 | 0.37 |
| DDIM | 20 | 72.4 | 72.4 | 32.1 | 0.24 |
| DDIM | 10 | 72.9 | 72.8 | 33.0 | **0.12** |

Με βάση τα αποτελέσματα στον Πίνακα 5, τα οποία αποτελούν συντομογραφία από τον Πίνακα **;;**, μπορούμε να αναλύσουμε την ισορροπία απόδοσης-αποδοτικότητας μεταξύ των διαφόρων διαμορφώσεων δειγματοληψίας.

**Ανάλυση Απόδοσης.** Ο δειγματολήπτης Euler-Maruyama με 100 NFEs επιτυγχάνει παραδόξως τη δεύτερη υψηλότερη βαθμολογία $ACC_2$ (73.5) και F1 (73.1), πίσω από τον δειγματολήπτη DDIM με 30 NFEs για την πρώτη και τον δειγματολήπτη Heun με 60 NFEs για τη δεύτερη. Ο **DDIM** με 30 NFEs προσφέρει ανταγωνιστική απόδοση ($ACC_2$: 73.8, F1: 72.5) ενώ είναι σημαντικά ταχύτερος (0.37 έναντι 1.17). Ο δειγματολήπτης **Heun** με 60 NFEs παρέχει εξαιρετική απόδοση ($ACC_2$: 73.4, F1: 73.3, $ACC_7$: 33.4) με μέτρια ταχύτητα (0.71s).

**Ανάλυση Ταχύτητας.** Ο **DDIM** επιδεικνύει ανώτερη ισορροπία ταχύτητας-ποιότητας, επιτυγχάνοντας απόδοση κοντά στη βασική με 3 φορές ταχύτερη εξαγωγή συμπερασμάτων (inference). Οι μέθοδοι **Predictor-Corrector** παρουσιάζουν ελάχιστα κέρδη απόδοσης σε σχέση με απλούστερες προσεγγίσεις, διατηρώντας παράλληλα υψηλότερο υπολογιστικό κόστος. Ο δειγματολήπτης **Heun** με 60 NFEs διατηρεί ισχυρή απόδοση ($ACC_2$: 73.4) με βελτίωση ταχύτητας σχεδόν 1.66 φορές.

**Table 6.** *Ανάλυση Ισορροπίας Ταχύτητας-Ποιότητας, εδώ συγκρίνουμε τις πιο αντιπροσωπευτικές διαμορφώσεις δειγματοληψίας.*

| Sampler Configuration | Relative Speed | $ACC_2$ Change | Sampling Time (s) | Recommended Use |
|---|---|---|---|---|
| Euler-Maruyama (100 NFEs) | 1.0× | 73.5 (baseline) | 1.17 | High-quality baseline |
| Euler-Maruyama (50 NFEs) | 1.9× | 73.3 (-0.3%) | 0.61 | Balanced quality-speed |
| Heun (60 NFEs) | 1.6× | **73.4 (-0.1%)** | 0.71 | **Fast with quality retention** |
| DDIM (30 NFEs) | 3.2× | **73.8 (+0.4%)** | 0.37 | **Optimal speed choice** |
| DDIM (10 NFEs) | 9.8× | 72.9 (-0.8%) | **0.12** | Ultra-fast deployment |

### Τελικές Προτάσεις Διαμόρφωσης

Βάσει της ολοκληρωμένης αξιολόγησής μας σε όλα τα τρία στάδια, προτείνουμε δύο βέλτιστες διαμορφώσεις:

### Διαμόρφωση Εστιασμένη στην Ποιότητα.

- **ΣΔΕ**: Variance Preserving (VP)
- **Αρχιτεκτονική Ραχοκοκαλιάς**: ScoreTransformer1D (3.2M παράμετροι)
- **Δειγματολήπτης**: Δειγματολήπτης Heun με 60 NFEs
- **Απόδοση**: 73.4 $ACC_2$, 73.3 F1, 33.4 $ACC_7$
- **Ταχύτητα**: 0.71s χρόνος δειγματοληψίας

- **Περίπτωση Χρήσης**: Εφαρμογές που απαιτούν μέγιστη ακρίβεια με αποδεκτό χρόνο εξαγωγής συμπερασμάτων

**Διαμόρφωση Βελτιστοποιημένη ως προς την Ταχύτητα.**

- **ΣΔΕ**: Variance Preserving (VP)
- **Αρχιτεκτονική Ραχοκοκαλιάς**: ScoreTransformer1D (3.2M παράμετροι)
- **Δειγματολήπτης**: DDIM με 30 NFEs
- **Απόδοση**: 73.8 $ACC_2$, 72.5 F1, 33.3 $ACC_7$
- **Ταχύτητα**: 0.37s χρόνος δειγματοληψίας (3.2 φορές ταχύτερος από τη βασική διαμόρφωση)
- **Περίπτωση Χρήσης**: Εφαρμογές πραγματικού χρόνου και σενάρια ανάπτυξης όπου η ταχύτητα είναι κρίσιμη

**Βασικές Διαπιστώσεις**

**Υπεροχή του DDIM.** Ο DDIM με 30 NFEs αναδεικνύεται ως η βέλτιστη επιλογή για γρήγορη εξαγωγή συμπερασμάτων (downstream inference), επιτυγχάνοντας στην πραγματικότητα ελαφρώς καλύτερη απόδοση $ACC_2$ από τη βασική διαμόρφωση, ενώ είναι σημαντικά ταχύτερος. Αυτό το αντιφατικό αποτέλεσμα υποδηλώνει ότι η ντετερμινιστική φύση της δειγματοληψίας DDIM μπορεί να παρέχει καλύτερες ιδιότητες σύγκλισης για το πολυτροπικό μας πλαίσιο διάχυσης.

**Φθίνουσες Αποδόσεις.** Πέρα από τα 60 NFEs, οι βελτιώσεις στην απόδοση είναι οριακές, ενώ το υπολογιστικό κόστος αυξάνεται σημαντικά. Αυτή η παρατήρηση ευθυγραμμίζεται με πρόσφατα ευρήματα στη βιβλιογραφία των μοντέλων διάχυσης που υποδηλώνουν ότι λιγότερα βήματα δειγματοληψίας μπορεί να είναι επαρκή για πολλές πρακτικές εφαρμογές.

### 0.6.4 Σύγκριση με τις Μεθόδους Αιχμής

Χρησιμοποιώντας τις δύο βέλτιστες διαμορφώσεις μας που εντοπίστηκαν στο Στάδιο 3, συγκρίνουμε με υπάρχουσες μεθόδους ελλειπούς πολυτροπικής ανάκτησης, συμπεριλαμβανομένης της αρχικής μεθόδου που βασιστίκαμε Vanilla IMDER.

**Table 7.** *Performance comparison under both **Random Missing Protocol** and **Fixed Missing Protocol** on CMU-MOSI and CMU-MOSEI datasets. Each cell reports $ACC_2$ / F1 / $ACC_7$. Baseline results for DCCA [27], DCCAE [28], MCTN [29], MMIN [6], and GCNet [30] are taken from prior work [1]. Our **Quality-Optimized** configuration uses VP SDE + ScoreTransformer1D + Heun (60 NFEs), while **Speed-Optimized** uses VP SDE + ScoreTransformer1D + DDIM (30 NFEs). **Bolded values** indicate the best score per metric.*

**(a)** *Results under the **Random Missing Protocol** at various missing rates (MR). We report the average over 5 random seeds for each missing rate case for a more robust result.*

| Dataset | MR | DCCA | DCCAE | MCTN | MMIN | GCNet | Vanilla IMDER | Quality-Opt | Speed-Opt |
|---|---|---|---|---|---|---|---|---|---|
| **MOSI** | 0.0 | 75.3 / 75.4 / 30.5 | 77.3 / 77.4 / 31.2 | 81.4 / 81.5 / 43.4 | 84.6 / 84.4 / 44.8 | 85.2 / 85.1 / 44.9 | 85.7 / 85.6 / 45.3 | **86.0** / **86.2** / 45.6 | **86.0** / **86.2** / 45.6 |
| | 0.3 | 68.4 / 67.8 / 25.1 | 70.2 / 69.5 / 25.8 | 73.9 / 74.0 / 33.4 | 76.3 / 75.7 / 34.5 | 77.4 / 76.9 / 35.7 | 79.9 / 79.6 / 39.1 | **81.0** / **80.8** / **40.7** | 80.3 / 79.8 / 39.9 |
| | 0.5 | 61.7 / 60.9 / 21.0 | 63.4 / 62.1 / 21.9 | 68.3 / 67.9 / 29.6 | 71.2 / 70.3 / 30.9 | 72.6 / 72.0 / 31.5 | 74.0 / 73.8 / **34.2** | **75.3** / **74.3** / 34.1 | 73.8 / 73.8 / 33.6 |
| | 0.7 | 55.2 / 53.8 / 18.2 | 56.7 / 55.0 / 19.3 | 61.5 / 61.2 / 26.1 | 64.1 / 62.4 / 27.3 | 65.8 / 64.9 / 28.0 | 70.8 / 70.3 / 31.6 | 70.4 / **70.5** / **32.1** | **71.0** / 70.5 / 31.8 |
| **Average** | - | 65.2 / 64.5 / 23.7 | 66.9 / 66.0 / 24.6 | 71.3 / 71.2 / 33.1 | 74.1 / 73.2 / 34.4 | 75.3 / 74.7 / 35.0 | 77.6 / 77.3 / 37.6 | **78.2** / **78.0** / **38.1** | 77.8 / 77.6 / 37.7 |
| **MOSEI** | 0.0 | 80.7 / 80.9 / 47.7 | 81.2 / 81.2 / 48.2 | 84.2 / 84.2 / 51.2 | 84.3 / 84.2 / 52.4 | 85.2 / 85.1 / 51.5 | 85.1 / 85.1 / 53.4 | **85.7** / **85.8** / 53.3 | **85.7** / **85.8** / 53.3 |
| | 0.3 | 75.1 / 74.2 / 44.0 | 76.3 / 75.6 / 44.3 | 78.6 / 78.3 / 47.1 | 79.7 / 79.2 / 48.3 | 80.6 / 80.0 / 48.5 | 80.2 / 79.7 / 50.1 | **80.8** / **80.4** / **50.3** | 79.9 / 80.0 / 50.1 |
| | 0.5 | 70.8 / 69.1 / 41.1 | 71.9 / 70.3 / 41.3 | 75.3 / 74.9 / 45.5 | 76.4 / 75.0 / 46.7 | 77.3 / 76.2 / 46.8 | 78.2 / 77.3 / 47.9 | **79.0** / **78.1** / **48.9** | 77.5 / 77.2 / 46.9 |
| | 0.7 | 66.3 / 64.2 / 38.0 | 67.2 / 65.0 / 38.7 | 70.4 / 70.1 / 43.2 | 71.5 / 70.6 / 44.5 | 72.7 / 71.7 / 44.9 | **74.2** / 73.2 / 46.0 | 73.2 / **73.6** / **46.4** | 73.2 / 73.1 / 45.3 |
| **Average** | - | 73.2 / 72.1 / 42.7 | 74.2 / 73.0 / 43.1 | 77.1 / 76.9 / 46.8 | 78.0 / 77.3 / 48.0 | 79.0 / 78.3 / 47.9 | 79.4 / 78.8 / 49.4 | **79.7** / **79.5** / **49.7** | 79.1 / 79.0 / 48.9 |

**(b)** *Results under the **Fixed Missing Protocol** for different modality subsets.*

| Dataset | Available Modalities | DCCA | DCCAE | MCTN | MMIN | GCNet | Vanilla IMDER | Quality-Opt | Speed-Opt |
|---|---|---|---|---|---|---|---|---|---|
| **MOSI** | {l} | 73.6 / 73.8 / 30.2 | 76.4 / 76.5 / 28.3 | 79.1 / 79.2 / 41.0 | 83.8 / 83.8 / 41.6 | 83.7 / 83.6 / 42.3 | 84.8 / 84.7 / 44.8 | **86.1** / **86.0** / 45.9 | 85.8 / 85.7 / **46.5** |
| | {v} | 47.7 / 41.5 / 16.6 | 52.6 / 51.1 / 17.1 | 55.0 / 54.4 / 16.3 | 57.0 / 54.0 / 15.5 | 56.1 / 55.7 / 16.9 | 61.3 / 60.8 / 20.5 | **61.2** / **61.1** / **22.3** | 61.4 / 59.6 / 18.9 |
| | {a} | 50.5 / 46.1 / 16.3 | 48.8 / 42.1 / 16.9 | 56.1 / 54.5 / 16.5 | 55.3 / 51.5 / 15.5 | 56.1 / 54.5 / 16.6 | 61.0 / **61.2** / 21.0 | 61.5 / 60.9 / 21.0 | **62.3** / 59.7 / **21.8** |
| | {l, v} | 74.9 / 75.0 / 30.3 | 76.7 / 76.8 / 30.0 | 81.1 / 81.2 / 42.1 | 83.8 / 83.9 / 42.0 | 84.3 / 84.2 / 43.4 | **85.5** / **85.4** / 45.3 | 85.0 / 85.0 / **46.0** | 85.0 / 85.0 / 45.6 |
| | {l, a} | 74.7 / 74.8 / 29.7 | 77.0 / 77.0 / 30.2 | 81.0 / 81.0 / 43.2 | 84.0 / 84.0 / 42.3 | 84.5 / 84.4 / 43.4 | 85.4 / 85.3 / 45.0 | 85.8 / 85.6 / **45.6** | **86.1** / **86.0** / 45.0 |
| | {v, a} | 50.8 / 46.4 / 16.6 | 54.0 / 52.5 / 17.4 | 57.5 / 57.4 / 16.8 | 60.4 / 58.5 / 19.5 | 62.0 / 61.9 / 17.2 | **62.0** / **62.1** / 20.2 | 61.0 / 61.2 / 19.5 | **62.0** / 59.2 / **22.1** |
| | {l, v, a} | 75.3 / 75.4 / 30.5 | 77.3 / 77.4 / 31.2 | 81.4 / 81.5 / 43.4 | 84.6 / 84.4 / 44.8 | 85.2 / 85.1 / 44.9 | 85.7 / 85.6 / 45.3 | **86.0** / **86.2** / **45.6** | **86.0** / **86.2** / **45.6** |
| **Average** | – | 63.9 / 61.9 / 20.0 | 66.1 / 64.8 / 24.4 | 70.2 / 69.9 / 31.3 | 72.7 / 71.4 / 31.6 | 73.1 / 72.8 / 32.1 | 75.1 / 75.0 / 34.5 | 75.2 / **75.1** / **35.1** | **75.5** / 74.5 / **35.1** |
| **MOSEI** | {l} | 78.5 / 78.7 / 46.7 | 79.7 / 79.5 / 47.0 | 82.6 / 82.8 / 50.2 | 82.3 / 82.4 / 51.4 | 83.0 / 83.2 / 51.2 | 84.3 / 84.2 / 52.7 | **85.6** / **85.5** / **53.1** | 83.5 / 83.7 / 51.9 |
| | {v} | 61.9 / 55.7 / 41.3 | 61.1 / 57.2 / 40.1 | 62.6 / 57.1 / 41.6 | 59.3 / 60.0 / 40.7 | 61.9 / 61.6 / 41.7 | 61.5 / **62.6** / 41.6 | **63.6** / **62.6** / **42.3** | 61.3 / 61.4 / 41.3 |
| | {a} | 62.0 / 50.2 / 41.1 | 61.4 / 53.8 / 40.9 | 62.7 / 54.5 / 41.4 | 58.9 / 59.5 / 40.4 | 60.2 / 60.3 / 41.1 | 61.6 / **61.5** / 41.3 | **63.3** / 60.6 / **41.4** | 62.3 / 61.3 / 40.5 |
| | {l, v} | 80.3 / 79.7 / 46.6 | 80.4 / 80.4 / 47.1 | 83.2 / 83.2 / 50.4 | 83.8 / 83.4 / 51.2 | 84.3 / 84.4 / 51.1 | 84.5 / 85.1 / 52.8 | 85.0 / 85.0 / **53.1** | **85.2** / **85.3** / 52.4 |
| | {l, a} | 79.5 / 79.2 / 46.7 | 80.0 / 80.0 / 47.4 | 83.5 / 83.3 / 50.7 | 83.7 / 83.3 / 52.0 | 84.3 / 84.4 / 51.3 | 85.1 / 85.1 / **53.1** | **85.5** / **85.5** / 52.9 | 84.4 / 83.8 / 52.3 |
| | {v, a} | 63.4 / 56.9 / 41.5 | 62.7 / 59.2 / 41.6 | 63.7 / 62.7 / 42.1 | 63.5 / 61.9 / 41.8 | 64.1 / 57.2 / 42.0 | 63.5 / 63.3 / **42.8** | **63.9** / **64.0** / **42.8** | 63.7 / 62.9 / 42.3 |
| | {l, v, a} | 80.7 / 80.9 / 47.7 | 81.2 / 81.2 / 48.2 | 84.2 / 84.2 / 51.2 | 84.3 / 84.2 / 52.4 | 85.2 / 85.1 / 51.5 | 85.1 / 85.1 / **53.4** | **85.7** / **85.8** / 53.3 | **85.7** / **85.8** / 53.3 |
| **Average** | – | 72.3 / 68.8 / 44.5 | 72.4 / 70.2 / 44.6 | 74.6 / 72.5 / 46.8 | 73.7 / 73.5 / 47.1 | 74.7 / 73.7 / 47.1 | 75.1 / 75.3 / 48.2 | **76.1** / **75.6** / **48.4** | 75.2 / 74.9 / 47.7 |

**Ανάλυση Αποτελεσμάτων.** Τα πειραματικά αποτελέσματα δείχνουν ότι οι βέλτιστες διαμορφώσεις μας αποδίδουν καλύτερα ή συγκρίσιμα με τις μεθόδους αιχμής, τόσο στο random missing protocol όσο και στο fixed missing protocol. Η Quality-Optimized διαμόρφωση επιτυγχάνει τις υψηλότερες επιδόσεις με αυξημένη αποδοτικότητα, ενώ η Speed-Optimized διαμόρφωση προσφέρει ανταγωνιστικά αποτελέσματα με μικρότερο υπολογιστικό κόστος. Στα δύο σύνολα δεδομένων (CMU-MOSI, CMU-MOSEI), παρατηρούνται σταθερά βελτιωμένες μετρικές σε σχέση με το Vanilla IMDER, ακόμα και σε δύσκολες περιπτώσεις όπως όταν είναι διαθέσιμη μόνο η οπτική ή η ακουστική πληροφορία.

**Υπολογιστική Αποδοτικότητα και Συνεισφορές.** Οι προτεινόμενες διαμορφώσεις παρέχουν σημαντική επιτάχυνση, με την Speed-Optimized να είναι έως και **15**× ταχύτερη από τη βασική γραμμή, γεγονός που τις καθιστά κατάλληλες για εφαρμογές πραγματικού

χρόνου. Οι αρχιτεκτονικές μας επιλογές η διατύπωση VP SDE, το υπόβαθρο ScoreTransformer1D, και οι δειγματολήπτες DDIM/Heun — συνδυάζονται αρμονικά για να βελτιώσουν τόσο την απόδοση όσο και την αποδοτικότητα. Ωστόσο, η απουσία της γλωσσικής πληροφορίας, που συνήθως μεταφέρει τη μεγαλύτερη σημασιολογική πυκνότητα (μέσω του BERT μοντέλου), οδηγεί σε αισθητή πτώση της επίδοσης, φαινόμενο γνωστό ως modality collapse.

### 0.6.5 Μελέτες αφαίρεσης

#### Σημασία των συνιστωσών

Για να επικυρώσουμε τη σημασία κάθε συνιστώσας στο προτεινόμενο πλαίσιό μας, διεξάγουμε συστηματικές μελέτες αφαίρεσης αφαιρώντας βασικές συνιστώσες και αξιολογώντας τον αντίκτυπό τους στη συνολική απόδοση. Αυτή η ανάλυση βοηθά στον προσδιορισμό της συμβολής κάθε ενότητας στην τελική ακρίβεια αναγνώρισης συναισθήματος.

**Πειραματική διάταξη**: Αξιολογούμε τη σημασία των συνιστωσών χρησιμοποιώντας τη βέλτιστη διάταξη μας (διατύπωση ΣΔΕ VP, ραχοκοκαλιά ScoreTransformer1D και δειγματολήπτη Heun με 60 NFEs) και στα δύο σύνολα δεδομένων CMU-MOSI και CMU-MOSEI υπό το πρωτόκολλο σταθερής έλλειψης (Fixed Missing Protocol).

**Table 8.** *Αποτελέσματα μελέτης αφαίρεσης συνιστωσών και για τα δύο σύνολα δεδομένων χρησιμοποιώντας τη διαμόρφωσή μας βελτιστοποιημένη ως προς την* ***ποιότητα****. Αναφέρουμε τις μέσες τιμές για το πρωτόκολλο σταθερής έλλειψης.*

| Configuration | CMU-MOSI | | | CMU-MOSEI | | |
|---|---|---|---|---|---|---|
| | **ACC2** | **ACC7** | **F1** | **ACC2** | **ACC7** | **F1** |
| **Full Framework (Ours)** | 75.2 | 75.1 | 35.1 | 76.1 | 75.6 | 48.4 |
| **w/o Diffusion Component** | 71.7 | 72.0 | 31.8 | 74.6 | 73.7 | 47.1 |
| **w/o Decoder Alignment** | 74.6 | 74.4 | 34.8 | 75.7 | 74.8 | 47.9 |
| **Performance Drop (w/o Diffusion)** | -3.5 | -3.1 | -3.3 | -1.5 | -1.9 | 1.3 |
| **Performance Drop (w/o Alignment)** | -0.6 | -0.7 | -0.3 | -0.4 | -0.8 | -0.5 |

**Ανάλυση των συνεισφορών των συνιστωσών**: Τα αποτελέσματα υπογραμμίζουν τον κρίσιμο ρόλο του μοντέλου διάχυσης (diffusion) στο πλαίσιό μας. Η αφαίρεση της διάχυσης οδηγεί στη μεγαλύτερη μείωση της απόδοσης και στα δύο σύνολα δεδομένων, ιδιαίτερα για τη δυαδική ακρίβεια και την ακρίβεια 7-κλάσεων, καθώς και για τη βαθμολογία F1, υποδεικνύοντας ότι αυτή η ενότητα είναι απαραίτητη για την αποτύπωση των λεπτών πολυτροπικών εξαρτήσεων.

Ο **αποκωδικοποιητής ευθυγράμμισης** (decoder alignment) συμβάλλει επίσης θετικά, αν και σε μικρότερο βαθμό. Η αφαίρεσή της οδηγεί σε μια μικρή αλλά σταθερή

πτώση της απόδοσης, υποδηλώνοντας ότι η ευθυγράμμιση των εξόδων του αποκωδικοποιητή μεταξύ των τροπικοτήτων βελτιώνει τη συνοχή των πολυτροπικών πληροφοριών και ενισχύει ελαφρώς την ευρωστία της πρόβλεψης. Το Σχήμα 9 είναι μια οπτικοποίηση των ανακατεσκευασμένων χαρακτηριστικών μέσω των μοντέλων διάχυσης με και χωρίς τον αποκωδικοποιήτη ευθυγράμμισης, παρατηρούμε πως χωρίς την ευθυγράμμιση αρκετά ανακατασκευασμένα δείγματα φαίνεται να εμφανίζονται μακριά άπο τα clusters των αρχικών χαρακτηριστικών.

## 0.7 Συμπέρασμα και μελλοντική δουλειά

Στην παρούσα εργασία μελετήθηκε ο σχεδιαστικός χώρος των πολυτροπικών μοντέλων διάχυσης για αναγνώριση συναισθήματος και διάθεσης υπό συνθήκες ελλιπών δεδομένων. Επεκτείνοντας το πλαίσιο του IMDER, δοκιμάστηκαν νέες αρχιτεκτονικές ύπο συνθήκη παραγωγής με Transformer ραχοκοκαλιές και συγκρίθηκαν με το αρχικό U-Net δίκτυο. Η αξιολόγηση σε γνωστά σύνολα δεδομένων (CMU-MOSI και CMU-MOSEI) έδειξε ότι τα προτεινόμενα μοντέλα είναι ανταγωνιστικά, ενώ ταυτόχρονα επιτυγχάνουν σημαντικές επιταχύνσεις στην εκπαίδευση και δειγματοληψία με μόνο οριακές απώλειες στην ακρίβεια. Έτσι, αναδείχθηκε η σημασία της βελτιστοποίησης για ταχύτητα και αποδοτικότητα.

Παρά τα θετικά αποτελέσματα, η μεθοδολογία παρουσιάζει ορισμένους περιορισμούς. Η μελέτη επικεντρώθηκε αποκλειστικά σε ανάλυση συναισθήματος, αφήνοντας ανοιχτό το ερώτημα της γενίκευσης σε πιο απαιτητικές πολυτροπικές εφαρμογές όπως η πολυτροπική μετάφραση ή η απάντηση σε οπτικο-γλωσσικές ερωτήσεις. Επιπλέον, τα μοτίβα απώλειας δεδομένων που εξετάστηκαν δεν αντικατοπτρίζουν πλήρως την πολυπλοκότητα πραγματικών σεναρίων, ενώ το υπολογιστικό κόστος παραμένει υψηλότερο από απλούστερες μεθόδους συγχώνευσης. Τέλος, η απόδοση είναι ευαίσθητη σε υπερπαραμέτρους όπως τα προγράμματα θορύβου και οι στρατηγικές δειγματοληψίας.

Ως μελλοντικές κατευθύνσεις, προτείνεται η εφαρμογή του πλαισίου σε πιο ποικίλες πολυτροπικές εργασίες και πραγματικά σενάρια χρήσης με διαφορετικούς περιορισμούς. Η ανάπτυξη πιο ευέλικτων μηχανισμών cross modal conditioning, η διερεύνηση εναλλακτικών γενετικών μοντέλων όπως τα Flow Matching, καθώς και η δημιουργία ενοποιημένων πλαισίων που θα χειρίζονται οποιοδήποτε μοτίβο απώλειας δεδομένων αποτελούν υποσχόμενες ερευνητικές προοπτικές.

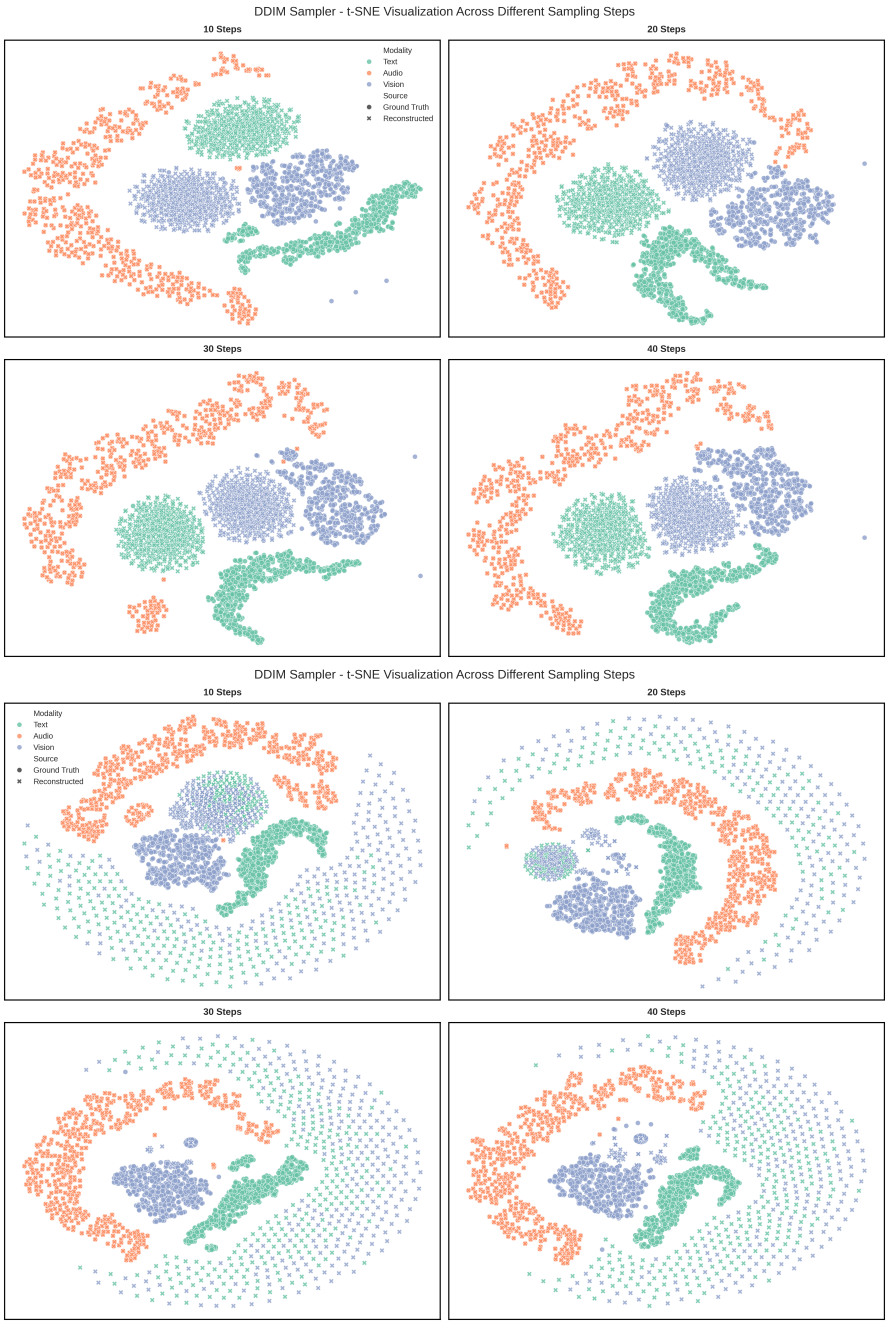


**Figure 9.** *t-SNE visualizations of reconstructed* ***audio features*** *under different sampling steps. Top: DDIM sampler with alignment decoder (10–40 steps). Bottom: DDIM sampler without alignment decoder (10–40 steps).*

Chapter 1

# Introduction

## 1.1 Background and Motivation

Human communication is inherently multimodal. People express their emotions not only through spoken words but also through tone of voice, facial expressions, and subtle behavioral cues. Capturing and understanding these complex affective signals is a cornerstone of affective computing and has broad applications across domains such as human-computer interaction, healthcare, education, entertainment, and social robotics. Multimodal Emotion Recognition (MER), illustrated in Figure 1.1, aims to address this challenge by combining information from different sensory channels, typically language, audio, and vision to achieve a more reliable and holistic estimation of human affective states including happiness, sadness, anger, fear, surprise, disgust, and neutral emotions.

The importance of MER lies in its ability to improve the robustness and naturalness of intelligent systems. Emotionally aware virtual assistants can provide more empathetic responses, online education platforms can better adapt to students' affective needs, and healthcare monitoring systems can detect early signs of psychological distress. Deep learning has accelerated progress in this direction by enabling powerful multimodal representation learning. Models like MISA [32] and MulT [26] explicitly capture cross-modal dependencies and achieve state-of-the-art results on benchmark datasets. However, these successes often rely on a critical assumption: that all modalities are fully observed and synchronized during both training and inference.

In real-world deployment scenarios, this assumption rarely holds. Sensor failures, occlusions, background noise, limited hardware resources, or inability to extract meaningful features due to the data nature can lead to missing or degraded modalities. For instance, a video call application may experience poor lighting conditions that impair facial expression recognition, or wearable sensors may fail to capture physiological signals consistently. In Figure 1.2 the individual covers their face with their hands and the

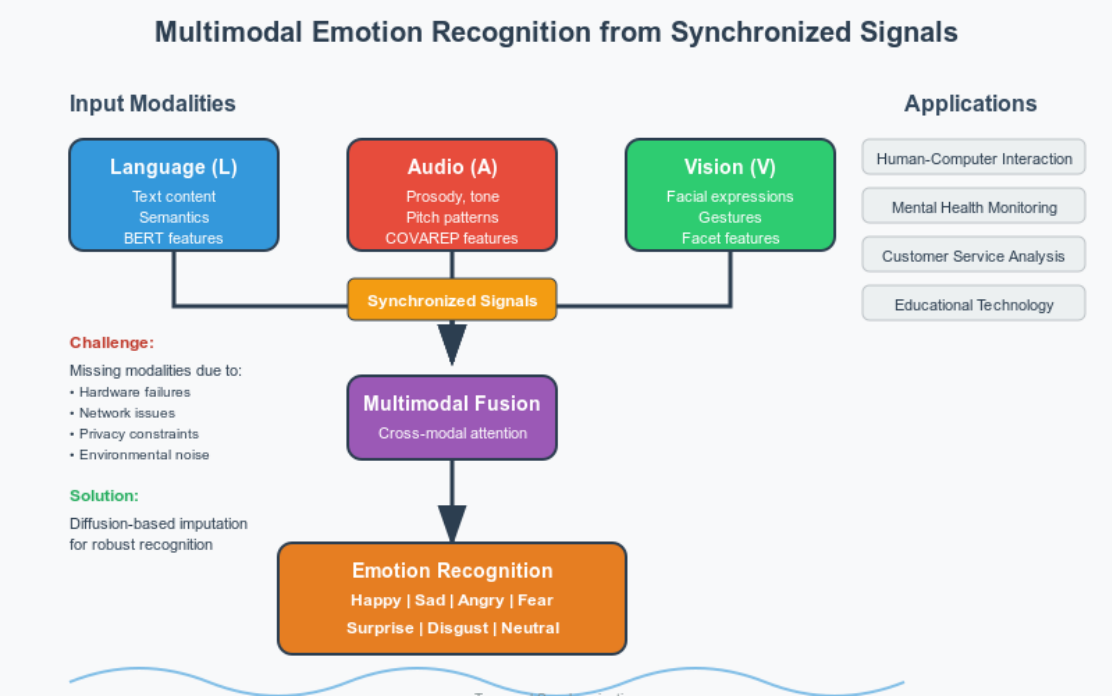


**Figure 1.1.** *Multimodal emotion recognition from synchronized signals.*

system cant extract visual information. Under such conditions, traditional multimodal learning approaches may fail catastrophically, as they are not designed to handle incomplete modality inputs. This fundamental limitation severely restricts the deployment of multimodal systems in unconstrained, real-world environments. To address missing modality conditions, two primary research directions have emerged:

**Imputation methods** attempt to estimate missing data from partially observed input. We review previous works and roughly divide them into three groups: zero/average imputation, low-rank imputation and DNN-based imputation.

**Zero/average Imputation:** Padding missing modalities with zero vectors or average values are widely utilized for data imputation [56, 57, 58]. For example, Parthasarathy et al. [57] filled missing frames of videos with zero vectors. Zhang et al. [56] padded missing modalities with average values based on the available samples within the same class. Zero/average imputation can achieve competitive performance in incomplete multimodal learning. However, since no supervision information is utilized, there is still a gap between filled values and original data, thus degrading the performance of downstream tasks.

**Low-rank Imputation:** Complete multimodal data exhibits correlations between

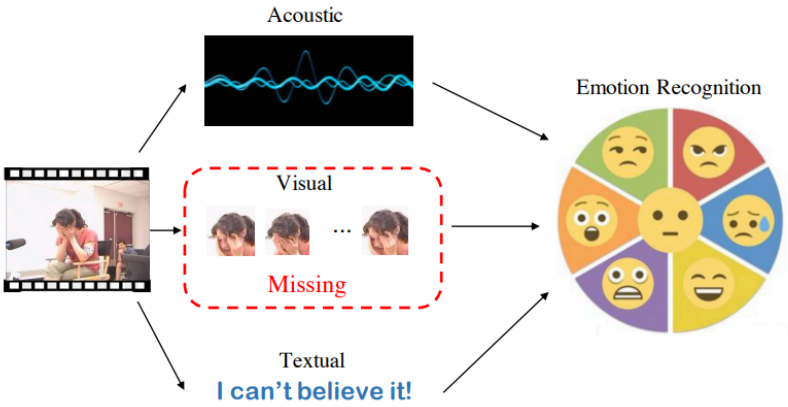


**Figure 1.2.** *Missing visual modality scenario where hand occlusion blocks facial feature extraction for emotion recognition. Figure from [6]*

different modalities and leads to the low-rank data matrix. However, incomplete data breaks these correlations and increases tensor rank [59, 60]. To capture multimodal correlations, previous works project data into a common space by using low-rankness. These approaches are usually based on nuclear norm minimization, such as singular value thresholding (SVT) [61] and Soft-Impture [62]. Besides nuclear norm, Fan et al. [63] also minimized tensor tubal rank to deal with various missing patterns. Furthermore, Liang et al. [59] combined the strength of non-linear functions to learn complex correlations in tensor rank minimization. However, these methods are usually computationally expensive for big data [64].

**DNN-based Imputation:** Due to the generative ability of DNNs, several DNN-based models have emerged to estimate missing data from partially observed input, e.g., autoencoder [65, 66], GAN [67, 68], VAE [69, 70] and Transformer [71, 72]. Among these approaches, autoencoder and its variants are widely utilized due to their promising results in incomplete multimodal learning [73]. For example, Duan et al. [74] leveraged autoencoders to impute missing data. To improve the modeling ability of autoencoders, Tran et al. [66] proposed the cascaded residual autoencoder (CRA). It combined a series of residual autoencoders [75] into a cascaded architecture for data imputation. Furthermore, Zhao et al. [73] incorporated CRA with cycle consistency loss for cross-modal imputation, which achieved superior performance over existing methods. More modern approaches utilize graph neural networks like GCNet by Lian et al.[30] and powerful generative models including, normalizing flows DiCMoR [35], or more recently, diffusion models such as in IMDER [1]. IMDER trains modality-specific score-based diffusion models

to learn the joint distribution of multimodal data, enabling conditional generation of missing signals from available modalities.

The second direction for dealing with incomplete modalities are **non-imputation** methods, which can be roughly divided into grouping strategies, correlation maximization and encoderless models.

**Grouping Strategy:** Complete data is easier to deal with than incomplete data. The grouping strategy directly partitions incomplete data into multiple complete subgroups, and then feature learning is carried out independently for each subgroup [76, 77, 78]. Despite its effectiveness, the number of subgroups grows exponentially with the number of modalities. Therefore, this strategy cannot work well for data with a large number of modalities or limited samples.

**Correlation Maximization:** To deal with the problem of incomplete data, an efficient approach is to maximize correlations between different modalities. In this way, we can constrain different modalities of the same sample to have related low-dimensional representations. Recently, several works based on correlation maximization have been proposed, including canonical correlation maximization [79, 80], HGR correlation maximization [81], mutual information maximization [82] and likelihood maximization [83]. Among these approaches, canonical correlation and its variants are widely utilized due to their promising results in incomplete multimodal learning. For example, Hotelling et al. [79] proposed CCA that learned relationships between multi-modalities by linearly mapping them into a low-dimensional common space with maximal canonical correlations. Different from CCA that focused on linear mappings, Andrew et al. [80] proposed DCCA that leveraged deep neural networks to learn more complex nonlinear combinations between multi-modalities. Wang et al. [84] further combined canonical correlations with reconstruction errors of autoencoders, trading off the structure information of each modality and the relationship between multi-modalities.

**Encoderless Model:** Unlike previous works that rely on encoders, encoderless models can learn latent representations without encoders. They directly optimize latent representations to reconstruct modality-incomplete data regardless of missing patterns [85, 86]. Typically, Zhang et al. [56] proposed CPM-Net, a robust encoderless model for incomplete multimodal learning. It combined the encoderless model with a clustering-like classification loss to learn well-structured features, which has validated its effectiveness on multimodal data with missing modalities.

The taxonomy and analysis of these approaches is modified from the work of Lian et al. on GCNet [30].

While computationally efficient, these methods may overlook useful cross-modal de-

pendencies and often exhibit performance degradation as the number of missing modalities increases.

## 1.2 Contributions

In this thesis we explored a **decoupled training framework** for efficient multimodal diffusion-based recovery MER and use optimized samplers, conditioning mechanisms and diffusion processes to outperform the baseline approach [1] in efficiency and match or outperform the performance. Our work builds upon IMDER but removes key bottlenecks by separating the training of the score networks from the final emotion prediction task. The initial approach in IMDER suggests an end-to-end system that uses untrained score networks to impute values to a trained fusion emotion prediction network resulting in gradients from generated samples that do not follow the proper distribution in the start of training. Furthermore there is no exploration of the diffusion process components. To address these limitations we present the following contributions:

1. A 2 stage training scheme is proposed firstly training the score networks and then deploying them in the MER task avoiding imputing untrained generated samples from the untrained diffusion models and degrading the starting training performance.

2. We consider and evaluate the most popular diffusion processes including **Variance Exploding (VE)** and **Variance Preserving (VP)** Stochastic Differential Equations (SDEs).

3. Furthermore, we swap the proposed Unet backbone with the more recent **Transformer backbone** [71] and evaluate different conditioning mechanisms like **AdaLN, FiLM and simple concatenation** [24, 18, 4] of the diffusion models literature.

4. Once trained, we evaluate four different diffusion samplers, including the default **Euler-Maruyama**, second order **Heun** and **Predictor Corrector** and the fast deterministic **DDIM** sampler to efficiently solve the reverse time SDEs and generate samples that will be used in downstream inference.

5. Finally we identify two optimized configurations one for quality based recovery and one optimized for inference speed based on all past experiments and compare them against all modern state of the art models, achieving accelerated efficiency without sacrificing performance compared to the original IMDER.

## 1.3 Thesis Outline

- In Chapter 2 we quickly review the landscape of generative modeling discussing some of the most impactful ones that later lead to the development of the most modern ones.
- In Chapter 3 we provide the background theory behind the breakthrough of diffusion models a deep generative network class.
- In Chapter 4 we explore the deep multimodal learning fundamentals, the missing modality problem found in multimodal datasets and we analyze some methods to alleviate it from the literature that we later use as benchmarks for our approach.
- In Chapter 5 all of the decisions regarding the decoupled architecture of the pipeline, proposed backbones, the hyperparameters of the model, the dataset and a dataset evaluation are provided.
- In Chapter 6 we experimentally explore the vast design space of the proposed method and infer the optimal configuration of components and compare it with the state of the art models proposed literature. Furthermore we conduct ablation studies to further confirm the robustness of our findings.
- In Chapter 7 we conclude the thesis and discuss some possible future work directions.

Chapter 2

# Classical Deep Generative Models

## 2.1 Vanilla Autoencoders

Autoencoders represent one of the earliest and most intuitive approaches in deep generative modeling. Their central purpose is to learn a compact representation (latent code) of input data by training a neural network to reproduce its input at the output. This is achieved by enforcing an information bottleneck: the encoder network $g_\phi(x)$ maps the input $x$ to a low-dimensional latent space $z$, and the decoder $f_\theta(z)$ reconstructs $x$ from $z$. Formally, the learning objective approximates the identity function $f_\theta(g_\phi(x)) \approx x$, while constraining $z$ to capture the most salient factors of variation [87].

Over the years, several variants of autoencoders have been introduced: **denoising autoencoders**, which learn robustness to noisy inputs by reconstructing clean signals; **sparse autoencoders**, which encourage sparsity in the latent code and promote disentanglement of features; and **contractive autoencoders**, which penalize sensitivity of latent representations to small input perturbations. Each of these extensions has played a role in shaping the research trajectory of generative modeling by emphasizing robustness, disentanglement, and stability.

Although simple autoencoders are not true generative models—as they lack a probabilistic interpretation and cannot easily generate new samples from the learned latent space—they form the basis for more advanced models such as VAEs and VQ-VAEs, where probabilistic structure and sampling are incorporated.

## 2.2 Variational AutoEncoders (VAE)

Variational AutoEncoders, [88] are also a very important idea of the generative AI toolkit, they aim to fit parameterized surrogate functions (Deep Networks) to the posteriors and the likelihoods of our latent codes and data. The encoder takes as input data

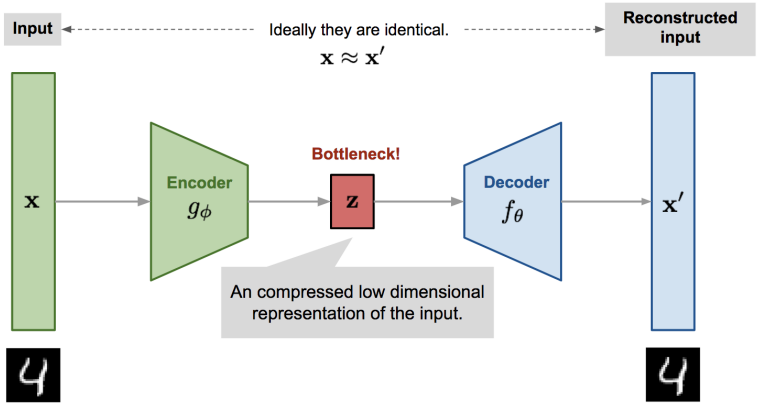


**Figure 2.1.** *Vanilla autoencoder architecture. Figure from [7].*

space samples and outputs the means and variances of the posterior density probability function of the latent code $z$, then we sample from it a code and pass it to the decoder that represents the likelihood of the data given a sample of $z$. A reparametrization trick is implemented to be able to backpropagate the loss to the encoder without any stochasticity in the actual outputs. The loss used in VAEs is called Evidence Lower Bound (ELBO):

$$\begin{aligned} \text{ELBO}(\theta, \phi; x) &= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}[q_\phi(z|x)||p(z)] \\ &= \int q_\phi(z|x) \log p_\theta(x|z)\, dz - \text{KL}[q_\phi(z|x)||p(z)], \end{aligned} \tag{2.1}$$

where:

- $p_\theta(x|z)$ is the generative model (decoder) parameterized by $\theta$.
- $q_\phi(z|x)$ is the variational distribution (encoder) parameterized by $\phi$.
- $\text{KL}[q_\phi(z|x)||p(z)]$ is the Kullback-Leibler (KL) divergence between the variational distribution and the prior distribution $p(z)$.
- The expectation $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ is taken with respect to the variational distribution.

The objective during training is to maximize the ELBO with respect to the model parameters $\theta$ and $\phi$:

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \frac{1}{N} \sum_{i=1}^{N} \text{ELBO}(\theta, \phi; x^{(i)}),$$

which states that we can solve an intractable problem, finding exact posteriors and maximizing the marginal, by optimizing another feasible objective. VAEs are a core concept in generative DL since they incorporate techniques used in more state of the art models and the underlying approximate variational optimization task they are trying to solve can be found in a lot of places in the field. More details can be found in this insightful discussion available [7].

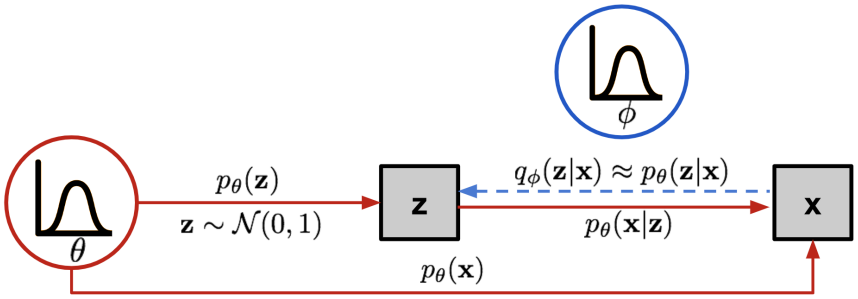


**Figure 2.2.** *The Graphical model that describes Variational AutoEncoders. Figure from [7].*

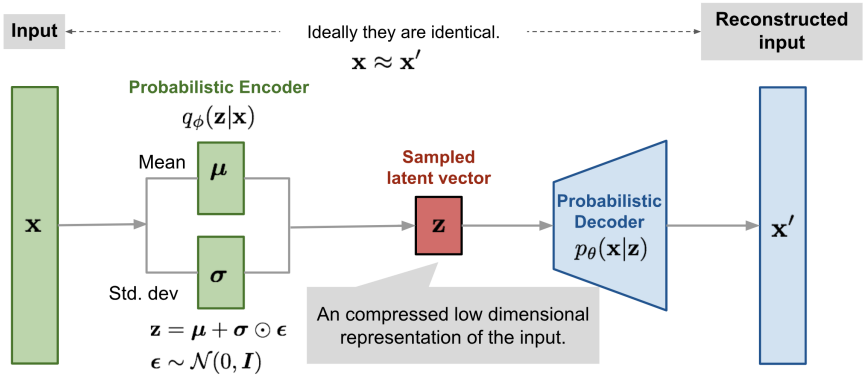


**Figure 2.3.** *VAE architecture using Gaussian parameters latent codes and a reparametrization trick to propagate the gradients. Figure from [7].*

## 2.3 Normalizing Flows

Normalizing Flows (NFs) [89, 90] represent a powerful family of generative models that build upon the principle of invertible transformations. Unlike VAEs, which approx-

imate posteriors through variational bounds, flows explicitly construct exact likelihoods by learning a series of bijective mappings $f_i$ that progressively transform a simple distribution (e.g., Gaussian) into a complex data distribution. Through the change-of-variables formula:

$$p_X(x) = p_Z(f^{-1}(x)) \left| \det \frac{\partial f^{-1}(x)}{\partial x} \right|,$$

flows provide tractable density evaluation and exact posterior inference.

Architectural innovations such as **NICE** [91] and **RealNVP** [92] introduced coupling layers that make the Jacobian determinant efficient to compute. Further extensions such as **Masked Autoregressive Flows (MAF)** [93] and **Inverse Autoregressive Flows (IAF)** [94] leveraged autoregressive networks for more expressive mappings. Glow [95] popularized flows in image generation by introducing invertible $1 \times 1$ convolutions, enabling large-scale applications. More recent innovations include continuous-time flows such as FFJORD [96], which model invertible transformations as ordinary differential equations, and invertible residual networks (i-ResNets) [97], which relax constraints on invertibility while preserving tractable log-determinants.

While normalizing flows provide exact likelihoods and interpretable latent spaces, they face challenges in terms of **expressivity versus efficiency**. Ensuring tractable Jacobian determinants requires restrictive architectural choices, which may limit flexibility compared to GANs or diffusion models. Despite this, flows remain an important foundation for modern likelihood-based approaches, and their invertible structure has influenced recent advances in score-based generative modeling and diffusion probabilistic models [**?**].

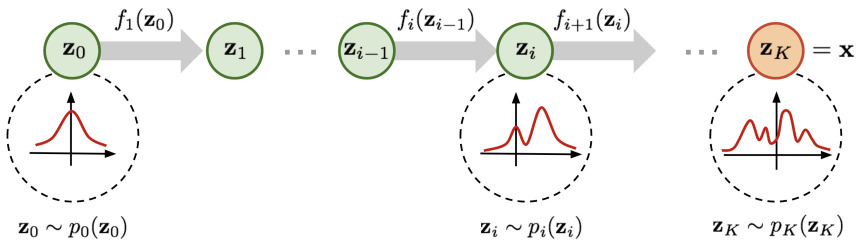


**Figure 2.4.** *Normalizing flows gradually transforming a prior distribution to a complex one. Figure from [8].*

## 2.4 Vector Quantised-Variational AutoEncoders (VQ-VAE)

The Vector Quantised-Variational Autoencoder (VQ-VAE) [9, 98] introduces discrete latent representations into the generative modeling landscape. In contrast to continuous latent codes used by standard VAEs, VQ-VAEs map encoder outputs to a discrete codebook of learned embeddings. The nearest codebook entry replaces the encoder's continuous output, producing quantized latent vectors that the decoder reconstructs into the original data.

This discrete latent structure confers several advantages: it prevents posterior collapse (a common issue in VAEs), provides a richer and more interpretable latent space, and facilitates the use of powerful autoregressive priors such as PixelCNN or Transformers to model sequences of discrete codes. As a result, VQ-VAEs have been widely adopted in applications such as speech synthesis (e.g., WaveNet [99], VQ-VAE-2) and image generation, often serving as the backbone for large-scale models like DALL-E [100].

However, the quantization step introduces non-differentiability. To address this, VQ-VAEs rely on a combination of straight-through gradient estimators and codebook updates, which while effective, can complicate optimization.

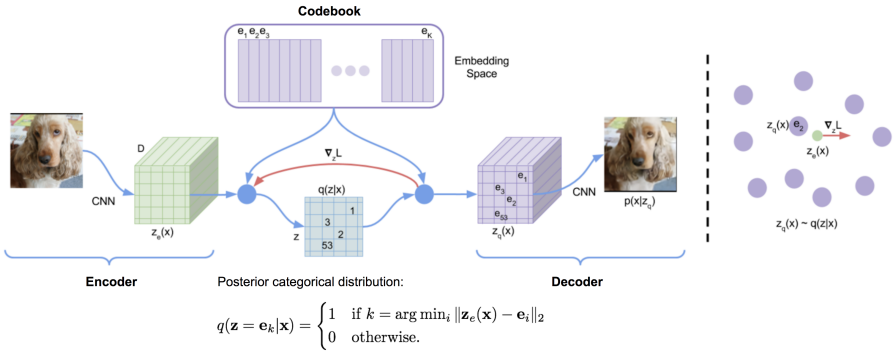


**Figure 2.5.** *Vector Quantised Variational Autoencoder. Figure from [9].*

## 2.5 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) [101] represent a major paradigm shift in generative modeling by framing sample generation as a two-player game between a generator $G$ and a discriminator $D$. The generator seeks to produce samples $G(z)$ from latent noise $z$ that resemble real data, while the discriminator aims to distinguish between real

samples $x \sim p_{\text{data}}$ and generated ones. The training objective is defined as a minimax optimization:

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

This adversarial formulation enables GANs to implicitly learn data distributions without explicit likelihood estimation, often producing remarkably sharp and realistic samples compared to VAEs.

Despite their success, GANs are notoriously difficult to train. Issues such as **mode collapse** (where the generator produces limited diversity), unstable convergence, and sensitivity to hyperparameters have driven research into numerous variants. Improvements such as Wasserstein GANs [102], Least-Squares GANs [103], and StyleGAN [104] have enhanced stability, interpretability, and controllability of generated outputs.

GANs have had profound impact on fields such as computer vision (image synthesis, super-resolution, style transfer) and are often benchmarked as the state-of-the-art in sample fidelity. Nonetheless, their lack of explicit likelihoods and fragile optimization dynamics distinguish them from likelihood-based approaches such as VAEs and diffusion models.

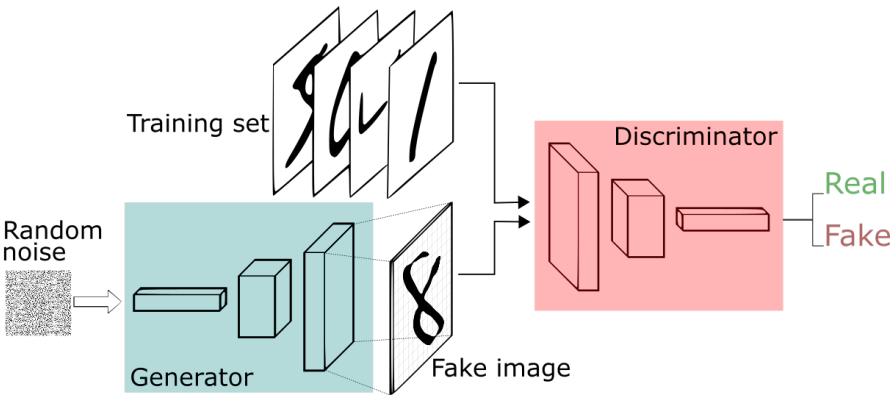


**Figure 2.6.** *Generative Adversarial Networks. Figure from [10].*

Chapter 3

# Diffusion Models

## 3.1 Variational Diffusion Models

### 3.1.1 Deep Unsupervised Learning using Nonequilibrium Thermodynamics

This paper, authored by [38], introduces the foundational concept of diffusion models, designed to systematically restore order from pure prior noise. Rooted in non-equilibrium statistical physics and sequential Monte Carlo methods, the approach leverages a Markov Chain as a graphical model. Unlike previous generative methods that aimed to generate samples in a single step, the proposed model proposes a forward trajectory through the Markov chain, gradually transforming data samples into a prior noise sample and then a learnt model tries to predict the reverse process.

This distinctive approach employs a forward transition kernel with small Gaussian perturbations at each step. The small step sizes introduced during the forward process yield a reverse process with an identical functional form. Exploiting this symmetry, the authors suggest training a model to predict the mean and variance of the reverse posterior conditioned on the previous sample. Consequently, maximizing the marginal probability of the generated data involves optimizing the conventional variational lower bound of this Markov Chain graphical model.

The paper underscores the critical role of the diffusion rate schedule in constructing these models, emphasizing its significant impact on performance. Notably, the diffusion schedule is dynamically learned through gradient ascent of the lower bound objective, for the Gaussian Transition Kernel, and is held constant during the parameter learning phase for Gaussian diffusion.

Experimental results demonstrate the model's capabilities across a spectrum of generative modeling tasks, including manifold learning, image inpainting, and image gener-

ation.

### 3.1.2 Denoising Diffusion Probabilistic Models

This seminal paper of [11] on diffusion models is a natural progression from the previous one stated above. Based on that the authors expanded and improved the ideas introduced. We will cover the mathematical backround here since its more consistent with current notation.

Diffusion models represent latent variable models defined by the expression $p_\theta(x_0) := \int p_\theta(x_{0:T})\, dx_{1:T}$, where $x_1, \ldots, x_T$ represent latent variables of the same dimensionality as the data $x_0$, sampled from the distribution $q(x_0)$.

The joint distribution $p_\theta(x_{0:T})$, referred to as the reverse process, is modeled as a Markov chain. Gaussian transitions originating from $p(x_T) = \mathcal{N}(x_T; 0, I)$, practically prior noise, guide the sample to the data distribution:

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \tag{3.1}$$

where

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{3.2}$$

The forward diffusion process is a Markov Chain where noise is gradually added according to a schedule $\beta_i$, for $i \in 1, 2, \ldots, T$. The noising one step distribution is called perturbation kernel:

$$q(x_t|x_{t-1}) := \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right). \tag{3.3}$$

One notable advancement lies in the discovery that, conditioned on the initial sample $x_0$, any desired noised sample $x_t$ can be efficiently obtained through a closed form, $\alpha_t := 1 - \beta_t \quad \text{and} \quad \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$,

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I\right), \tag{3.4}$$

thereby accelerating the training process.

Additionally, the paper demonstrates that during the reverse diffusion process, it is possible to track the reverse posterior $q$ conditioned on the sample $x_0$:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \widetilde{\mu}_t(x_t, x_0), \widetilde{\beta}_t I\right), \tag{3.5}$$

where

$$\widetilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha}_t(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t, \tag{3.6}$$

and

$$\widetilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{3.7}$$

This insight forms the basis for a novel training objective for the model at each time step.

The authors formulated the loss function as a variational objective over the steps of the Markov chain:

$$\mathbb{E}_q \left[ \begin{array}{l} D_{KL}\left(q(x_T|x_0) \,||\, p(x_T)\right) \\ + \sum_{t>1} D_{KL}\left(q(x_{t-1}|x_t, x_0) \,||\, p_\theta(x_{t-1}|x_t)\right) \\ - \log p_\theta(x_0|x_1) \end{array} \right], \tag{3.8}$$

and by further formula derivations it can be shown that predicting the mean of the noise added to the sample is equivalent. With a reparametrization trick we can even show that predicting the noise at the next state is also equivalent, simplifing the objective further. Lastly the researchers concluded that dropping the constants in front of the loss gave better results, so the simplified objective used to train their models was:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,x_0,\epsilon}\left[\frac{1}{2}\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right], \tag{3.9}$$

Algorithm 1: *Training*

1: **repeat**
2: $x_0 \sim q(x_0)$
3: $t \sim \text{Uniform}(\{1, \ldots, T\})$
4: $\epsilon \sim \mathcal{N}(0, I)$
5: Take gradient descent step on $\nabla_\theta \left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2$
6: **until** converged

After training our noise predictor we can start from a prior distributed noise sample and iteratively denoise it to sample from the data distribution:

The paper adopts a fixed variance schedule, increasing linearly, and employs a U-Net architecture as the backbone of the reverse process model. Experimental results demonstrate performance remarkably close to state-of-the-art GANs that at the time

Algorithm 2: *Sampling*

$x_T \sim \mathcal{N}(0, I)$
**for** $t = T, \ldots, 1$ **do**
  **if** $t > 1$ **then**
    $z \sim \mathcal{N}(0, I)$
  **else**
    $z \leftarrow 0$
  **end if**
  $x_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$
**end for**
**return** $x_0$

had evolved over nearly six years although, the training and sampling where much more expensive.

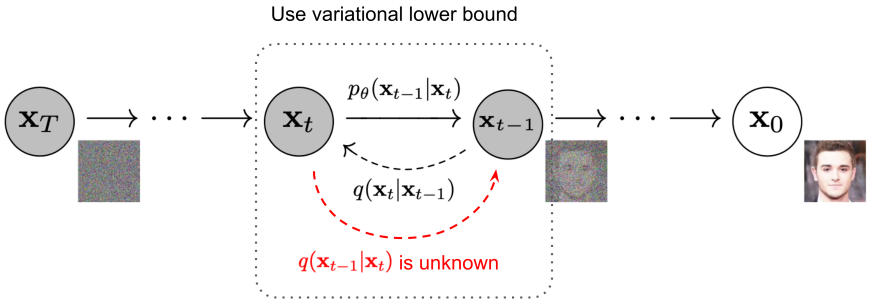


**Figure 3.1.** *Markov chain Graphical Model of the forward and reverse diffusion process. Figure from [11].*

## 3.2 Score Based Generative Modeling

### 3.2.1 Generative Modeling by Estimating Gradients of the Data Distribution

In the pursuit of directly modeling complex data distributions, where the model's output represents the actual distribution function, the computation of normalizing constants poses a computationally challenging problem. Score-based modeling offers a solution by estimating the gradient with respect to the data variable $x$ of the corresponding distri-

bution. Lets say that the output of a model gives us the PDF:

$$p_\theta(\mathbf{x}) = \frac{e^{f_\theta(\mathbf{x})}}{Z_\theta} \tag{3.10}$$

- $p_\theta(\mathbf{x})$: Probability density function (PDF) of the distribution parameterized by $\theta$, evaluated at the value $\mathbf{x}$.
- $f_\theta(\mathbf{x})$ is often the logit or energy function.
- $Z_\theta$: Normalization constant, also known as the partition function. Ensures that the PDF integrates (or sums, in discrete cases) to 1 over the entire range of possible values of $x$.

The Score function is:

$$\nabla_\mathbf{x} \log p_\theta(\mathbf{x}) = \nabla_\mathbf{x} f_\theta(\mathbf{x}) + \nabla_\mathbf{x} \log Z_\theta = \nabla_\mathbf{x} f_\theta(\mathbf{x}) = \mathbf{s}_\theta(\mathbf{x})$$

This approach proves advantageous as the normalization constant is not needed when trying to model the score function resulting in its elimination after computing the gradient, and it is easily demonstrated that having either the constant or its gradient allows for a seamless transition between the two through differentiation or integration. Score based Generative modeling aims to train deep networks that output the score function of the underlying data distribution. The objective is stated as follows:

$$\frac{1}{2}\mathbb{E}_{p_{data}(\mathbf{x})}\left[\|\mathbf{s}_\theta(\mathbf{x}) - \nabla_\mathbf{x} \log p_{\text{data}}(\mathbf{x})\|_2^2\right]$$

Gauss's theorem shows that we can approximate the actual distribution score as follows:

$$\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\text{tr}(\nabla_\mathbf{x} s_\theta(\mathbf{x})) + \frac{1}{2}\|\mathbf{s}_\theta(\mathbf{x})\|_2^2\right] \tag{3.11}$$

Where $\nabla_\mathbf{x}\mathbf{s}_\theta(\mathbf{x})$ is the Jacobian of $\mathbf{s}_\theta(\mathbf{x})$. The term $\text{tr}(\nabla_\mathbf{x}\mathbf{s}_\theta(\mathbf{x}))$ needs linear number of backpropagations growing with the data dimensions and is not scallable.

Sliced score matching aims to solve this by random projections in order to approximate the term. Another way is with denoising score matching where we fully bypass the gradient of the score. Once we obtain a good approximation of the score function, $\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_\mathbf{x} \log p_{data}(\mathbf{x})$, we can use Lavengin dynamics to produce samples starting from a prior distributed initial value.

The main problem that arises in high dimensional data is data sparsity. The manifold assumption states that data reside in a low dimensional manifold inside the data space,

as a result, most of the space is empty or sparse and we can't obtain an accurate approximation of the score function at those areas. Sparsity also affects the sampling process even if we have a ground truth scores disregarding the actual weights of the distribution modes. The result is slow mixing when using Lavengin dynamics in order to avoid sample density errors, meaning that sampling doesn't obey the actual distribution.

In order to deal with these problems the Song et al. [105] introduced Noise Conditional Score Networks (NCSN), in short they aim to perturb the data in different noise levels and train a Score network conditioned on the noise level. This process effectively "spreads" the samples decreasing sparsity. The objective is to enable the network to approximate the score function in previously low data density regions. We define the noise distribution as $q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}|\mathbf{x}, \sigma^2 I)$; therefore the score is,

$$\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = -\frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2}.$$

The updated objective becomes:

$$l(\theta; \sigma) \triangleq \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\tilde{\mathbf{x}}\sim\mathcal{N}(\mathbf{x},\sigma^2 I)}\left[\left\| s_\theta(\tilde{\mathbf{x}}, \sigma) + \frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2}\right\|_2^2\right].$$

Subsequently, the network undergoes training via score matching, refining the accuracy of score predictions.

For sampling the authors proposed an sampling approach called annealed Langevin dynamics using the learned network to guide a prior noise sample towards the data manifold. Basically its an updated version of Lavegin Dynamics where we sample in every noise level using the corresponding NC score function, the step sizes are reduced with every noise level update in order to eventually converge to the actual distribution.

The experiments presented were very competitive with other state of the art models at the time.

### 3.2.2 Score-Based Generative Modeling through Stochastic Differential Equations

This is yet another seminal work by [106] on the matter where they developed a unified view of the score based and denoising diffusion probabilistic generative models. Basically an Ito SDE can model the diffusion process:

$$d\mathbf{x} = f(\mathbf{x}, t)dt + g(t)d\mathbf{w} \tag{3.12}$$

Algorithm 3: *Annealed Langevin Dynamics*

**Require:** $\{\sigma_i\}_{i=1}^{L}$, $\epsilon$, $T$
Initialize $\tilde{\mathbf{x}}_0$
**for** $i \leftarrow 1$ to $L$ **do**
  $\alpha_i \leftarrow \epsilon \cdot \frac{\sigma_i^2}{\sigma_L^2}$ ▷ Step size
  **for** $t \leftarrow 1$ to $T$ **do**
    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2} s_\theta(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\mathbf{z}_t$
  **end for**
  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
**end for**
**return** $\tilde{\mathbf{x}}_T$

Where $w$ is the standard Wiener process or known as Brownian Motion and $f(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ is called the drift coefficient and $g(\cdot) : \mathbb{R} \to \mathbb{R}$ diffusion coefficient of $\mathbf{x}(t)$. In this framework the transition kernel receives samples from the continuous $t$ rather than the discrete levels of NCSM or DDPMs. What this SDE gives us is a general framework for diffusing samples in a continuum of intermediate noisy distributions, in Figure 10 you can see how a prior toy distribution of a Gaussian mixture of two modes can be diffused into a single Gaussian prior by running it through the diffusion SDE for some time $T$. Consciously reversing the above SDE can give us samples from the data distribution starting from a known prior, Anderson (1982) found that the reverse time SDE is also diffusion process running backward in time and is given by:

$$d\mathbf{x} = [f(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})]dt + g(t)d\bar{\mathbf{w}} \tag{3.13}$$

where the wiener process $\bar{\mathbf{w}}$ now flows backward, and $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is the score function of the perturbed distribution at time step $t$. Once we define $f$, $g$ and learn this parameterized score function we are able to reverse the process and sample it by simulating it using some type of numerical solver.

The authors also note that all the previous attempts to tackle the generative approach of diffusion models (SMLD and DDPM) can be regarded as discretizations of these SDEs for the forward case, specifically for the case of SMLD we can derive a Markov chain for the perturbation kernel of the form:

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \sqrt{\sigma_i{}^2 - \sigma_{i-1}{}^2}\mathbf{z}_{i-1}\ ,\ i = 1\ldots, N$$

where $\mathbf{z}_{i-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and when we take its limit with regard to the noise levels we get a continuous stochastic process given by the following SDE:

$$d\mathbf{x} = \sqrt{\frac{d[\sigma^2(t)]}{dt}}d\mathbf{w} \tag{3.14}$$

For the case of DDPM perturbation kernels we have the Markov Chain:

$$\mathbf{x}_i = \sqrt{1-\beta_i}\mathbf{x}_{i-1} + \sqrt{\beta_i}\mathbf{z}_{i-1}\ ,\ i = 1\ldots, N$$

which also converges to the SDE:

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}. \tag{3.15}$$

The authors state and prove that the former SDE describing the SMLD approach has exploding variances when $t \to \infty$ and named it Variance Exploding SDE (VE) and the latter DDPM approach yields a process with fixed varieance hence the name Variance Preserving SDE (VP). Also they proposed another type of SDEs that perform well on likelihoods and called it sub-VP SDE that is bounded by the VP SDE:

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)(1 - \mathrm{e}^{\left(-2\int_0^t b(s)\,ds\right)})}d\mathbf{w}. \tag{3.16}$$

Using numerical methods we can produce approximate trajectories from SDEs. Euler-Maruyama and Runge-Kutta are existing methods that can solve these systems and be used for sample generation. Lastly the authors compared ancestral sampling used in DDPMs with reverse diffusion samplers and concluded a slight advantage with the latter.

Another significant contribution of this paper is the introduction of a deterministic ODE that exists for all SDEs and simulating trajectories gives us the same marginal probabilities as the SDEs. This ODE was named by the authors the probability flow ODE:

$$d\mathbf{x} = \left[f(\mathbf{x}, t) - \frac{1}{2}g(t)^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\right]dt \tag{3.17}$$

the process above allows a plethora of capabilities, firstly we can use it to get exact likelihoods, latent encoding of data samples that can be used for interpolation and since

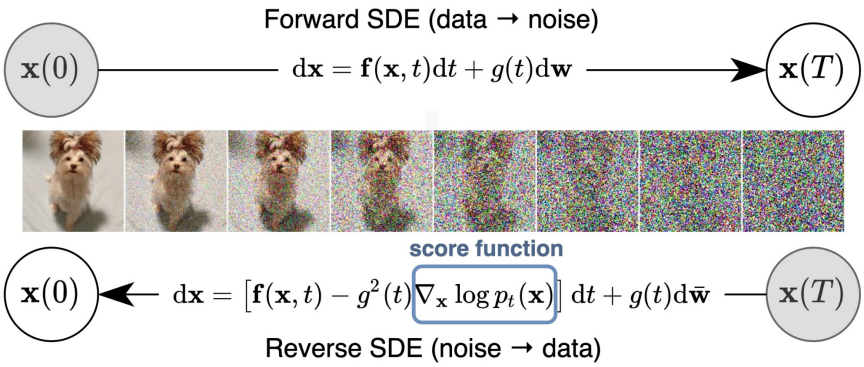


**Figure 3.2.** *Transforming data to a simple noise distribution using a continuous-time Ito SDE. This process can be reversed once we learn the score of the distribution at each intermediate time step. Figure from [12].*

the reverse process now is deterministic the latent encoding and its corresponding data space sample are uniquely identifiable since the forward SDE has no trainable parameters, and the correspoinding probability flow grants deterministic trajectories.
Lastly its noteworthy to mention the ease of controllable generation meaning that we can condition our reverse process to sample from a specific mode of the data distribution (class, text embedding, etc). So the conditional reverse-time SDE:

$$d\mathbf{x} = \{f(\mathbf{x}, t) - g(t)^2[\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(\mathbf{y}|\mathbf{x})]\}dt + g(t)d\bar{\mathbf{w}}. \tag{3.18}$$

From the above expression we observe that we don't have to train conditional models for each class but use the pre-trained unconditional model and a forward model $p_t(\mathbf{y}|\mathbf{x}(\mathbf{t}))$ that "guides" the process to the correct mode of our data distribution. The authors used controllable generation to tackle applications of class-conditional image generation, image imputation and colorization.
In the context of our study, the Frechet Inception Distance (FID) scores mentioned were considered state-of-the-art at the time. The authors conducted extensive experiments, exploring various architectures and Stochastic Differential Equation (SDE) types.

The findings indicated that employing the sub-VP SDE and VE SDE, coupled with deep architectures, yielded state-of-the-art results for both FID and negative log likelihoods (NLL). This outcome underscores the effectiveness of these specific SDE types in capturing and generating high-quality samples, contributing to the advancement of generative models.

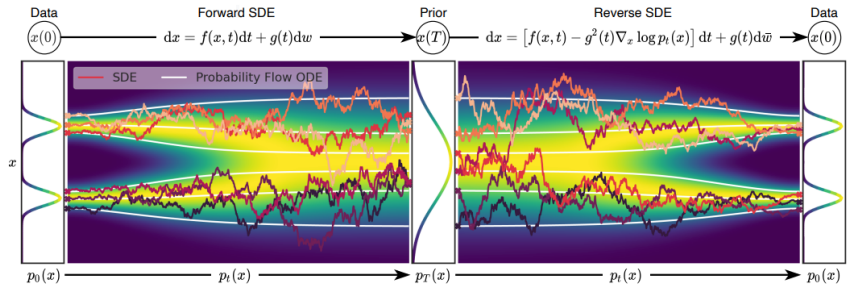


**Figure 3.3.** *The Forward path uses SDE to smoothly transform a complex data distribution to a known prior distribution and the backward path uses reverse-time SDE to transform the prior distribution into the data distribution. Figure from [12].*

## 3.3 Optimizing and Accelerating Diffusion Models

Even though diffusion models showed promising performance they were held back by their slow inference spanning multiple model evaluations thus making them computationally expensive and practically unusable. Most recent research is focused on accelerating diffusion models by optimizing noise schedules, higher order samplers, progressive distillation techniques, optimizing training schemes and more. We will briefly analyse some core literature in this chapter.

### 3.3.1 Diffusion Process Optimization

#### Improved Denoising Diffusion Probabilistic Models

In this work by Nichol et al. [41] significant enhancements are introduced, extending the horizon to 4000 steps and incorporating a hybrid loss to effectively learn the reverse posterior's variance. Recognizing the crucial impact of these variances on performance, learning these variances is an unstable process and direct inference of these parameters is hard for a neural network since they are really small, even for the log domain. As a result the authors considered an learned interpolation of the forward and reverse variances.

Additionally, the noise schedule undergoes refinement, transitioning to a cosine schedule. This adjustment reduces the rapid corruption of samples into noise, as observed in the previous linear schedule, resulting in a broader range of better training samples. The presented state-of-the-art results further underscore the efficacy of these model enhancements.

Lastly, the authors achieve a notable improvement in sample speed for diffusion models by reducing the sample steps by an order of magnitude in the reverse process. This

optimization proves sufficient to achieve near-optimal FID in fully trained models on the image generation task, showcasing the efficiency gains achieved through these modifications.

#### Improved Techniques for Training Score-Based Generative Models

Song et al. [107] address the challenges of unstable training and slow sampling in Noise Conditional Score Networks (NCSNs). Through a theoretical analysis focused on high-dimensional spaces, they introduce a stability-enhancing technique involving an exponential moving average of the model weights and an updated noise schedule. The outcomes showcase remarkable success, yielding high-fidelity samples that rival the quality achieved by leading GANs.

### 3.3.2 Fast Sampling Based Approaches

#### Denoising Diffusion Implicit Models

DDIMs where proposed by [108] as a way to tackle the long sampling process that diffusion models suffer from. They extend the diffusion process into non-Markovian chains, aiming to speed things up. The core idea involves making an educated guess about the initial sample $x_0$ and then, through a reverse process, obtaining the next latent variable based on that guessed sample. Below we define the non-Markovian case of the reverse diffusion conditioned on $x_0$ which is tractable:

$$q_\sigma(x_{1:T}|x_0) := q_\sigma(x_T|x_0) \times \prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0), \tag{3.19}$$

$$\text{where } q_\sigma(x_T|x_0) = \mathcal{N}\left(\sqrt{\alpha_T}x_0, (1-\alpha_T)I\right), \text{ for } t > 1,$$

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-{\sigma_t}^2} \cdot \frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 I\right). \tag{3.20}$$

Using the above we can "guess" the sample $x_0$ and find the next component of the chain. For some $x_0 \sim q(x_0)$ and $\epsilon_t \sim \mathcal{N}(0, I)$, $x_t$ by rewriting:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon.$$

one can then predict the denoised observation, which is a prediction of $x_0$ given $x_t$:

$$f_\theta^{(t)}(x_t) := \frac{(x_t - \sqrt{1-\alpha_t} \cdot \varepsilon_\theta^{(t)}(x_t))}{\sqrt{\alpha_t}}. \tag{3.21}$$

We can then define the generative process with a fixed prior $p_\theta^{(t)}(x_T) = \mathcal{N}(0, I)$ and

$$p_\theta(x_{t-1}|x_t) = \begin{cases} \mathcal{N}\left(f_\theta^{(1)}(x_1), \sigma_1^2 I\right) & \text{if } t = 1, \\ q_\sigma(x_{t-1}|x_t, f_\theta^{(t)}(x_t)) & \text{otherwise,} \end{cases}$$

Hence, the actual model becomes implicit since we don't have an explicit parametric specification for the reverse distribution (except the last step), but we parameterize the condition of the reverse non-Markovian process.

Now by taking a subset of steps $\{x_{\tau_1}, \ldots, x_{\tau_S}\}$, where $\tau$ is an increasing sub-sequence of $[1, \ldots, T]$ of length $S$ and running the above procedure we can generate samples of high quality with a lot fewer steps:

$$\begin{aligned} x_{t-1} =& \sqrt{\alpha_{t-1}} \frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \\ &+ \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t) + \sigma_t \epsilon_t \end{aligned} \tag{3.22}$$

When we set $\sigma_t = 0$ the generative process becomes deterministic and the resulting model becomes implicit.

A major part of the success of this approach is that you don't have to retrain DDPMs to implement this accelerated sampling, you just need to tweak the sampling strategy. The proposed methods for sampling turned out to be much more effective than regular DDPMs, especially for a small number of reverse steps. Even for about 100 reverse steps, the results were surprisingly close to doing the full 1000 steps in reverse after training on 1000 forward steps which is a critically important result.

### Elucidating the Design Space of Diffusion-Based Generative Models

In their seminal work, [42] made significant contributions to the theoretical underpinnings of Diffusion-based Generative Modeling. Drawing inspiration from the foundational work of [106] and the Markovian approach proposed by [11], Karras et al. elegantly connected these perspectives within a unified framework.

Essentially, they achieved the formalization of both approaches as discretizations of a more generalized Stochastic Differential Equation (SDE). This formalization allowed for the coherent tuning and selection of parameters within a unified structure. Armed with this versatile framework, Karras et al. combined the different approaches, exploring and finding the best possible parts for this unified equation, as a result achieving state-of-the-art accuracy. Another contribution of their lies in the training scheme proposed in order

to mitigate noise amplification from the raw denoising process. Lastly they proposed sampling techniques for deterministic and stochastic sampling based on second order methods and a discretization scheme for choosing noise levels, parameterized so that its more flexible. In summary, due to their work diffusion based models were made more flexible uniting them within a cohesive theoretical framework and SOTA performance in terms of sampling steps and FID benchmark scores.

### 3.3.3 Progressive Distilation for Fast Sampling of Diffusion Models

In progressive distillation proposed by Salimans et al. [13] the authors manage to distil knowledge from a teacher model trained on a normal diffusion setting to a student model that tries to predict multiple DDIM steps of the teacher. Basically they initialize an identical student-teacher pair and iteratevly try to optimize the student to match the teachers 2-step DDIM prediction until convergence. They then repeat the same process doubling the DDIM sampling steps the student tries to predict until they reduce them by orders of magnitude without too much loss in sampling quality. This work showed great progress in accelerating diffusion models and is commonly implemented in commercial diffusion models.

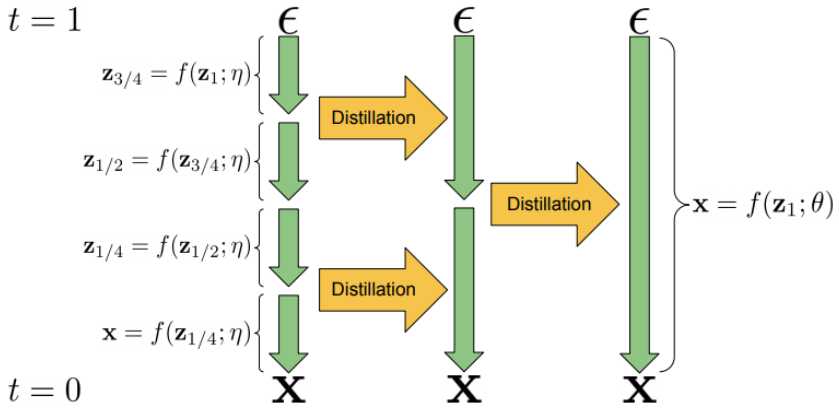


**Figure 3.4.** *Progressive distillation technique scheme. Figure from [13].*

## 3.4 Guiding Diffusion Models

Another important aspect of diffusion models is conditioning them in order to generate specific classes or modes of a distribution. This subject is at the core of most commercial uses of diffusion models found online and is what allowed text guided models

**Algorithm 1** Standard diffusion training

**Require:** Model $\hat{\mathbf{x}}_\theta(\mathbf{z}_t)$ to be trained
**Require:** Data set $\mathcal{D}$
**Require:** Loss weight function $w()$

**while** not converged **do**
  $\mathbf{x} \sim \mathcal{D}$ ▷ Sample data
  $t \sim U[0, 1]$ ▷ Sample time
  $\epsilon \sim N(0, I)$ ▷ Sample noise
  $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ ▷ Add noise to data
  $\tilde{\mathbf{x}} = \mathbf{x}$ ▷ Clean data is target for $\hat{\mathbf{x}}$
  $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ ▷ log-SNR
  $L_\theta = w(\lambda_t)\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$ ▷ Loss
  $\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$ ▷ Optimization
**end while**

**Algorithm 2** Progressive distillation

**Require:** Trained teacher model $\hat{\mathbf{x}}_\eta(\mathbf{z}_t)$
**Require:** Data set $\mathcal{D}$
**Require:** Loss weight function $w()$
**Require:** Student sampling steps $N$
**for** $K$ iterations **do**
  $\theta \leftarrow \eta$ ▷ Init student from teacher
  **while** not converged **do**
    $\mathbf{x} \sim \mathcal{D}$
    $t = i/N, \;\; i \sim Cat[1, 2, \ldots, N]$
    $\epsilon \sim N(0, I)$
    $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
    `# 2 steps of DDIM with teacher`
    $t' = t - 0.5/N, \;\; t'' = t - 1/N$
    $\mathbf{z}_{t'} = \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_t) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\eta(\mathbf{z}_t))$
    $\mathbf{z}_{t''} = \alpha_{t''} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}_{t'} - \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}))$
    $\tilde{\mathbf{x}} = \frac{\mathbf{z}_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
    $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$
    $L_\theta = w(\lambda_t)\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$
    $\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$
  **end while**
  $\eta \leftarrow \theta$ ▷ Student becomes next teacher
  $N \leftarrow N/2$ ▷ Halve number of sampling steps
**end for**

**Figure 3.5.** *Progressive distillation algorithm. Figure from [13].*

to rise in popularity. We will analyse diffusion guidance and conditioning of different denoiser backbones in this chapter.

### 3.4.1 Diffusion Models Beat GANS in Image Synthesis

Dhariwal et al. [14] searched the architecture space and scaled diffusion models for large image generation and proposed a novel method for trading diversity with fidelity in the sampling process called classifier guidance.

Based on the prior work of [105, 106, 11] they ablated the design space of the U-Net architecture trying global attention layers in different resolutions, multiple residual blocks and timestep and class embedding injections with adaptive group normalization based on the work of [109]. Furthermore, they also used the up-scalling stack found in BigGAN by [110] for large image generation tasks such as LSUN dataset [111] or the large imagenet images.

The second contribution of this work is classifier guidance, in essence by training a noise conditional image classifier and using its gradients we can guide the diffusion process to sample from specific modes of the distribution. As seen and in Figure 3.6 by incorporating a conditional classifier we can enforce direct conditional sampling.

Finally, state of the art results in multiple datasets were reported by the authors showcasing that the improvements proposed have a significant impact. Classifier guidance

**Algorithm 1** Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale $s$.

Input: class label $y$, gradient scale $s$
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
**for all** $t$ from $T$ to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
**end for**
**return** $x_0$

**Algorithm 2** Classifier guided DDIM sampling, given a diffusion model $\epsilon_\theta(x_t)$, classifier $p_\phi(y|x_t)$, and gradient scale $s$.

Input: class label $y$, gradient scale $s$
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
**for all** $t$ from $T$ to 1 **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1-\bar{\alpha}_t}\, \nabla_{x_t} \log p_\phi(y|x_t)$
  $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left( \frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$
**end for**
**return** $x_0$

**Figure 3.6.** *Classifier guidance for different sampling strategies. Figure from [14].*

although an expensive procedure, since you have to jointly train a second network on the dataset thus increasing computational costs, was a first attempt at exploring the capabilities of these models on conditional generation.

### 3.4.2 Classifier-Free Diffusion Guidance (CFG)

CFG by Ho et al. [15] is a seminal paper that proposed a method to avoid training a separate classifier (Classifier guidance) to guide diffusion models as previously proposed by [14]. Classifier free guidance refers to training a conditional and unconditional diffusion model and then combining them to sample conditionally from the target distribution:

$$\tilde{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = (1+w)\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\epsilon_\theta(z_\lambda)$$

Theoretically the above formula can be derived by considering an implicit classifier if we had access to the exact scores (denoted by *):

$$\nabla_{z_\lambda} \log p_i(c|z_\lambda) = -\frac{1}{\sigma_\lambda}[\epsilon^*(z_\lambda, c) - \epsilon*(z_\lambda)]$$

Now, by using this classifier to do classifier guidance we derive to the score estimation formula:

$$\epsilon^*(z_\lambda, c) = (1+w)\epsilon^*(z_\lambda, c) - w\epsilon^*(z_\lambda)$$

We can train both conditional and unconditional diffusion models using a single neural network and use some null class embedding when training the unconditional model, see Figure 3.7 for more details. Authors also note that we can interpret classifier free guidance

**Algorithm 1** Joint training a diffusion model with classifier-free guidance

**Require:** $p_{\text{uncond}}$: probability of unconditional training

**repeat**
  $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$ ▷ Sample data with conditioning from the dataset
  $\mathbf{c} \leftarrow \varnothing$ with probability $p_{\text{uncond}}$ ▷ Randomly discard conditioning to train unconditionally
  $\lambda \sim p(\lambda)$ ▷ Sample log SNR value
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$ ▷ Corrupt data to the sampled log SNR value
  Take gradient step on $\nabla_\theta \left\| \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon} \right\|^2$ ▷ Optimization of denoising model
**until** converged

**Figure 3.7.** *Classifier free guidance training algorithm. Figure from [15].*

as trading off Inception (IS) and FID scores as we vary the guidance weight or by trading mode coverage and sample fidelity. The strong point of this approach lies in the training scheme where little to no changes have to be applied to be able to sample conditionally, due to this advantage it has become an essential method when using diffusion models.

## 3.5 Conditioning Diffusion Models

### 3.5.1 Conditioning with Cross Attention

#### Unet Architecture

The U-Net architecture, originally introduced by Ronneberger [16], stands as a pivotal framework, particularly within the domain of diffusion models. This architectural design leverages the structure of an Encoder-Decoder Convolutional Network with residual connections that facilitate the transfer of features from the encoder to the decoder.

The primary objective of the encoder is to extract features at multiple scales and subsequently downsample them, capturing increasingly abstract, high-level features. The latent representation, situated in the bottleneck and comprising these abstract features, is then relayed to the decoder. By passing, either concatenating or adding, the encoder feature maps through the residual connections we enforce information from different scales to contribute to the task. Full Initial architecture in Figure 3.8.

Notably, the U-Net architecture has demonstrated its efficacy in various applications, with a prominent example being medical image segmentation. Even when confronted with limited datasets, U-Net exhibits impressive performance, showcasing its adaptability and robustness.

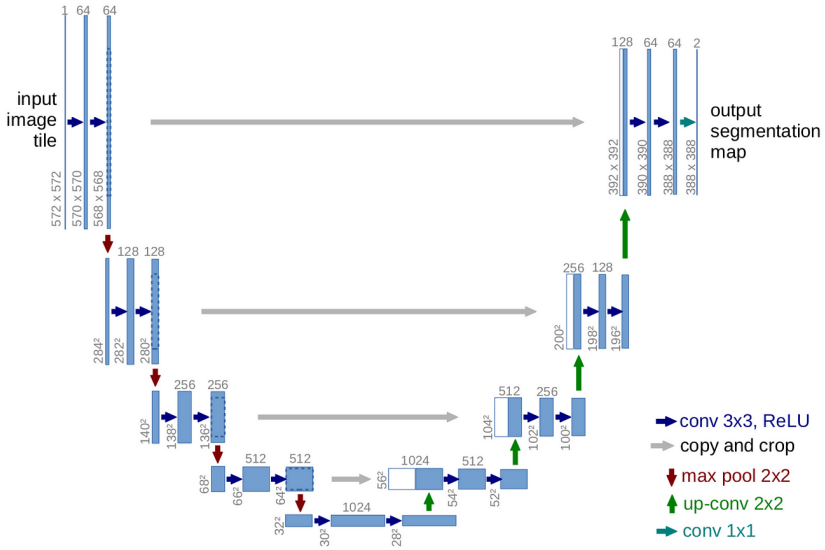


**Figure 3.8.** *The U-Net architecture used in the first publication. Figure modified from [16].*

### Stable Diffusion

Rombach elt al. [17] proposed Latent Diffusion Models by leveraging autoencoders to perceptually compress data and then apply diffusion method in the latent space. Furthermore a conditioning mechanism was proposed in order to generate samples conditioned on other modalities. Figure 3.9 shows the whole architecture, starting from the perceptual encoder we compress the input image after that we proceed to a diffusion process in the latent space. The U-Net backbone that is mostly used in the literature is modified to include a cross attention layer that token-based encoded modalities can be included to condition the process.

More specifically to condition on another modality $y$ the authors use a modality specific encoder $\tau_\theta(y)$ that projects y to an intermediate representation $\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$, which is then mapped to the intermediate layers of the UNet via a cross-attention layer implementing

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

with $Q = W_Q^{(i)} \cdot \phi_i(z_t)$, $K = W_K^{(i)} \cdot \tau_\theta(y)$, and $V = W_V^{(i)} \cdot \tau_\theta(y)$. Where $\phi_i(z_t) \in \mathbb{R}^{N \times d_i}$ denotes the flattened intermediate representation of the U-Net and $W_V^{(i)} \in \mathbb{R}^{d \times d_i}$, $W_Q^{(i)} \in \mathbb{R}^{d \times d_\tau}$, and $W_K^{(i)} \in \mathbb{R}^{d \times d_\tau}$ are learnable projection matrices.

The diffusion training objective remains unchanged and the only difference is the use of the noised latent representation instead of the pixel space one:

$$\mathcal{L}_{\text{LDM}} := \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2$$

The authors managed to accelerate diffusion generation and training by implementing the ideas above and made diffusion models more accessible. Lastly the conditioning mechanism showcased great results and is commonly used in the literature.

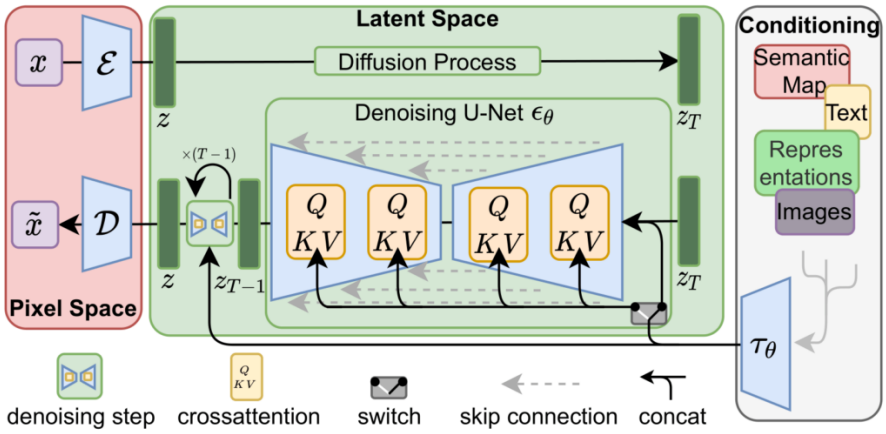


**Figure 3.9.** *Diffusion model architecture used in Stable Diffusion utilizing a U-net with conditioning in another modality through fusion. Figure from [17].*

### 3.5.2 Transformer based diffusion models

#### Diffusion Transformer

Peebles et al. [24] and Bao et al. [19] proposed a different backbone as the denoising network in diffusion models. Literature until their work used only U-Net based architectures and their scalability was limited. Inspired by the work of [17] on latent diffusion models, [49] on adaptive layer normalization and ResNet [112] the Diffusion Transformer (DiT) architecture was proposed. Since scalability is at the forefront the authors didn't diverge from the classic Vision ViT design ([113]) so most of the components remain unchanged.

Mainly, the conditioning mechanism of the backbone is the most core part of this work, the authors tried a variety of methods commonly found in the literature. In Figure 3.10 you can see in more detail each one. We will mainly analyze adaLN-Zero block since it had the best performance, for more analysis on the rest please refer to their work.

As seen in Figure 3.10 the Diffusion Transformer Block is basically a ViT block with an adaptive layer normalization and dimensional scaling parameters $\alpha$ that are computed by the context and timestep representations. The conditioning mechanism is called adaptive layer normalization zero initialized (adaLN-Zero) block. Basically an adaptive normalization layer with zero initialized scale factor $\gamma$, this is done to accelerate large-scale training, this technique is also found in the diffusion U-Net models by zero initializing the final convolutional layer. All the scaling and normalization parameters are regressed using an MLP.

The resulting models outperformed all current SOTA approaches managing great FID scores with fewer compute resources needed (Gflops as stated by the authors). A comprehensive study of scaling and performance was given by the authors since a main concern for these models is scalability, for more details refer to their work.

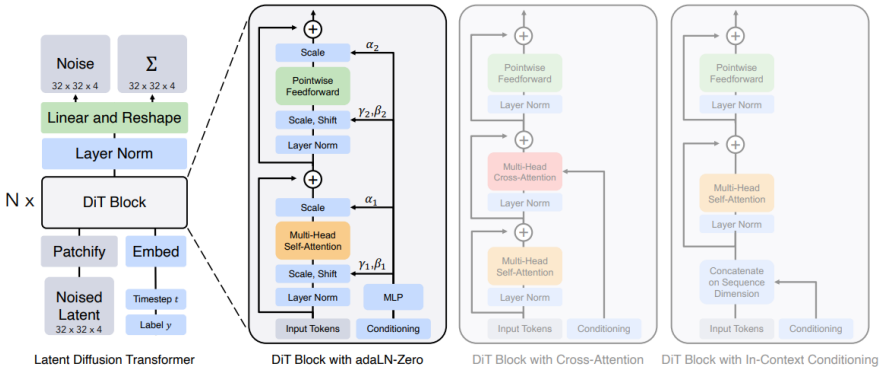


**Figure 3.10.** *Diffusion Transformer architectures with all the different proposed conditioning mechanisms. Figure from [5].*

### Multimodal Diffusion Transformer

Recent progress in rectified flow and transformer-based generative models has motivated the design of architectures that can unify large-scale multimodal generation. Esser et al. [4] propose the *Multimodal Diffusion Transformer (MMDiT)*, a flexible architecture that extends the principles of diffusion modeling into a transformer backbone equipped with feature-wise modulation mechanisms.

**Motivation.** Traditional diffusion U-Nets are highly effective but face scalability limitations when extended to very high-resolution images or multimodal tasks that require flexible conditioning. Transformers, on the other hand, offer better scalability with respect to model size and training data, as well as natural compatibility with sequential

multimodal inputs such as text, audio, or image patches. MMDiT combines the advantages of diffusion and transformers while introducing a structured mechanism for multimodal conditioning.

**Architecture.** MMDiT leverages *Feature-wise Linear Modulation (FiLM)* [49] to inject conditioning signals into the diffusion transformer blocks. For each input sequence token $x_i$, the conditioning flow generates scale and shift parameters $(\gamma_i, \beta_i)$ that are applied element-wise:

$$\text{Modulation}(x_i) = \gamma_i \odot F_{i,c} + \beta_i,$$

where $F_{i,c}$ denotes the conditioned feature representation and $\odot$ is element-wise multiplication. This mechanism enables a fine-grained interaction between timestep embeddings, input features, and conditioning modalities.

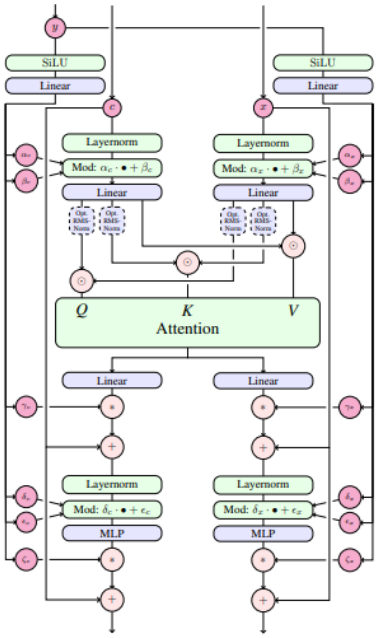


**Figure 3.11.** *Multimodal Diffusion Transformer block. Input and conditioning sequences are first modulated independently, then concatenated for joint self-attention. Additional modulation layers propagate conditioning signals through the network. Figure from [4].*

After modulation, the conditioned and input sequences are concatenated and passed through a multi-head self-attention block. Additional modulation layers are then applied to refine the conditioning signal across multiple depths of the transformer. Figure 5.4 illustrates the architecture of a single MMDiT block.

### UniDiffuser

Recent advances in diffusion models have enabled powerful joint generative modeling across multiple modalities. A prominent example is *UniDiffuser* [18], which introduces a unified diffusion framework capable of simulating joint, conditional, and marginal distributions of heterogeneous modalities within a single model.

**Unified Joint Modeling** Unlike conventional multimodal systems that require separately trained models for each distribution (e.g., $p(x|y)$, $p(y|x)$, or unconditional $p(x)$), UniDiffuser parameterizes all three with one diffusion backbone See figure 3.12. By leveraging a shared latent space, the model learns the joint distribution $p(x, y)$ directly and can perform:

- **Conditional generation:** e.g., generating text given an image, or vice versa.
- **Unconditional generation:** sampling either modality independently.
- **Joint generation:** producing aligned multimodal pairs simultaneously.

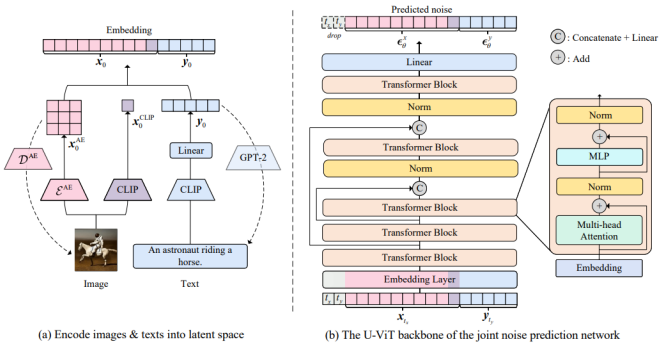


**Figure 3.12.** *Implementation of UniDiffuser [18] on image-text data. (a) First, images and texts are encoded into latent space. (b) Second, we train UniDiffuser parameterized by a transformer [19] in the way illustrated in Figure 2 on the latent embeddings.*

**Training Objective** The model employs a noise-conditional score network with modality-specific conditioning mechanisms. During training, input pairs are randomly masked (one or both modalities), enabling the diffusion process to learn to recover missing modalities or generate consistent multimodal samples. Formally, given a data pair $(x, y)$, the denoising score function $s_\theta$ is trained across three regimes:

$$\mathcal{L}_{\text{UniDiffuser}} = \mathbb{E}_{t,(x,y)}\left[\|s_\theta(x_t, y_t, t_x, t_y) - \nabla_{(x,y)} \log p_t(x, y)\|^2\right]$$

where $(x_t, y_t)$ denotes the noised inputs at timestep $t$.

Chapter 4

# Multimodal Deep Learning and the Missing Modality problem

## 4.1 Introduction

The ability to process and integrate information from multiple sensory channels is fundamental to human intelligence. We naturally combine visual, auditory, and textual cues to understand our environment in ways that far exceed the capabilities of any single modality alone [114]. This observation has motivated decades of research in multimodal learning, culminating in the deep learning era where neural architectures can automatically learn complex cross-modal representations [115].

However, the promise of multimodal systems is often undermined by a practical reality: real-world data is messy, incomplete, and unreliable. Sensors fail, network connections drop, and data collection pipelines break down. This leads to the *missing modality problem*, where systems trained on complete multimodal data must operate with only partial information [116]. This chapter examines the fundamental challenges of multimodal learning and the emerging solutions for handling missing modalities, with particular focus on recent advances in generative recovery approaches.

## 4.2 Multimodal Fusion: From Concatenation to Attention

### 4.2.1 The Challenge of Heterogeneous Data Integration

Multimodal learning fundamentally differs from single-modal approaches due to the heterogeneous nature of different data types 4.1. Consider the task of emotion recognition: audio features might capture prosodic information through spectrograms, visual features encode facial expressions via convolutional or vision-transformer networks, and

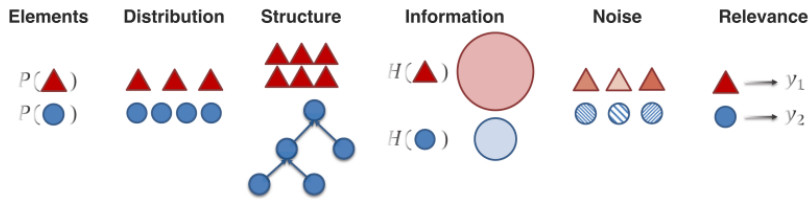


**Figure 4.1.** *Different task relevance and heterogeneity across modalities. Figure from [20].*

textual features represent semantic content through word embeddings or contextual encoders such as BERT [117, 118]. Each modality operates in different dimensional spaces, exhibits distinct noise characteristics, and contributes varying amounts of discriminative information. For example, the language modality is often the most semantically dense, whereas acoustic signals may convey subtler cues that are more ambiguous or context-dependent.

The core challenge lies not only in effectively processing these modalities individually, but also in learning meaningful *interactions* between them. These interactions may be complementary (where modalities provide additional cues), redundant (overlapping information), or even conflicting (where one modality introduces noise). A successful multimodal framework must therefore integrate signals in a way that enhances discriminative power while being robust to noise, misalignment, and missing data.

Early attempts in multimodal fusion treated the task as a straightforward engineering problem, often relying on simple strategies such as concatenation of features from different modalities. However, such approaches ignore the complex statistical dependencies between modalities and often lead to suboptimal performance due to the "curse of dimensionality," modality imbalance, or the inability to model higher-order interactions [119].

### 4.2.2 Evolution of Fusion Strategies

The field has since evolved through multiple generations of fusion approaches, each addressing the limitations of its predecessors:

**Early Fusion** integrates raw or low-level features from different modalities into a single representation prior to further processing [120]. While computationally simple and appealing for shallow models, early fusion assumes perfect temporal and semantic alignment between modalities. Moreover, concatenation dramatically increases the input dimensionality, making learning more difficult and less robust. Crucially, this approach also prevents the model from learning modality-specific representations before integra-

tion, often leading to information loss.

**Late Fusion** adopts the opposite perspective: each modality is processed independently, and the outputs are combined at the decision or prediction stage [121]. This strategy has the advantage of preserving modality-specific processing pipelines, which can be tailored to the strengths of each modality. However, because the modalities only interact at the final stage, late fusion misses the opportunity to exploit cross-modal dependencies during feature extraction. As a result, it performs well when modalities provide largely independent evidence but struggles in tasks—such as sentiment or emotion recognition—where subtle cross-modal interactions carry essential information.

**Hybrid Fusion** emerged as a compromise between the two extremes, combining modalities at multiple intermediate levels of abstraction [122]. For instance, features may be fused at both the representation and decision stages, allowing the network to capture some degree of cross-modal interaction while still preserving modality-specific structure. However, determining the optimal fusion points and strategies often requires domain expertise and extensive hyperparameter search. Furthermore, hybrid approaches often lack flexibility when faced with missing or corrupted modalities.

### 4.2.3 The Attention Revolution and Modern Fusion Approaches

The true breakthrough in multimodal fusion came with the introduction of **attention mechanisms**, which allowed models to dynamically weight information across modalities depending on context [123]. Instead of statically combining features, attention enables fine-grained, content-dependent integration. In particular, *cross-modal attention* mechanisms allow one modality (e.g., language) to query and selectively extract relevant information from another (e.g., vision), thereby modeling conditional dependencies more explicitly.

**Transformer-based architectures** extended this principle by introducing multi-head attention, which captures different types of relationships between modalities simultaneously [71, 124]. For example, one head might focus on aligning temporal signals in audio with textual cues, while another might capture visual-semantic correspondences. This multi-perspective mechanism dramatically increased the expressiveness and flexibility of multimodal fusion, leading to state-of-the-art performance in many downstream tasks, from visual question answering to emotion recognition.

Building on these foundations, recent advances have pushed fusion beyond supervised alignment by leveraging **contrastive learning** and **generative modeling**. Contrastive multimodal models such as CLIP [125] learn a shared embedding space for vision and language by pulling paired samples together and pushing apart unpaired ones. This enables

zero-shot transfer and robust cross-modal retrieval without requiring explicit supervision. Similarly, diffusion-based generative models have demonstrated how multimodal alignment can emerge naturally from joint generation objectives [126, 18].

In parallel, multimodal transformers such as MMBT, VisualBERT, and more recently PaLI and Flamingo, show how large-scale pretraining on paired multimodal corpora can produce flexible and generalizable representations [127, 128]. These models are capable of handling not only fusion but also imputation, cross-modal transfer, and even few-shot learning scenarios, significantly expanding the scope of multimodal learning.

This introduction section of multimodal learning is adapted from the dissertation of Efthymi Georgiou [129].

## 4.3 The Missing Modality Problem

### 4.3.1 Problem Formulation and Real-World Implications

The missing modality problem arises from the fundamental mismatch between training and deployment conditions. Multimodal systems are typically trained on carefully curated datasets where all modalities are present and well-aligned. However, deployment scenarios rarely offer such luxury. In healthcare, certain medical imaging modalities may be unavailable due to equipment failures or patient contraindications [130]. In autonomous driving, sensors may fail due to weather conditions or hardware malfunctions. In social media analysis, users may post text without images or videos without captions.

Formally, given a set of modalities $\mathcal{M} = \{M_1, M_2, \ldots, M_K\}$ with corresponding feature representations $\mathbf{X} = \{x^{(1)}, x^{(2)}, \ldots, x^{(K)}\}$, the missing modality problem occurs when only a subset $\mathcal{M}_{obs} \subset \mathcal{M}$ is available. The system must learn a function $f : \mathcal{X}_{obs} \to \mathcal{Y}$ that performs comparably to the complete function $f : \mathcal{X} \to \mathcal{Y}$.

### 4.3.2 Naive Approaches and Their Limitations

Initial attempts to address missing modalities relied on simple strategies that quickly revealed their inadequacy:

**Zero Imputation** replaces missing modalities with zero vectors. This approach is computationally trivial but introduces artificial patterns that confuse learned representations.

**Mean Imputation** substitutes missing modalities with dataset means. While slightly better than zero imputation, it eliminates the natural variance present in real data.

**Modality Dropout** during training attempts to make models robust to missing inputs by randomly masking modalities [131]. However, this approach often leads to models that ignore weaker modalities entirely, reducing overall performance.

These naive approaches fail because they don't address the fundamental issue: missing modalities represent lost information that cannot be simply filled with generic values. They require sophisticated recovery or adaptation mechanisms that preserve the meaningful relationships between modalities.

## 4.4 Deep Generative Approaches to Modality Recovery

The recognition that missing modalities require sophisticated treatment has led to the emergence of generative recovery approaches. These methods attempt to reconstruct missing modalities using information from available ones, leveraging advances in generative modeling to produce realistic substitutes.

### 4.4.1 Distribution-Consistent Recovery with Normalizing Flows

Wang et al. [21] identified a critical limitation in previous recovery approaches: the distribution gap between generated and real modalities. Their DiCMoR (Distribution-Consistent Modal Recovery) framework addresses this issue through a principled approach based on normalizing flows.

The key insight is that successful modality recovery requires maintaining not just perceptual quality, but distributional consistency. Previous methods often generated plausible-looking outputs that nonetheless exhibited subtle distribution shifts, leading to degraded performance in downstream tasks.

DiCMoR employs a three-stage pipeline:

1. **Feature Extraction**: Shallow encoders project observed modalities into a common feature space, ensuring alignment while preserving modality-specific information.

2. **Latent Mapping**: Normalizing flows learn bijective mappings between modality representations and a shared latent space. This ensures that the transformation preserves the full distribution of each modality.

3. **Recovery Generation**: An aggregation mechanism combines latent representations of observed modalities to generate the latent representation of the missing modality. The reverse flow then produces the recovered modality in the original feature space.

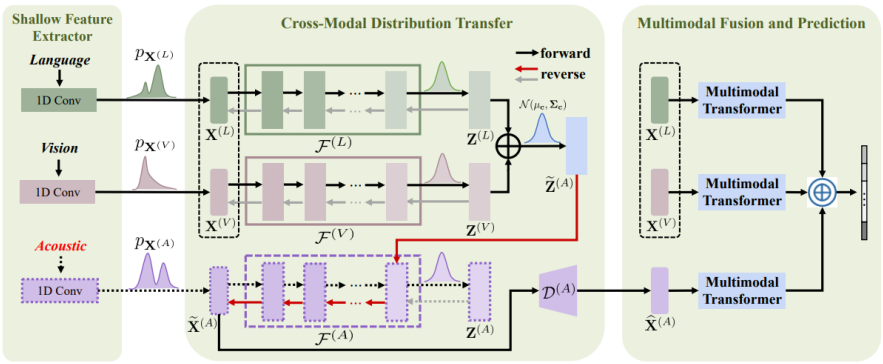


**Figure 4.2.** *DiCMoR architecture for acoustic modality recovery. The framework uses normalizing flows to ensure distribution consistency between generated and real modalities. Figure from [21].*

The training objective balances multiple goals:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \beta(\mathcal{L}_{rec} + \mathcal{L}_{cdt}) \tag{4.1}$$

where $\mathcal{L}_{task}$ ensures good downstream performance, $\mathcal{L}_{rec}$ measures reconstruction quality, and $\mathcal{L}_{cdt}$ enforces class-aware distribution consistency.

### 4.4.2 Diffusion Models for Modality Generation

Building on the success of diffusion models in image generation [11], researchers have adapted these approaches for missing modality recovery [3]. Diffusion-based recovery offers several advantages over flow-based methods:

**Generation Quality**: Diffusion models have demonstrated superior generation quality across various domains, producing more realistic and diverse outputs.

**Conditional Generation**: The denoising process naturally accommodates conditioning information, allowing fine-grained control over the generation process based on available modalities.

**Robust Training**: Unlike adversarial approaches, diffusion models exhibit stable training dynamics without mode collapse or training instabilities.

The diffusion-based approach maintains the same architectural principles as DiCMoR while replacing normalizing flows with a diffusion backbone. Cross-attention mechanisms enable conditioning on observed modalities, similar to techniques used in stable diffusion [132].

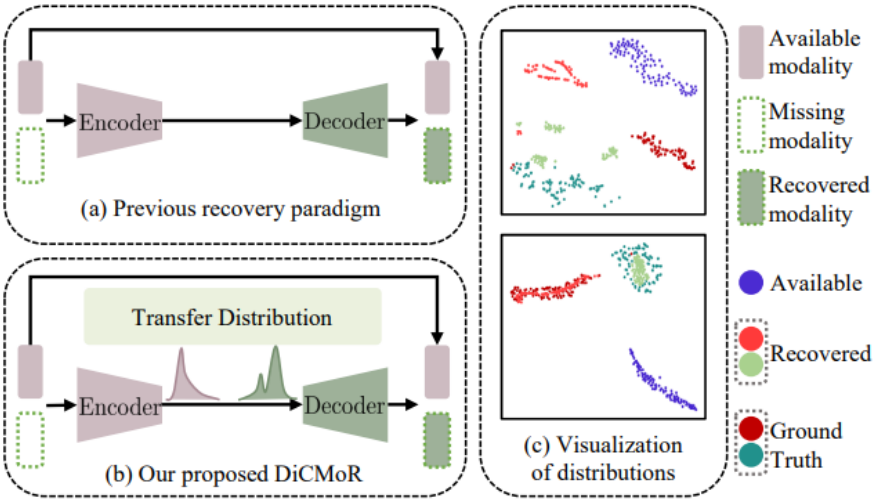


**Figure 4.3.** *Evolution of modality recovery paradigms. Figure from [21]. (a) Traditional encoder-decoder approaches. (b) Distribution-consistent transfer paradigm. (c) Visualization showing improved distribution alignment with DiCMoR compared to previous methods.*

## 4.5 Feature-Level Recovery Strategies

While generative approaches attempt to recover missing modalities in their original representation space, an alternative paradigm operates at the feature level, learning shared representations that can substitute for missing inputs.

### 4.5.1 Cross-Modal Imagination for Unified Missing Modality Handling

A critical limitation of previous missing modality approaches lies in their specificity: different models must be trained for each possible missing modality configuration. This scalability problem becomes particularly acute in multimodal systems with three or more modalities, where the number of possible missing combinations grows exponentially. Zhao et al. [6] address this challenge through their Missing Modality Imagination Network (MMIN), which introduces a unified framework capable of handling arbitrary missing modality patterns during both training and inference.

MMIN addresses this through a unified triplet input format $(x^{(a)}, x^{(v)}, x^{(t)})$ where missing modalities are replaced with zero vectors during both training and inference, See Figure 4.5. This standardization allows a single model to handle all possible missing modality combinations while learning robust cross-modal representations.

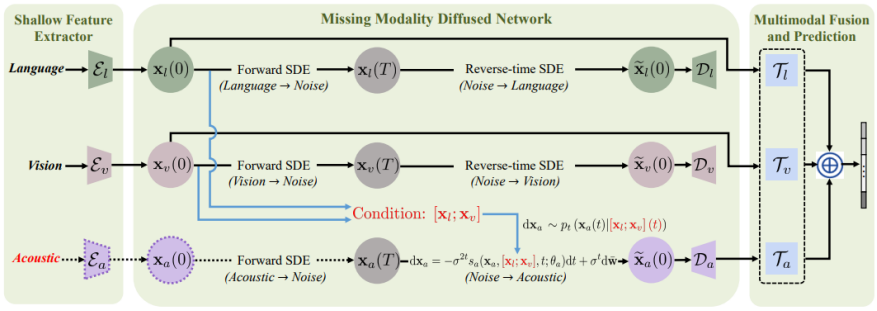


**Figure 4.4.** *Diffusion-based modality recovery network. The approach adapts conditional diffusion models for cross-modal generation tasks. Figure from [3].*

### Cross-Modal Imagination Architecture

The core innovation of MMIN lies in its imagination module, which explicitly models the process of inferring missing modality representations from available ones. The architecture consists of three main components:

**Modality Encoder Network** extracts sentence-level embeddings for each modality using specialized encoders: LSTM networks for temporal acoustic and visual features, and TextCNN for textual content. These encoders are first pre-trained on complete multimodal data and then fine-tuned within the unified framework.

**Imagination Module** employs Cascade Residual Autoencoders (CRA) [66] to perform cross-modal inference. Given available modality embeddings $h_{available}$, the module predicts missing modality representations through a series of residual transformations:

$$\Delta z^k = \begin{cases} \phi^k(h_{available}) & \text{if } k = 1 \\ \phi^k(h_{available} + \sum_{j=1}^{k-1} \Delta z^j) & \text{if } k > 1 \end{cases} \tag{4.2}$$

The final imagined representation combines the input with all residual outputs: $h_{imagined} = h_{available} + \sum_{k=1}^{B} \Delta z^k$.

**Cycle Consistency Learning** ensures bidirectional imagination quality through coupled forward and backward imagination networks. The forward network predicts missing modalities from available ones, while the backward network reconstructs the original available modalities from the imagined complete representation. This bidirectional constraint helps maintain information preservation during the imagination process.

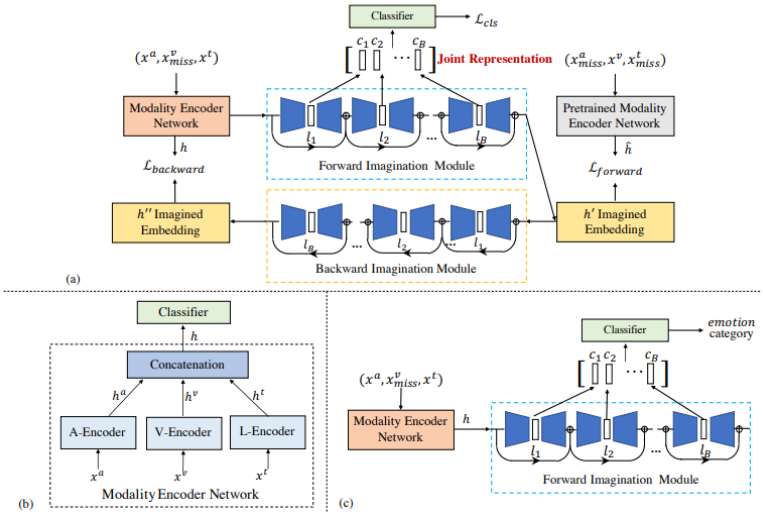


**Figure 4.5.** *MMIN architecture overview. Figure from [6]. (a) Training phase with visual modality missing: the network learns cross-modal imagination using all possible missing modality combinations. (b) Modality encoder structure: pre-trained encoders (gray) remain fixed while updated encoders (orange) are fine-tuned during MMIN training. (c) Inference phase: unified model handles arbitrary missing modality patterns through learned imagination module.*

### Joint Optimization Strategy

MMIN employs a multi-objective loss function that balances downstream task performance with imagination quality:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{forward} + \lambda_2 \mathcal{L}_{backward} \tag{4.3}$$

where $\mathcal{L}_{cls}$ represents the emotion classification loss, $\mathcal{L}_{forward}$ measures the quality of forward imagination (available $\rightarrow$ missing), and $\mathcal{L}_{backward}$ evaluates backward reconstruction (imagined $\rightarrow$ original). The imagination losses use L2 reconstruction error between predicted and ground-truth modality representations.

The joint representation for classification combines latent vectors from all autoencoder stages: $R = \text{concat}(c_1, c_2, \ldots, c_B)$, where $c_k$ represents the latent vector from the $k$-th residual autoencoder. This aggregation captures information at multiple levels of abstraction.

### 4.5.2 Multi-modal Learning with Missing Modality via Shared-Specific Feature Modeling

The shared-specific framework [22] recognizes that multimodal data contains both modality-specific information (unique to each data type) and shared information (common across modalities). By explicitly modeling this decomposition, systems can use shared information from available modalities to compensate for missing ones.

The architecture employs dual encoding pathways:

**Modality-Specific Encoders** $E_s^{(i)}$ extract features unique to modality $i$:

$$h_s^{(i)} = E_s^{(i)}(x^{(i)}) \tag{4.4}$$

**Shared Encoders** $E_{sh}^{(i)}$ extract cross-modal information from each modality:

$$h_{sh}^{(i)} = E_{sh}^{(i)}(x^{(i)}) \tag{4.5}$$

When modality $j$ is missing, its shared representation is approximated by aggregating shared features from available modalities:

$$\hat{h}_{sh}^{(j)} = \frac{1}{|\mathcal{M}_{obs}|} \sum_{i \in \mathcal{M}_{obs}} h_{sh}^{(i)} \tag{4.6}$$

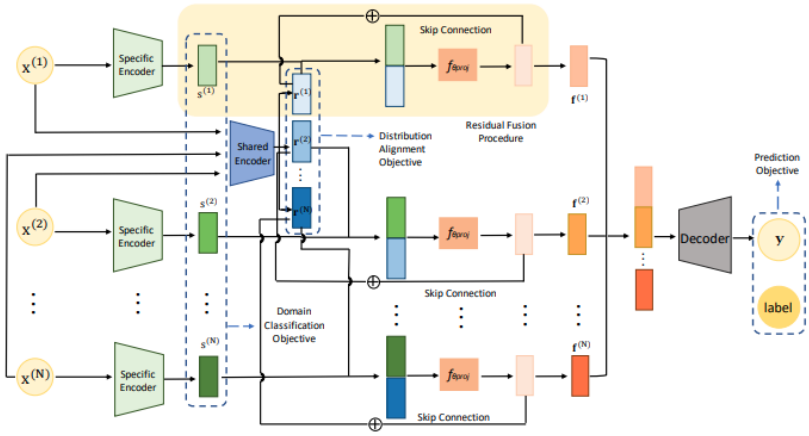


**Figure 4.6.** *Shared-specific feature modeling with complete modalities. Each modality is processed through both specific and shared encoders. Figure from [22]*

The training process uses adversarial objectives to ensure proper separation:

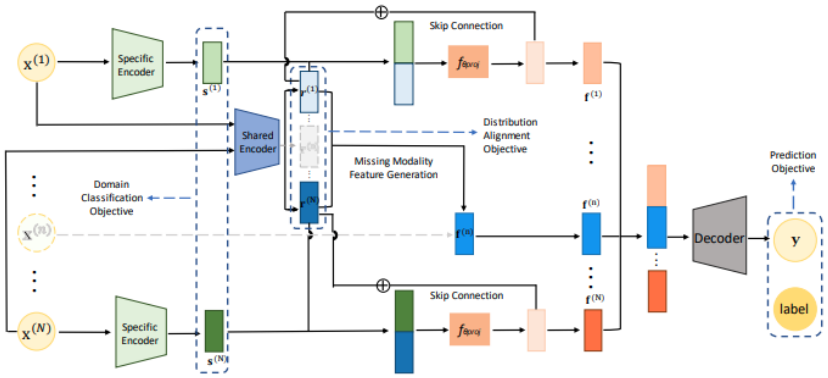


**Figure 4.7.** *Shared-specific feature modeling with missing modalities. Shared features from available modalities substitute for the missing modality's representation. Figure from [22]*

**Domain Classification Loss** encourages modality-specific features to retain sufficient information for modality identification, ensuring they capture unique characteristics of each data type.

**Domain Confusion Loss** promotes shared features that are invariant across modalities through adversarial training, ensuring they capture truly common information.

### 4.5.3 Missing Modalities Imputation via Cascaded Residual Autoencoder

The CRA framework [66] decomposes the imputation process into multiple cascaded stages, each designed to iteratively reduce the discrepancy between the generated and ground-truth modalities. Unlike a conventional single-pass autoencoder, CRA leverages a residual learning scheme where each subsequent autoencoder stage focuses on correcting the reconstruction errors of its predecessor.

#### Stage-Wise Residual Autoencoders

CRA is composed of a sequence of autoencoder modules. The first module generates a coarse reconstruction of the missing modality. Each subsequent module then receives the residual error from the previous reconstruction and learns to correct it. This cascaded residual learning progressively improves the quality of the imputation.

Formally, let $X^{(m)}$ denote the ground-truth missing modality and $\tilde{X}_t^{(m)}$ the recon-

structed modality at stage $t$. The residual update is expressed as:

$$R_t = X^{(m)} - \tilde{X}_t^{(m)},$$

where $R_t$ is the residual at stage $t$. The subsequent autoencoder stage learns a mapping $f_{t+1}$ that predicts $R_t$, yielding the refined reconstruction:

$$\tilde{X}_{t+1}^{(m)} = \tilde{X}_t^{(m)} + f_{t+1}(R_t).$$

This formulation ensures that each stage does not relearn the full modality but instead focuses only on reducing the remaining reconstruction error.

#### Progressive Imputation

The cascaded design stabilizes training and improves convergence. By distributing the reconstruction task across multiple stages, CRA avoids the pitfalls of overfitting and under-reconstruction often observed in single-step autoencoders. Furthermore, the progressive refinement aligns with human perception, where coarse-to-fine processing is frequently observed in multimodal understanding.

### 4.5.4 Learning Robust Joint Representations by Cyclic Translations Between Modalities

The central idea of Pham et al. [29] is that corresponding audio and visual streams carry semantically aligned information when they originate from the same event. By contrasting aligned (positive) and misaligned (negative) audio-visual pairs, the model learns a joint embedding space.

#### Network Architecture

The framework employs modality-specific convolutional neural networks:

- A vision network processes image frames or spatio-temporal features from video segments.
- An audio network encodes short-term spectrogram representations of sound.

Outputs are projected into a common embedding space, where their similarity is measured.

**Training Objective**

The system is trained using a binary classification objective: predict whether an audio-visual pair is synchronized. Formally, given embeddings $v$ (visual) and $a$ (audio), their similarity $s(v, a)$ is computed, and the model minimizes the cross-entropy loss:

$$\mathcal{L} = -\big(y \log \sigma(s(v,a)) + (1-y)\log(1-\sigma(s(v,a)))\big),$$

where $y = 1$ for aligned pairs and $y = 0$ otherwise.

## 4.6 Noise-Robust Representations

Fan et al. [133] observe that the "missing" modality problem often represents an extreme case of data corruption rather than complete absence. Their approach treats missing modalities as heavily corrupted inputs and employs variational autoencoders for joint denoising and recovery.

This perspective shifts the problem from discrete missing/present states to a continuous spectrum of data quality. The VAE framework provides principled uncertainty estimation, allowing systems to assess the reliability of both observed and recovered information. This approach shows particular promise in scenarios where "missing" modalities are actually present but heavily degraded.

## 4.7 Domain-Specific Applications and Insights

### 4.7.1 Medical Imaging: Handling Clinical Constraints

Medical imaging presents unique challenges and opportunities for missing modality recovery. In MRI imaging, multiple sequences (T1, T2, FLAIR) provide complementary diagnostic information, but acquisition time, cost, and patient comfort often limit which sequences can be obtained [130].

CoLa-Diff addresses these challenges through several medical-specific innovations:

**Anatomical Consistency**: Brain region masks guide the generation process, ensuring recovered modalities respect known anatomical structures.

**Clinical Validation**: Generated modalities must not only look realistic but preserve diagnostic information relevant to clinical decision-making.

**Computational Efficiency**: Operating in latent space reduces memory requirements, crucial for high-resolution medical images.

The medical domain highlights the importance of domain expertise in modality recovery. Generic generation approaches may produce visually plausible results that lack clinical validity, emphasizing the need for domain-specific constraints and evaluation metrics.

### 4.7.2 Conversational AI: Dynamic Modality Availability

Conversational multimodal systems face unique challenges where modality availability changes dynamically throughout interactions. Graph neural network approaches model these scenarios by representing conversations as dynamic graphs where nodes represent utterances and edges capture temporal and speaker relationships [134].

The graph structure naturally accommodates missing modalities through information propagation. When visual information is unavailable for certain utterances, graph convolutions can propagate relevant information from neighboring nodes where visual data is present.

Chapter 5

# Methodology

## 5.1 Proposed Methodology

### 5.1.1 General Formulation

Let $\mathcal{X} = \{x_l, x_v, x_a\}$ denote the input space of three modalities—language ($x_l$), vision ($x_v$), and audio ($x_a$)—corresponding to a single utterance. In the Multimodal Emotion Recognition (MER) task, the goal is to learn a function $\mathcal{F} : \mathcal{X} \to y$ that maps the observed modalities to a discrete or continuous emotion label $y$.

In real-world applications, however, not all modalities may be present at inference time. Let $\mathcal{M} \subseteq \{l, v, a\}$ denote the set of modalities observed in a given sample, and $\mathcal{M}^c$ its complement—the set of missing modalities. The central challenge is to infer or approximate the missing elements $\{x_m : m \in \mathcal{M}^c\}$ conditioned on the available ones $\{x_o : o \in \mathcal{M}\}$, such that downstream emotion classification remains robust.

Formally, we aim to model the conditional distributions:

$$p_\theta(x_m(0) \mid x_o(0)), \quad \text{for all } m \in \mathcal{M}^c, \tag{5.1}$$

where $x(0)$ refers to clean data (i.e., at time $t = 0$ in the diffusion process). Once the missing modalities are sampled or imputed, we pass both observed and reconstructed modalities to a fusion model $\mathcal{T}_k$ for final prediction:

$$\hat{y} = \mathcal{T}_k(x_l^*, x_v^*, x_a^*), \tag{5.2}$$

where $x_m^* = x_m$ if $m \in \mathcal{M}$ and $x_m^* = \hat{x}_m$ (sampled) if $m \in \mathcal{M}^c$.

To model the conditional distributions $p_\theta(x_m(0) \mid x_o(0))$, we adopt a denoising diffusion probabilistic model (DDPM) framework. Each modality is assigned a separate score network $s_m(\cdot, t)$ trained via score matching to approximate the gradient of the data

distribution at time $t$:

$$s_m(x_m(t), t \mid x_o(t)) \approx \nabla_{x_m} \log p_t(x_m \mid x_o). \tag{5.3}$$

After the imputation of the missing modalities via sampling using the above score function a modality specific alignment decoder $\mathcal{D}_m$ is used to further improve the generated modalities features and finally a downstream fusion classifier $\mathcal{T}$ with modality specific transformers in employed. The whole netowrk can be see in this Figure 5.2, we will analyze each component of the total network.

### 5.1.2 SDE Formulation and Reverse Process

We consider the two most popular diffusion processes **Variance Exploding (VE)** and **Variance Preserving (VP)** stochastic differential equations (SDEs) as described in the seminal work of Song et al. [2]. Both formulations provide continuous-time frameworks for progressive data corruption and subsequent generation through reverse-time processes.

#### Variance Exploding (VE) SDE

The **Variance Exploding (VE)** stochastic differential equation formulation defines a continuous-time forward diffusion process where noise is progressively added to the data sample without altering the signal magnitude:

$$d\mathbf{x} = \sigma(t) d\mathbf{w}, \quad \text{with} \quad \sigma(t) = \sigma^t, \tag{5.4}$$

where $\mathbf{w}$ denotes standard Brownian motion, and $\sigma(t)$ is a time-dependent diffusion coefficient that grows exponentially over time $t \in [0, 1]$. For our experiments, we set $\sigma = 25$ following [1].

The forward SDE defines a family of corrupted distributions $p_t(\mathbf{x})$, where the marginal distribution at time $t$ has standard deviation:

$$\text{std}_t = \sqrt{\frac{\sigma^{2t} - 1}{2 \ln \sigma}}, \tag{5.5}$$

such that $\mathbf{x}(t) = \mathbf{x}(0) + \mathbf{z} \cdot \text{std}_t$ where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, and $\mathbf{x}(t)$ becomes increasingly noisy as $t \to 1$.

The generative process corresponds to simulating the **reverse-time SDE**:

$$d\mathbf{x} = -\sigma(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) dt + \sigma(t) d\bar{\mathbf{w}}, \tag{5.6}$$

where $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is the score function and $\bar{\mathbf{w}}$ denotes reverse-time Brownian motion.

### Variance Preserving (VP) SDE

The **Variance Preserving (VP)** formulation maintains approximately constant variance throughout the diffusion process by simultaneously adding noise and scaling the signal:

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}, \tag{5.7}$$

where $\beta(t)$ is a time-dependent noise schedule. We use a linear schedule $\beta(t) = \beta_0 + t(\beta_1 - \beta_0)$ with $\beta_0 = 0.1$ and $\beta_1 = 20.0$ following standard practice.

The marginal distribution at time $t$ for the VP SDE has standard deviation:

$$\text{std}_t = \sqrt{1 - \exp(2 \log \alpha(t))}, \tag{5.8}$$

where $\log \alpha(t) = -0.25t^2(\beta_1 - \beta_0) - 0.5t\beta_0$ is the log of the signal scaling factor, such that $\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \mathbf{z} \cdot \text{std}_t$ where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$.

The corresponding reverse-time SDE for generation is:

$$d\mathbf{x} = \left[-\frac{1}{2}\beta(t)\mathbf{x} - \beta(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + \sqrt{\beta(t)}d\bar{\mathbf{w}}. \tag{5.9}$$

### Score Function Approximation

For both SDE formulations, we approximate the intractable score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ using a neural network $s_\theta(\mathbf{x}, t)$, trained with denoising score matching. The training objective minimizes:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{z}}\left[\|s_\theta(\mathbf{x}_t, t) \cdot \text{std}_t + \mathbf{z}\|^2\right], \tag{5.10}$$

where $t \sim \mathcal{U}[\epsilon, 1 - \epsilon]$ with $\epsilon = 10^{-5}$, $\mathbf{x}_0$ is clean data, $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ is noise, and $\mathbf{x}_t = \mathbf{x}_0 + \mathbf{z} \cdot \text{std}_t$ is the perturbed sample at time $t$. For conditional generation, we apply the same perturbation to both target and conditioning modalities as discussed in our joint perturbation approach.

The choice between VE and VP formulations depends on the specific application and data characteristics. VE SDEs are often preferred for unconditional generation tasks, while VP SDEs provide more stable training dynamics for conditional generation scenarios like ours.

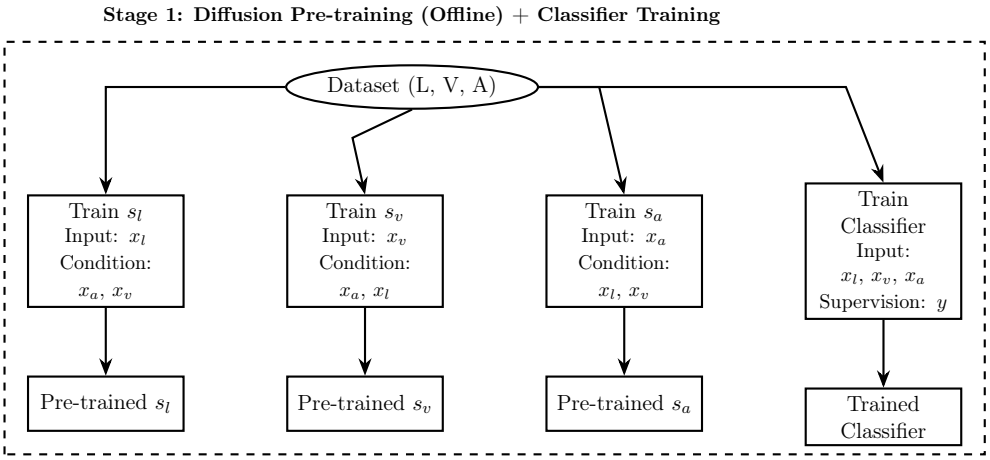


**Figure 5.1.** *Diagram representing the first Stage of Training our decoupled modality diffusion MER network. The modality specific score networks are trained using the full dataset with randomized conditioning, furthermore the downstream classifier is also trained on the full dataset for the MER task with access to fully observable modalities.*

### 5.1.3 Decoupled Generative Training

In our proposed framework, each modality-specific diffusion model is trained independently to estimate its conditional distribution given the other modalities. Unlike joint end-to-end training pipelines such as IMDER [1], we decouple the generative training of each score network from the final downstream classifier as shown in Stage 1 of Figure 5.2. Basically, this allows one to use a fully trained score model to impute values in the downstream task. Unlike the previous approach of IMDER where an end-to-end approach jointly trains the pretrained classifier (trained on the full dataset without missing modalities as show in Figure 5.2) with an untrained diffusion model, resulting with junk imputed modalities and gradients that result in unstable training. Furthermore, this allows efficient and modular training, and facilitates better control over the model architecture, conditioning mechanism, and sampling strategy.

Let $x = \{x_l, x_v, x_a\}$ denote the full multimodal feature tuple for an utterance. To impute a missing modality $x_m \in x$, we aim to model the conditional distribution $p(x_m(0) \mid x_o(0))$, where $x_o(0) \subset x \setminus x_m$ are the available modalities, and $x_m(0)$ is the clean data of the missing modality. However, as shown in [2], one can train a single score network to approximate the gradients of the joint distribution $\nabla_x \log p_t(x)$ through perturbing all variables, not just the target.

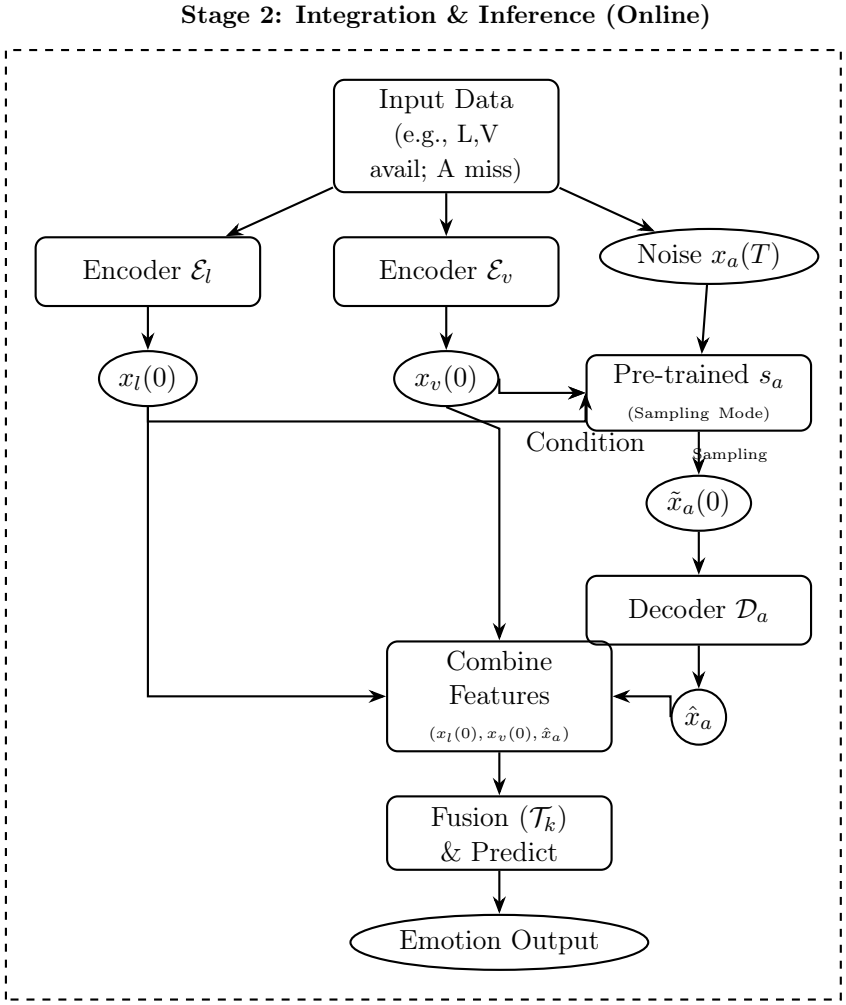


**Figure 5.2.** *Diagram illustrating the proposed Stage 2 for our framework showcasing an example that the acoustic modality is missing. Firstly we sample a noise Latent and condition the score network with the observed modalities. We reverse the diffusion sampling process through our trained audio score network $s_\alpha$ and obtain a rough reconstructed modality $\tilde{x}_\alpha$. After that, we further refine it passing it through our alignment decoder $D_\alpha$ before we use it in downstream emotion inference through our fusion classifier $T_k$.*

**Perturbing both target and conditioning modalities.** Instead of freezing the conditioning modalities and only perturbing the target modality, we inject noise into both the target $x_m$ and the conditioning modalities $x_o$ using the same forward SDE (like in IMDER):

$$d\mathbf{x} = f(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \quad \mathbf{x}(0) \sim p_0(\mathbf{x}). \tag{5.11}$$

This leads to noisy versions $x_m(t) \sim q(x_m(t) \mid x_m(0))$ and $x_o(t) \sim q(x_o(t) \mid x_o(0))$. During training, we provide the noisy conditioning $x_o(t)$ instead of the clean $x_o(0)$, which encourages the score network to learn a true joint distribution over $(x_m, x_o)$ instead of relying on shortcut correlations from clean inputs.

Theoretically, this follows from the formulation in [2] that minimizing the denoising score matching (DSM) loss on all components perturbed by the SDE yields an estimator of the score function $\nabla_x \log p_t(x)$. In our case, this enables modeling of the conditional score function:

$$\nabla_{x_m} \log p_t(x_m \mid x_o) \approx s_\theta(x_m(t), t, x_o(t)), \tag{5.12}$$

where $s_\theta$ is a neural network approximator of the conditional score function. The use of noisy $x_o(t)$ regularizes the learning and allows the model to generalize to a wider range of missingness patterns during inference.

**Training Objective.** Given this formulation, each score network $s_m(\cdot)$ for modality $m \in \{l, v, a\}$ is trained using a DSM objective:

$$\mathcal{L}_m = \mathbb{E}_{x_m(0), x_o(0), t, \epsilon} \left[ \left\| \sqrt{\sigma(t)} s_m(x_m(t), t, x_o(t), \theta_m) + \epsilon \right\|^2 \right], \tag{5.13}$$

where $x_m(t) = \alpha(t)x_m(0) + \sigma(t)\epsilon$, and $x_o(t)$ is constructed analogously. Here $\sigma(t)$ is a weighting function in order to infer the score function from noise as described in [2]. We simulate different missingness configurations at each batch, ensuring that the model learns to condition on variable subsets of the modalities. This approach effectively learns a family of conditional generative models $p_\theta(\mathbf{x}_m \mid \mathbf{x}_o)$ for each modality $m$.

After training, the score networks are frozen and used in inference mode to sample missing modalities (e.g., through reverse SDE or probability-flow ODE samplers). These sampled representations are then passed through a decoder and fused via a downstream transformer for emotion prediction (see Figure 5.2, Stage 2).

### 5.1.4 Score Network Backbones and Conditioning

The architecture of the score networks $s_m$ and its conditioning mechanism are critical design choices that determine both the quality of generated missing modalities and computational efficiency. We systematically evaluate different backbone architectures paired with various conditioning strategies to identify the optimal configuration for multimodal emotion recognition.

#### Conditioning Architectures

We compare four distinct conditioning score network architectures, each representing different approaches to modeling temporal dependencies and cross-modal interactions:

- **U-Net with Cross-Attention (Baseline IMDER):** Following the standard approach in diffusion models [48, 17], this architecture uses convolutional layers for feature extraction with skip connections. Time embeddings projected at intermediate , while conditioning on available modalities is achieved through cross-attention mechanisms as developed in Stable Diffusion [17] (The full network archecture in Figure 5.3).

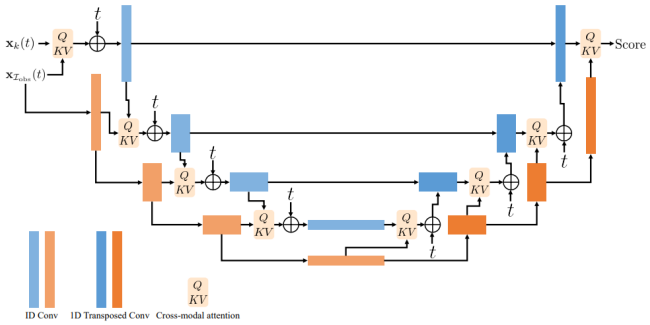


**Figure 5.3.** *Illustration of the network used in our experiments [23], a 4 layer encoder decoder with residual connections unet was used together with cross attention mechanisms on the observed modalities.*

- **Multimodal Diffusion Transformer:** Drawing from Feature-wise Linear Modulation [49] we adopt the architecture proposed by Esser et al. [4]. Conditioning modalities are processed through a modulation flow in each transformer block to fuse the timestep information by generating different scale ($\gamma$) and shift ($\beta$) parameters for input and conditioning modalities that element-wise affect the sequences via $\text{Modulation}(x_i) = \gamma_i \odot F_{i,c} + \beta_i$, where $\odot$ denotes element-wise multiplication

further scaling is also implemented for more conditioning expressivity. Lastly conditioning and input are concatenated in the self attention mechanism to extract information and condition the input and the condition (See Figure 5.4).

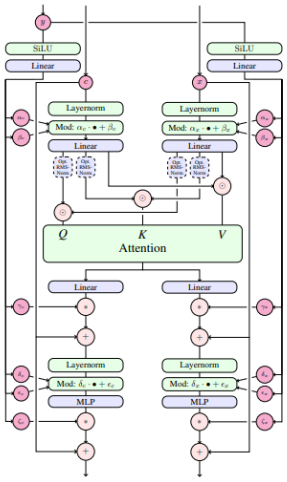


**Figure 5.4.** *Multimodal Diffusion Transformer block [4] modulating conditioning and input sequences separately, then concatenating both for self attention and after that further modulation layers are added for further conditioning.*

- **Diffusion-Transformer:** This variant was proposed by Peebles et al. [5], it utilizes a special case of Adaptive Layer Normalization (AdaLN) conditioning caled AdaLN-zero. AdaLN modulates the scale and shift parameters of layer normalization based on available modalities, providing fine-grained control over conditioning information flow without the computational overhead of additional attention mechanisms. In the Di-Transformer another scaling factor $\alpha$ is injected before every residual connection within the DiT block for further conditioning control (Full mechanism can be see in Figure 5.5). We utilize this architecture in our multimodal scenario and test if AdaLN can perform in such generation tasks.

- **ScoreTransformer1D (Ours):** We propose a novel lightweight transformer-based architecture specifically tailored for 1D feature sequences such as time-aligned multimodal vectors. This architecture is inspired by the Unidiffuser [18], where a transformer model concatenates modalites in as input to the transformer each with its own timestep embedding. We omit the extra timestep embeddings and only inject the timestep to the input since its always common with the timestep of the conditioning. Unlike the convolutional U-Net architectures, ScoreTransformer1D

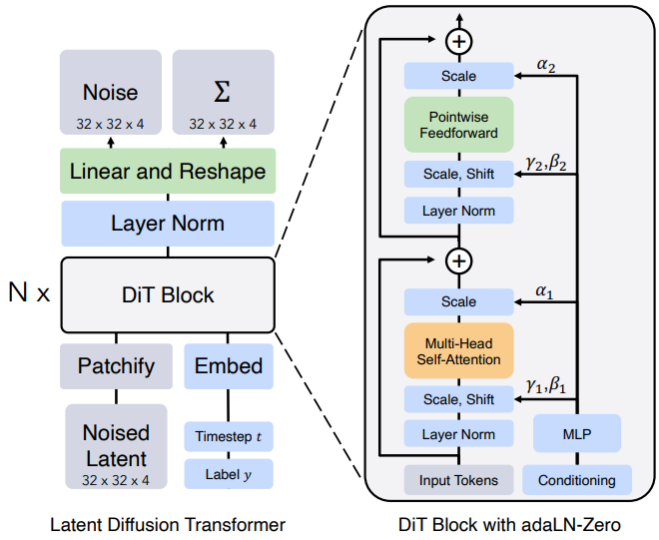


**Figure 5.5.** *The architecture of a single Diffusion Transformer block [24]. One can say that its an improved FiLM conditioning utilizing scaling and shifting after each layer normalization and further scaling factors $\alpha$.*

replaces spatial convolutions with token-mixing transformer blocks that can model long-range temporal dependencies. With concatenation-based conditioning, we are treating all modalities uniformly within a single transformer framework and allow the model to learn cross-modal dependencies through self-attention mechanisms. Figure 5.6 illustrates the core computation of our proposed ScoreTransformer1D. The input consists of three components:

- $x \in \mathbb{R}^{B\times C\times T}$: a noisy feature tensor for the target modality[1]
- $\gamma(t) \in \mathbb{R}^{B\times 1\times D}$: the timestep embedding added channel-wise to $x$
- $c \in \mathbb{R}^{B\times C\times T^c}$: the conditional input composed of concatenated noisy modalities $x_{\setminus m}(t)$

The model first computes the time-conditioned latent representation:

$$\tilde{x} = x + \gamma(t) \tag{5.14}$$

[1] All modalities are projected into a common feature space via a shallow feature encoder.

Then concatenates the conditional input:

$$z = [\tilde{x}; c] \in \mathbb{R}^{B \times C \times (T+T^c)} \tag{5.15}$$

This tensor is processed through a transformer encoder that applies self-attention across channels for each timestep. The output is filtered to extract only the predicted score for the target modality $x_m$, discarding other channels.

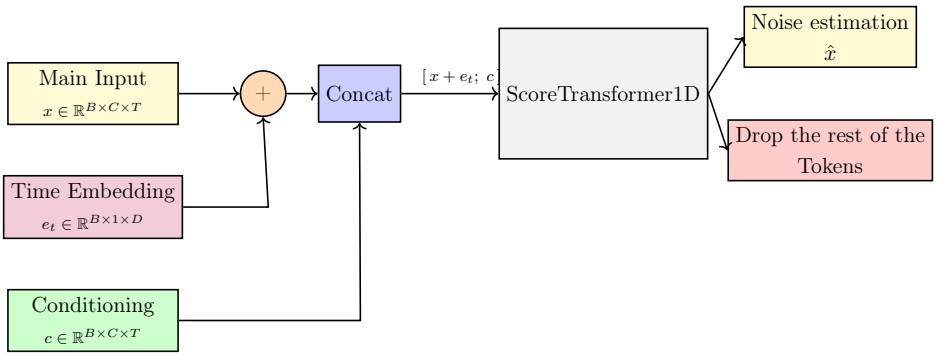


**Figure 5.6.** *High-level data flow in* `ScoreTransformer1D`.

### Timestep Embeddings:

Following prior work [50], we employ Gaussian Fourier time embeddings for all backbones to represent the continuous diffusion timestep $t$ as a high-dimensional periodic signal:

$$\gamma(t) = [\sin(2\pi W t), \cos(2\pi W t)], \quad W \in \mathbb{R}^{D/2} \tag{5.16}$$

where $W$ is a fixed Gaussian matrix and $D$ is the total embedding dimensionality.

### Diffusion Training Scheme

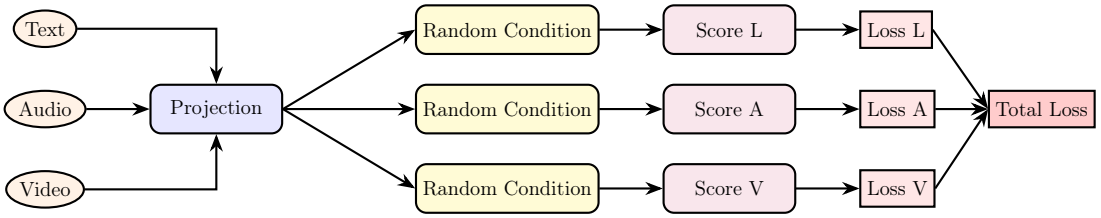


**Figure 5.7.** *High-level diagram of the diffusion model training process, all score nets are trained in a single forward.*

Figure 5.7 illustrates our unified training approach for all backbone architectures. The training process follows a multi-target scheme where all three modality-specific score networks are trained simultaneously in a single forward pass:

1. **Input Processing**: Raw text, audio, and video inputs are processed through modality-specific projection layers to map them into a common feature space.

2. **Random Conditioning**: For each target modality, we randomly omit 1-2 modalities and use the remaining available modalities as conditioning input. This random conditioning strategy ensures that the model learns to handle various missing data patterns during training, improving generalization to different missingness scenarios during inference.

3. **Parallel Score Estimation**: Three separate score networks (Score L, Score A, Score V) simultaneously predict the score functions for language, audio, and video modalities respectively. Each network receives its corresponding noisy target modality along with the randomly selected conditioning modalities.

4. **Joint Loss Computation**: Individual losses are computed for each modality-specific score network using the denoising score matching objective, and these losses are aggregated into a total loss for end-to-end optimization.

This training scheme enables efficient learning of cross-modal dependencies while maintaining computational efficiency through parallel processing. The random conditioning mechanism ensures that each score network learns to leverage different combinations of available modalities, making the system robust to various missing data patterns encountered during inference.

### 5.1.5 Sampling

After training, each modality-specific score network $s_\theta^m$ is used as a pre-trained generative module that reconstructs missing modalities through the learned reverse-time SDE. During inference, we discard the forward diffusion pass used during training and instead apply dedicated numerical samplers to solve the reverse SDE:

$$d\mathbf{x}_m = -\sigma(t)^2 s_\theta^m(\mathbf{x}_m, t \mid \mathbf{x}_{\setminus m})dt + \sigma(t)d\bar{\mathbf{w}}. \tag{5.17}$$

The choice of sampling algorithm significantly impacts both generation quality and computational efficiency. We explore four distinct sampling strategies that represent different trade-offs between accuracy and speed:

- **Euler-Maruyama** — The default stochastic sampling method used in Score-SDE [2]. While providing high-quality samples, it is computationally expensive, typically requiring 100+ function evaluations (NFEs) for convergence.

- **Predictor-Corrector (PC)** — Combines a predictor step (Euler-Maruyama) with corrector steps using Langevin dynamics for sample refinement [2]. This approach offers improved sample quality at the cost of additional computational overhead per timestep.

- **Heun** — A second-order stochastic sampler proposed by Karras et al. [42] that achieves effective sampling with significantly fewer steps, often requiring as few as 30 NFEs while maintaining competitive sample quality through higher-order numerical integration.

- **DDIM** — The Denoising Diffusion Implicit Models sampler [40] provides deterministic, ODE-based sampling that can dramatically reduce the number of required steps. DDIM enables fast sampling by skipping intermediate timesteps while maintaining sample coherence, making it particularly suitable for applications requiring rapid inference.

Algorithm 1: *Euler–Maruyama Sampler*

**Require:** Score model $s_\theta$, terminal time $T$, number of steps $N$, initial noise $\mathbf{x}_T$, noise schedule $\sigma(t)$

Initialize $\mathbf{x} \leftarrow \mathbf{x}_T$, timestep $\Delta t \leftarrow T/N$
**for** $i = N, \ldots, 1$ **do**
    $t \leftarrow i \cdot \Delta t$
    $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
    $\mathbf{x} \leftarrow \mathbf{x} - \sigma(t)^2 s_\theta(\mathbf{x}, t) \cdot \Delta t + \sigma(t)\sqrt{\Delta t} \cdot \mathbf{z}$
**end for**
**return** $\mathbf{x}_0$

Algorithm 2: *Predictor–Corrector Sampler*

**Require:** Score model $s_\theta$, number of steps $N$, noise schedule $\sigma(t)$, Langevin step size $\eta$, corrector steps $J$

Initialize $\mathbf{x} \leftarrow \mathbf{x}_T$, timestep $\Delta t \leftarrow T/N$
**for** $i = N, \dots, 1$ **do**
  **Predictor step (Euler–Maruyama):**
  $t \leftarrow i \cdot \Delta t$
  $\mathbf{z}_1 \sim \mathcal{N}(0, \mathbf{I})$
  $\mathbf{x} \leftarrow \mathbf{x} - \sigma(t)^2 s_\theta(\mathbf{x}, t) \cdot \Delta t + \sigma(t)\sqrt{\Delta t} \cdot \mathbf{z}_1$
  **Corrector step (Langevin MCMC):**
  **for** $j = 1$ to $J$ **do** ▷ $J$ typically 1–2
    $\mathbf{z}_2 \sim \mathcal{N}(0, \mathbf{I})$
    $\mathbf{x} \leftarrow \mathbf{x} + \eta s_\theta(\mathbf{x}, t) + \sqrt{2\eta} \cdot \mathbf{z}_2$
  **end for**
**end for**
**return** $\mathbf{x}_0$

Algorithm 3: *Heun Sampler (2nd Order)*

**Require:** Score model $s_\theta$, terminal time $T$, steps $N$, initial noise $\mathbf{x}_T$, noise schedule $\sigma(t)$

Initialize $\mathbf{x} \leftarrow \mathbf{x}_T$, $\Delta t \leftarrow T/N$
**for** $i = N, \dots, 1$ **do**
  $t \leftarrow i \cdot \Delta t$, $t_{\text{prev}} \leftarrow (i-1) \cdot \Delta t$
  $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
  $\mathbf{d}_1 \leftarrow -\sigma(t)^2 s_\theta(\mathbf{x}, t) \cdot \Delta t$
  $\mathbf{x}_{\text{temp}} \leftarrow \mathbf{x} + \mathbf{d}_1 + \sigma(t)\sqrt{\Delta t} \cdot \mathbf{z}$
  $\mathbf{d}_2 \leftarrow -\sigma(t_{\text{prev}})^2 s_\theta(\mathbf{x}_{\text{temp}}, t_{\text{prev}}) \cdot \Delta t$
  $\mathbf{x} \leftarrow \mathbf{x} + \frac{1}{2}(\mathbf{d}_1 + \mathbf{d}_2) + \sigma(t)\sqrt{\Delta t} \cdot \mathbf{z}$
**end for**
**return** $\mathbf{x}_0$

Algorithm 4: *DDIM Sampler*

**Require:** Score model $s_\theta$, steps $N$, timestep sequence $\{\tau_i\}_{i=0}^N$, noise schedule $\{\alpha_t\}$

Initialize $\mathbf{x}_{\tau_N} \sim \mathcal{N}(0, \mathbf{I})$
**for** $i = N, N-1, \dots, 1$ **do**
  $t \leftarrow \tau_i$, $t_{\text{prev}} \leftarrow \tau_{i-1}$
  $\hat{\mathbf{x}}_0 \leftarrow \frac{\mathbf{x}_t - \sqrt{1-\alpha_t^2} \cdot s_\theta(\mathbf{x}_t, t)}{\alpha_t}$
  $\mathbf{dir} \leftarrow \sqrt{1 - \alpha_{t_{\text{prev}}}^2} \cdot s_\theta(\mathbf{x}_t, t)$
  $\mathbf{x}_{t_{\text{prev}}} \leftarrow \alpha_{t_{\text{prev}}} \hat{\mathbf{x}}_0 + \mathbf{dir}$
  $\mathbf{x}_t \leftarrow \mathbf{x}_{t_{\text{prev}}}$
**end for**
**return** $\mathbf{x}_0$

### Computational Complexity and Trade-offs

The computational cost of each sampler varies significantly:

- **Euler-Maruyama**: $O(N)$ score function evaluations, where $N \approx 100 - 1000$
- **Predictor-Corrector**: $O(N \cdot (1 + J))$ evaluations, where $J = 1 - 2$ corrector steps

- **Heun**: $O(2N)$ evaluations but with $N \approx 30 - 50$, resulting in $\sim 60 - 100$ total NFEs
- **DDIM**: $O(N)$ evaluations with $N \approx 10 - 50$, achieving the fastest inference

This flexibility in sampling methods is particularly valuable for multimodal imputation tasks, where different applications may prioritize either generation quality (medical imaging) or inference speed (real-time systems).

### 5.1.6 Alignment Decoder

Once the missing modality sample $\tilde{\mathbf{x}}_m(0)$ is generated from the diffusion model, it often does not match the same feature distribution as the original training data due to imperfect denoising, especially at low NFEs. To bridge this gap, we introduce an alignment decoder $\mathcal{D}_m$, trained to map noisy/generated features to the original feature distribution space.

We adopt a lightweight version of the **Residual Channel Attention Network (RCAN) Figure 5.10** architecture [25], originally designed for image super-resolution. In our 1D adaptation, the RCAN consists of a residual convolutional block (RCAB blocks), scheme in Figure 5.9), with channel attention (CA) seen in Figure 5.8:

- **Residual Convolution:** Extracts local temporal patterns from $\tilde{\mathbf{x}}_m$.
- **Channel Attention:** Re-weights feature channels to enhance discriminative cues using squeeze-and-excitation [51].

This decoder is trained independently with a reconstruction loss also enhanced by a perceptual loss on downstream model activations $\mathcal{L}_{task}$ 5.22:

$$\mathcal{L}_{\text{align}} = \|\mathcal{D}_m(\tilde{\mathbf{x}}_m(0)) - \mathbf{x}_m(0)\|_1, \tag{5.18}$$

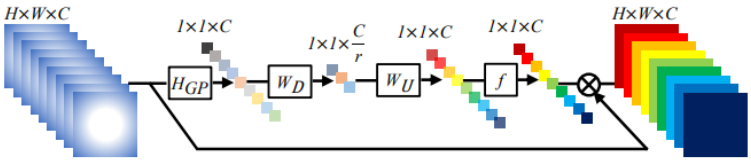


**Figure 5.8.** *Channel attention (CA). Figure from [25].*

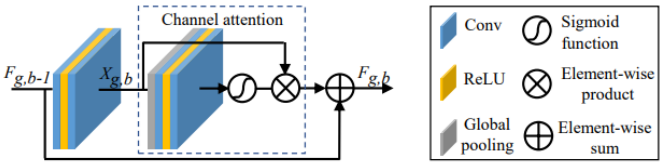


**Figure 5.9.** *Residual channel attention block (RCAB). Figure from [25].*

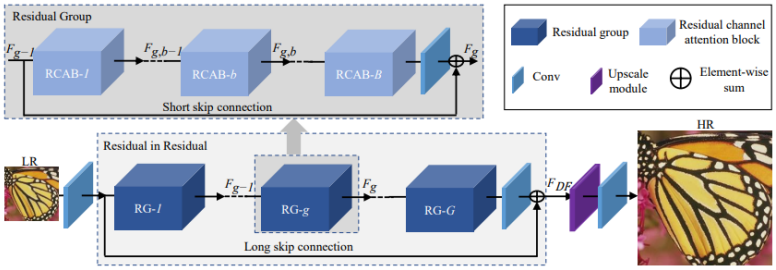


**Figure 5.10.** *Residual channel attention network (RCAN). Figure from [25].*

### 5.1.7 Downstream Fusion Classifiers

To perform emotion recognition from the fused modality features, we adopt a fusion mechanism inspired by the Multimodal Transformer (MulT) architecture [26] as displayed in Figures 5.11,5.12. In particular, we use modality-specific encoders $\mathcal{E}_m$ to project features into a common latent space, and employ a series of pairwise crossmodal transformers to model directional dependencies between modalities. Each transformer learns how to reinforce one modality using information from another through crossmodal attention, effectively capturing long-range interactions across streams of differing lengths and sampling rates. The fused representations are then aggregated via a memory transformer $\mathcal{T}_k$, then the last token for each modality representing all the sequence emotion feature is concatenated with the rest, and passed to a multi-layer perceptron (MLP) for final prediction. The fusion transformers is trained jointly with the classification head using cross-entropy loss for classification or L1 loss for regression[2]. Importantly, during this stage, missing modalities are either masked with zeros or substituted with reconstructions from the diffusion model.

Formally, let $x_m$ denote the input features for modality $m \in \{1, \ldots, M\}$. Each modality is first encoded via a modality-specific shallow encoder $\mathcal{E}_m$:

[2] In our experiments we will only use the regression loss.

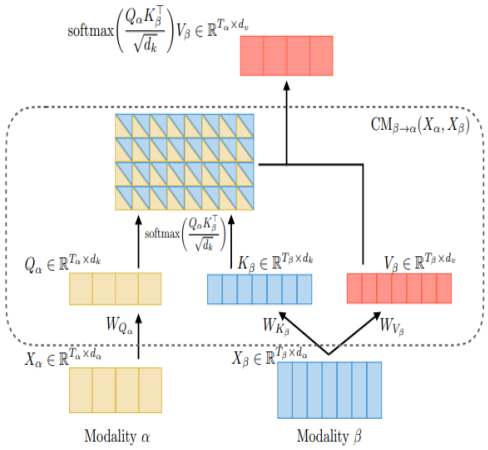


**Figure 5.11.** *Cross-Modal attention mechanism inside a fusion transformer. The modalities we want to enchance serve as queries while the enchancing one serves as keys and values. Figure from [26].*

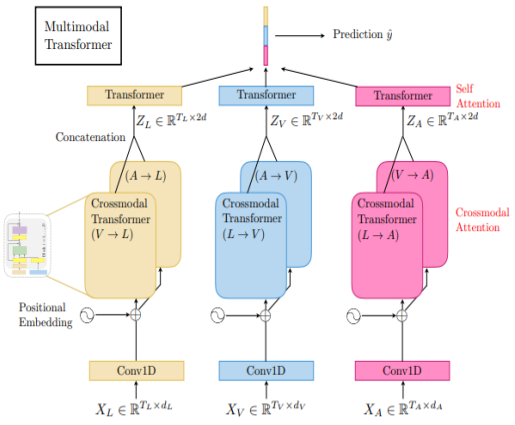


**Figure 5.12.** *MulT CM transformers applied to each pair of language (L), visual (V), and acoustic (A) modalities. Figure from [26].*

$$h_m = \mathcal{E}_m(x_m), \tag{5.19}$$

Next, for each modality $m$, we apply pairwise crossmodal transformers $\mathcal{T}_{m \leftarrow j}$ to incorporate context from all other modalities $j \neq m$. These representations are then aggregated through a memory transformer $\mathcal{T}_m^{\text{mem}}$:

$$\tilde{h}_m = \mathcal{T}_m^{\text{mem}} \left( [\mathcal{T}_{m \leftarrow j}(h_m, h_j)]_{j \neq m} \right), \tag{5.20}$$

The final fused representation is the concatenation of all $\tilde{h}_m$ vectors, passed through a prediction head:

$$\hat{y} = \text{MLP} \left( \left[ \tilde{h}_1, \ldots, \tilde{h}_M \right] \right), \tag{5.21}$$

and the learning objective is given by the L1 loss:

$$\mathcal{L}_{\text{task}} = \|\hat{y} - y\|_1, \tag{5.22}$$

where missing modalities are either replaced with imputed outputs from the decoder or zero-masked during training.

**Total Stage 2 Loss:** In stage 2 training (See Figure 5.2) the total loss is a weighted average of the decoder alignment loss combined with the task loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{align}} \tag{5.23}$$

Where in our experiments we chose $\lambda = 0.2$.

## 5.2 Experimental Setup

### 5.2.1 Multimodal Emotion Recognition Datasets

We evaluate our approach on two widely used multimodal emotion recognition (MER) datasets:

- **CMU-MOSI** [46]: Consists of 2,199 opinionated video clips from YouTube. The dataset is divided into 1,284 training samples, 229 validation samples, and 686 testing samples.
- **CMU-MOSEI** [47]: Contains 22,856 utterance-level video clips annotated with sentiment and emotion labels. The official split includes 16,326 training samples, 1,871 validation samples, and 4,659 testing samples.

### 5.2.2 Feature Extraction

We use modality-specific pre-processing tools as follows:

- **Text**: We use the final hidden state of a pre-trained BERT model [52] to extract 768-dimensional word embeddings.
- **Acoustic**: We extract 74-dimensional acoustic features using the COVAREP toolkit [53], capturing pitch, glottal source parameters, and other prosodic features.
- **Vision**: We extract 35 facial expression features from each frame using the Facet toolkit [54].

### 5.2.3 Evaluation Metrics

We follow prior work [1] and report three standard metrics:

- **Binary Accuracy ($\mathbf{ACC_2}$)**: Binary classification of sentiment (positive/negative).

- **7-Class Accuracy ($ACC_7$)**: Fine-grained classification over 7 ordinal sentiment categories.
- **F1 Score**: Macro-averaged F1 score over the binary classification.

These metrics provide a balanced view of performance across both coarse and fine sentiment granularity.

### 5.2.4 Random and Fixed Missing Protocols

In the context of handling incomplete multimodal data, two primary missing protocols are employed:

- **Fixed Missing Protocol**: Under this protocol, a consistent set of one or two modalities is deliberately discarded across all samples in the dataset. For example, experiments might be conducted where:
  - One modality is missing (e.g., only language, only vision, or only acoustic data is available).
  - Two modalities are missing (e.g., only language and vision, language and acoustic, or vision and acoustic data are available).

  This ensures a predefined missing pattern throughout the experimental setup.
- **Random Missing Protocol**: This protocol introduces variability by randomizing the missing patterns for each individual sample. Consequently, for any given sample, either one or two modalities might be absent. The degree of missingness in this protocol is quantified by the Missing Rate (MR), defined as:

$$MR = 1 - \frac{\sum_{i=1}^{N} m_i}{N \times M}$$

  where $m_i$ represents the number of available modalities for the $i^{th}$ sample, $N$ is the total number of samples, and $M$ is the total number of modalities (in this case, $M = 3$). It is a crucial constraint that at least one modality must always be available for each sample ($m_i \geq 1$), which implies that the maximum possible Missing Rate is $\frac{M-1}{M}$. For experiments with three modalities, MR values were selected from the set $\{0.0, 0.3, 0.5, 0.7\}$. The chosen MR is consistently maintained across the training, validation, and testing phases to ensure fair evaluation.

### 5.2.5 Implementation Details

For all experiments we used the Adam optimizer [55] with Learning rate $\lambda = 0.002$ amd weight decay $\beta = 0.005$. Furthermore, early stopping with *patience* $= 10$ and model checkpoints was used for the results of stage 2 training. All extracted modality features are projected into a common feature space through a shallow encoder[3] with channel dimention $d = 32$ and sequence length $T = 48$. For the diffusion part we opted to design all score networks to be of the same trainable parameter size for a fair comparison:

- **UNet with Cross Attention**: U-Net architecture with cross-attention conditioning, featuring 4-level encoder-decoder with channels $[32, 64, 128, 256]$, time embedding dimension $d_{emb} = 256$ with layerwise dense layers for fusion, and Transformer Encoder Cross Attention blocks with 2 layers and 8 attention heads per level.
- **Multimodal Diffusion Transformer**: Transformer architecture with element-wise Linear Modulation (FiLM) conditioning, model dimension $d_{model} = 256$, depth of 6 transformer layers, 8 attention heads, MLP dimension $d_{mlp} = 512$, and conditioning dimension $d_{cond} = 256$ and time embedding dimension $d_{time} = 128$.
- **Diffusion Transformer**: Diffusion Transformer with model dimension $d_{model} = 256$, depth of 6 transformer layers, 8 attention heads, MLP dimension $d_{mlp} = 512$, and conditioning dimension $d_{cond} = 256$ which also contains the timestep information.
- **ScoreTransformer1D**: 1D Score Transformer with model dimension $d_{model} = 256$, depth of 6 layers, 8 attention heads, MLP dimension $d_{mlp} = 512$, and time embedding dimension $d_{time} = 256$.

To add to that, the score networks were trained for 50 epochs maximum[4]. Next, the alignment decoder $D_m$ used to further refine the diffused modalities has 20 RCAB blocks with reduction 16. All downstream fusion classifiers $T_k$ are default Pytorch Transformer Encoders with 4 layers, 8 heads and *attention dropout* $= 0.2$.

[3] A single layer 1D convolutional kernel with size 3.
[4] if early stopping didn't occur

# Chapter 6

# Experimental Results

We adopt a sequential optimization strategy to identify the best configuration. We systematically evaluate: (1) SDE formulations (VP vs VE) with the default cross attention U-Net and Euler-Maruyama Sampler, (2) backbone architectures again with the same default baseline sampler, and (3) sampling methods, selecting the best performing option at each stage before proceeding to the next evaluation. At each stage we will be comparing the performance of the model with the **baseline IMDER** model that our work is based on.

## 6.1 Stage 1: SDE Formulation Comparison

We first compare the two SDE formulations using a baseline configuration to determine which provides better performance for multimodal emotion recognition tasks.

### Starting Configuration

For this comparison, we use:

- **Baseline Architecture**: U-Net with Cross-Attention conditioning
- **Sampler**: Euler-Maruyama with 100 NFEs
- **Missing Protocols**: Both Fixed and Random missing patterns
- **Datasets**: CMU-MOSI

### Analysis of SDE Formulations on CMU-MOSI

Table 6.1 presents a comparison between Variance Exploding (VE) and Variance Preserving (VP) SDE formulations across various missing modality scenarios on the

**Table 6.1.** *Comparison of SDE Formulations and Vanilla IMDER on CMU-MOSI Dataset for the fixed missing protocol. Red coloring is used to indicate the previous comparing method results (IMDER [1]).*

| Available Modalities | Variance Exploding (VE) | | | Variance Preserving (VP) | | | Vanilla IMDER (VE) | | |
|---|---|---|---|---|---|---|---|---|---|
| | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ |
| Language | 84.9 | 84.9 | 45.3 | **85.3** | **85.3** | **45.6** | 84.8 | 84.7 | 44.8 |
| Acoustic | 60.4 | 60.3 | 18.7 | **62.0** | **62.1** | 17.7 | 61.3 | 60.8 | **20.5** |
| Vision | 58.1 | 58.3 | 19.1 | 58.6 | 58.8 | 18.3 | **61.0** | **61.2** | **21.0** |
| Language + Acoustic | 85.6 | 85.4 | **46.7** | **86.4** | **86.3** | 45.4 | 85.4 | 85.3 | 45.0 |
| Language + Vision | 85.5 | 85.4 | **45.6** | **86.1** | **85.9** | 45.0 | 85.5 | 85.4 | 45.3 |
| Acoustic + Vision | 60.6 | 59.3 | 19.5 | 59.6 | 59.7 | **21.8** | **62.0** | **62.1** | 20.2 |
| **Average** | 72.5 | 72.2 | 32.4 | 73.0 | 73.0 | 32.3 | **73.3** | **73.2** | **32.8** |

**Table 6.2.** *SDE Formulations and Vanilla IMDER approach under Random Missing Protocol (CMU-MOSI). For each experiment listed in the table below, we ran the model with 5 different random seeds on the test set and averaged the results for more robust metrics.*

| Missing Rate | Variance Exploding (VE) | | | Variance Preserving (VP) | | | Vanilla IMDER (VE) | | |
|---|---|---|---|---|---|---|---|---|---|
| | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ |
| MR = 0.3 | 80.6 | 80.7 | **41.8** | **81.0** | **81.0** | 41.5 | 79.9 | 79.6 | 39.1 |
| MR = 0.5 | 73.6 | 72.5 | 33.7 | **74.6** | 73.2 | 33.5 | 74.0 | **73.8** | **34.2** |
| MR = 0.7 | 69.5 | 69.7 | 29.5 | 70.2 | 69.9 | 30.2 | **70.8** | **70.3** | **31.6** |
| **Average** | 74.5 | 74.3 | **35.0** | **75.2** | **74.7** | **35.0** | 74.9 | 74.6 | 34.9 |

CMU-MOSI dataset. The metrics reported include binary accuracy ($ACC_2$), F1 score, and 7-class accuracy ($ACC_7$).

**VP Generally Outperforms VE.** Across most missing modality patterns, the VP formulation consistently yields higher $ACC_2$ and F1 scores compared to VE. For instance, in the *Language + Acoustic* setting, VP achieves 86.4 $ACC_2$ and 86.3 F1, outperforming VE's 85.6 and 85.4, respectively. A similar trend holds in the *Language + Vision* configuration.

**Performance when Acoustic and Vision are available.** Both formulations show significantly lower performance when only the acoustic or vision modality is available. This reflects the dominance of the language modality in sentiment prediction tasks on CMU-MOSI, as seen in prior work. Also its notable that when both acoustic and vision the model doesnt perform much better and in the VP case it performs slightly worse that just having the Acoustic modality in $ACC_2$ and F1.

**$ACC_7$ Results Are Mixed.** The 7-class accuracy ($ACC_7$) shows more variability. VE slightly outperforms VP in certain settings (e.g., *Language + Acoustic* and *Vision-only*), while VP yields better scores in others (e.g., *Acoustic + Vision*). These results indicate that VE may occasionally preserve finer-grained prediction capabilities, although the differences are minor.

**Multimodal Fusion Leads to Strongest Performance.** The best overall results are observed when language is fused with another modality, demonstrating the benefit of multimodal learning. In these cases, VP remains the more reliable formulation.

**Averaged Performance.** On average, VP achieves slightly better binary accuracy and F1 score (73.0 for both) compared to VE (72.5 $ACC_2$, 72.2 F1), while VE slightly surpasses VP on $ACC_7$ (32.4 vs. 32.3). The differences, however, are **marginal**.

Summary

Overall, VP offers more consistent and robust performance across missing modality scenarios, particularly in binary and F1 metrics. While VE may retain slight advantages in fine-grained classification in select cases, VP is generally more effective for multimodal sentiment analysis on CMU-MOSI.

## 6.2 Stage 2: Conditioning Architecture Comparison

Using the optimal SDE formulation from Stage 1, we compare different backbone architectures and their conditioning mechanisms.

### Experimental Configuration

For this comparison, we use:

- **Selected SDE**: Variance Preserving SDE (VP SDE)
- **Sampler**: Euler-Maruyama with 100 NFEs
- **Architectures**: U-Net with Cross-Attention, Multimodal Diffusion Transformer with Concatenation and FiLM like layers, Diffusion Transformer with AdaLN and ScoreTransformer1D with simple concatenation
- **Missing Protocols**: Fixed missing patterns for comprehensive evaluation

### Conditioning comparison Results

From Table 6.3, we observe that performance differences across backbones are relatively modest, but consistent trends emerge. For unimodal cases, the Di-Transformer slightly outperforms on language-only tasks, while U-Net remains strong for acoustic inputs. Interestingly, MMDi-Transformer achieves the highest $ACC_7$ when recovering acoustic or vision modalities, suggesting that its FiLM-style conditioning provides advantages for fine-grained classification. ScoreTransformer1D, however, matches or surpasses competitors in several settings and obtains the best average $ACC_2$ and F1 across missing patterns.

Table 6.4 highlights a contrast in efficiency. Despite having the fewest parameters and the second lowest FLOPs number, the U-Net backbone is over twice as slow at inference compared to transformer-based alternatives. In contrast, our proposed ScoreTransformer1D is not only the most parameter-efficient model (3.2M parameters) but also achieves the fastest inference time (13.1 ms) with a relatively low FLOP number, over 5× faster than U-Net, while maintaining competitive accuracy. This makes it especially attractive for deployment in time-sensitive multimodal applications.

**Summary**

In summary, ScoreTransformer1D achieves the best trade-off between performance and efficiency. While U-Net and Di-Transformer variants show slightly stronger results in isolated cases, ScoreTransformer1D consistently matches their accuracy while drastically outperforming them in computational cost, establishing it as the most practical backbone for missing modality recovery, thus we will adopt it for the rest of our experiments.

**Table 6.3.** *Conditioning Architecture Comparison on CMU-MOSI Dataset. Red coloring is used to indicate the previous comparing method results (IMDER [1]).*

| Available Modalities | U-Net Cross-Attn | | | Di-Transformer | | | MMDi-Transformer | | | ScoreTransformer1D | | | Vanilla IMDER (Unet) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ | $ACC_2$ | F1 | $ACC_7$ |
| Language | 85.3 | 85.3 | 45.6 | **86.1** | **86.0** | 45.4 | 84.4 | 84.4 | **46.6** | 85.6 | 85.5 | 45.3 | 84.8 | 84.7 | 44.8 |
| Acoustic | **62.0** | **62.1** | 17.7 | 61.2 | 61.1 | 18.8 | 61.8 | 60.9 | **20.9** | 61.0 | 60.0 | 20.0 | 61.3 | 60.8 | 20.5 |
| Vision | 58.6 | 58.8 | 18.3 | 59.6 | 58.5 | 17.4 | 60.2 | 59.8 | **18.6** | **61.1** | 60.8 | 17.6 | 61.0 | **61.2** | 21.0 |
| Language + Acoustic | **86.4** | **86.3** | 45.4 | 86.0 | 85.9 | **46.2** | 85.6 | 85.5 | 45.0 | 85.5 | 85.4 | 45.0 | 85.4 | 85.3 | 45.0 |
| Language + Vision | 86.1 | 85.9 | 45.0 | 86.0 | 85.9 | **47.8** | 85.3 | 85.3 | 46.0 | **86.4** | **86.3** | 46.3 | 85.5 | 85.4 | 45.3 |
| Acoustic + Vision | 59.6 | 59.7 | **21.8** | 61.0 | 60.4 | 19.7 | 61.2 | 61.3 | 19.5 | 61.3 | 60.4 | 19.5 | **62.0** | **62.1** | 20.2 |
| **Average** | 73.0 | 73.0 | 32.3 | 73.3 | 72.9 | 32.6 | 73.1 | 72.9 | **32.8** | **73.5** | 73.1 | 32.3 | 73.3 | **73.2** | **32.8** |

**Table 6.4.** *Model Efficiency Analysis: This table presents the Sizes, inference time, and computational complexity (FLOPs) for a single pass through a modality score network. Notable even though the Unet has very few parameters compared to its Transformer competitors its inference time is over double of the worse transformer based one.*

| Architecture | Parameters (M) | Training Time (hrs) | Inference Time (ms) | FLOPs (G) | Memory (MB) |
|---|---|---|---|---|---|
| U-Net Cross-Attention | 3.5 | 0.55 | 72.1 | 0.66 | 13.1 |
| Multimodal Di-Transformer | 8.8 | 0.52 | 31.2 | 1.21 | 33.7 |
| Di-Transformer | 9.3 | 0.45 | 24.6 | **0.53** | 35 |
| ScoreTransformer1D | **3.2** | **0.30** | **13.1** | 1.04 | **12.5** |

**Table 6.5.** *Results under the **Fixed Missing Protocol** for Euler and PC samplers for different step numbers (the number of the steps is indicated next to the samplers name).*

| Dataset | Available Modalities | Euler 80 | Euler 50 | PC 50 | PC 40 | PC 30 |
|---|---|---|---|---|---|---|
| **MOSI** | {l} | 86.1 / 86.0 / 47.6 | 85.9 / 85.7 / 45.3 | 86.0 / 85.9 / 45.1 | 86.0 / 86.0 / 47.4 | 84.7 / 84.7 / 46.3 |
| | {a} | 61.9 / 59.6 / 21.4 | 60.3 / 59.4 / 21.4 | 61.7 / 60.3 / 22.0 | 61.1 / 60.7 / 20.0 | 60.6 / 61.1 / 20.9 |
| | {v} | 58.1 / 58.3 / 17.2 | 60.0 / 60.2 / 17.2 | 59.7 / 59.7 / 20.8 | 59.0 / 59.1 / 19.1 | 59.6 / 60.1 / 19.8 |
| | {l,a} | 84.7 / 84.7 / 47.1 | 85.2 / 85.1 / 46.2 | 85.6 / 85.6 / 45.0 | 85.2 / 85.1 / 46.2 | 85.6 / 85.6 / 46.5 |
| | {l,v} | 84.7 / 84.7 / 45.9 | 85.2 / 85.1 / 47.1 | 85.0 / 85.0 / 46.8 | 85.0 / 85.0 / 45.6 | 85.6 / 85.5 / 46.6 |
| | {v,a} | 62.0 / 62.2 / 18.1 | 63.2 / 63.0 / 23.4 | 59.4 / 59.7 / 19.2 | 61.1 / 61.1 / 19.2 | 60.8 / 60.9 / 20.1 |
| **Average** | – | 72.9 / 72.6 / 32.9 | 73.3 / 73.1 / 33.3 | 72.9 / 72.7 / 33.2 | 72.9 / 72.8 / 32.9 | 72.8 / 72.9 / 33.4 |

## 6.3 Stage 3: Sampling Algorithm Comparison

Using the optimal SDE-backbone combination from previous stages, we evaluate different sampling algorithms to find the best speed-quality trade-off.

### Experimental Configuration

For this comparison, we use:

- **Selected Configuration**: VP SDE + Score Transformer Backbone
- **Samplers**: Euler-Maruyama, Predictor-Corrector, Heun, DDIM
- **NFE Range**: 10-100 function evaluations
- **Evaluation**: Performance vs. speed trade-offs

### Speed-Quality Trade-off Analysis

Based on the results in Table 6.7 which are abriviated from Table **??**, we can analyze the performance-efficiency trade-offs across different sampling configurations.

**Table 6.6.** *Results under the **Fixed Missing Protocol** for Heun and DDIM samplers for different step numbers (the number of the steps is indicated next to the samplers name).*

| Dataset | Available Modalities | Heun 80 | Heun 60 | Heun 40 | DDIM 30 | DDIM 20 | DDIM 10 |
|---|---|---|---|---|---|---|---|
| MOSI | {l} | 86.1 / 86.0 / 46.9 | 86.1 / 86.0 / 45.9 | 85.5 / 85.4 / 45.0 | 85.8 / 85.7 / 46.5 | 86.1 / 86.0 / 45.3 | 85.8 / 85.7 / 47.2 |
| | {a} | 62.8 / 62.2 / 20.7 | 61.5 / 60.9 / 21.0 | 60.9 / 60.5 / 20.5 | 62.3 / 59.7 / 21.8 | 60.2 / 60.4 / 15.9 | 60.6 / 60.5 / 20.0 |
| | {v} | 60.0 / 58.8 / 19.6 | 61.2 / 61.1 / 22.3 | 60.5 / 59.8 / 18.0 | 61.4 / 59.6 / 18.9 | 58.6 / 58.7 / 18.2 | 59.1 / 59.1 / 18.5 |
| | {l,a} | 86.1 / 86.0 / 45.3 | 85.8 / 85.6 / 45.6 | 85.0 / 85.0 / 45.2 | 86.1 / 86.0 / 45.0 | 85.2 / 85.1 / 46.9 | 85.8 / 85.7 / 45.9 |
| | {l,v} | 85.2 / 85.1 / 47.3 | 85.0 / 85.0 / 46.0 | 84.5 / 84.4 / 45.0 | 85.0 / 85.0 / 45.6 | 84.0 / 84.0 / 44.9 | 86.1 / 86.0 / 45.3 |
| | {v,a} | 60.5 / 60.7 / 18.3 | 61.0 / 61.2 / 19.5 | 60.5 / 60.0 / 19.0 | 62.0 / 59.2 / 22.1 | 60.0 / 60.1 / 21.1 | 60.2 / 60.4 / 21.1 |
| **Average** | – | 73.4 / 73.1 / 32.9 | 73.4 / 73.3 / 33.4 | 72.8 / 72.5 / 32.9 | 73.8 / 72.5 / 33.3 | 72.4 / 72.4 / 32.1 | 72.9 / 72.8 / 33.0 |

**Table 6.7.** *Sampling Algorithm Comparison on CMU-MOSI Dataset derived from tables 6.5,6.6 , performance is the average of the fixed missing protocol for each sampler configuration. Red coloring is used to indicate the previous comparing method results (IMDER [1]).*

| Sampler | NFEs | $ACC_2$ | F1 | $ACC_7$ | Sampling Time (s) |
|---|---|---|---|---|---|
| Vanilla IMDER | 100 | 73.3 | 73.2 | 32.8 | 1.17 |
| Euler-Maruyama | 100 | 73.5 | 73.1 | 32.3 | 1.17 |
| Euler-Maruyama | 80 | 72.9 | 72.6 | 32.9 | 0.95 |
| Euler-Maruyama | 50 | 73.3 | 73.1 | 33.3 | 0.61 |
| Predictor-Corrector | 100 | 72.9 | 72.7 | 33.2 | 1.17 |
| Predictor-Corrector | 80 | 72.9 | 72.8 | 32.9 | 0.95 |
| Predictor-Corrector | 60 | 72.8 | 72.9 | **33.4** | 0.71 |
| Heun | 80 | 73.4 | 73.1 | 32.9 | 0.95 |
| Heun | 60 | 73.4 | **73.3** | **33.4** | 0.71 |
| Heun | 40 | 72.8 | 72.5 | 32.9 | 0.49 |
| DDIM | 30 | **73.8** | 72.5 | 33.3 | 0.37 |
| DDIM | 20 | 72.4 | 72.4 | 32.1 | 0.24 |
| DDIM | 10 | 72.9 | 72.8 | 33.0 | **0.12** |

**Performance Analysis.** The Euler-Maruyama sampler with 100 NFEs odly achieves the second highest $ACC_2$ (73.5) and F1 (73.1) scores behind the DDIM sampler with 30 NFEs for the former and the Heun sampler with 60 NFEs for the latter. **DDIM** with 30 NFEs delivers competitive performance ($ACC_2$: 73.8, F1: 72.5) while being significantly faster (0.37s vs 1.17s). The **Heun** sampler with 60 NFEs provides excellent performance ($ACC_2$: 73.4, F1: 73.3, $ACC_7$: 33.4) with moderate speed (0.71s).

**Speed Analysis.** **DDIM** demonstrates superior speed-quality trade-offs, achieving near-baseline performance with 3× faster inference. **Predictor-Corrector** methods show minimal performance gains over simpler approaches while maintaining higher computational costs. Heun sampler with 60 NFEs maintains strong performance ($ACC_2$: 73.4) with nearly 1.66× speed improvement.

**Table 6.8.** *Speed-Quality Trade-off Analysis, here we compare the most representative sampling configurations.*

| Sampler Configuration | Relative Speed | $ACC_2$ Change | Sampling Time (s) | Recommended Use |
|---|---|---|---|---|
| Euler-Maruyama (100 NFEs) | 1.0× | 73.5 (baseline) | 1.17 | High-quality baseline |
| Euler-Maruyama (50 NFEs) | 1.9× | 73.3 (-0.3%) | 0.61 | Balanced quality-speed |
| Heun (60 NFEs) | 1.6× | **73.4 (-0.1%)** | 0.71 | **Fast with quality retention** |
| DDIM (30 NFEs) | 3.2× | **73.8 (+0.4%)** | 0.37 | **Optimal speed choice** |
| DDIM (10 NFEs) | 9.8× | 72.9 (-0.8%) | **0.12** | Ultra-fast deployment |

### Final Configuration Recommendations

Based on our comprehensive evaluation across all three stages, we recommend two optimal configurations:

**Quality-Focused Configuration.**

- **SDE**: Variance Preserving (VP)
- **Backbone**: ScoreTransformer1D (3.2M parameters)
- **Sampler**: Heun sampler with 60 NFEs
- **Performance**: 73.4 $ACC_2$, 73.3 F1, 33.4 $ACC_7$
- **Speed**: 0.71s sampling time
- **Use Case**: Applications requiring maximum accuracy with acceptable inference time

**Speed-Optimized Configuration.**

- **SDE**: Variance Preserving (VP)
- **Backbone**: ScoreTransformer1D (3.2M parameters)
- **Sampler**: DDIM with 30 NFEs
- **Performance**: 73.8 $ACC_2$, 72.5 F1, 33.3 $ACC_7$
- **Speed**: 0.37s sampling time (3.2× faster than baseline)
- **Use Case**: Real-time applications and deployment scenarios where speed is critical

Key Observations

**DDIM Superiority.** DDIM with 30 NFEs emerges as the optimal choice for fast downstream inference, actually achieving slightly better $ACC_2$ performance than the baseline while being significantly faster. This counter-intuitive result suggests that the deterministic nature of DDIM sampling may provide better convergence properties for our multimodal diffusion framework.

**Diminishing Returns.** Beyond 60 NFEs, performance improvements are marginal while computational costs increase substantially. This observation aligns with recent findings in diffusion model literature that suggest fewer sampling steps can be sufficient for many practical applications.

## 6.4 Comparison with State-of-the-Art Methods

Using our two optimal configurations identified in Stage 3, we compare against existing multimodal imputation methods including the original Vanilla IMDER baseline.

**Results Analysis.** The experimental results demonstrate the effectiveness of our optimized configurations across both random and fixed missing protocols. Our approach consistently matches or outperforms existing state-of-the-art methods, with the Quality-Optimized configuration achieving the highest performance in most scenarios while maintaining superior computational efficiency. Our Speed-Optimized configuration also performs competitively given its lower computational overhead.

**Performance Under Fixed Missing Protocol.** The fixed missing protocol results further reveal the robustness of our approach across different modality availability scenarios. Our **Quality-Optimized** configuration achieves similar average performance with the speed advantages of our sampler and architectural choices on both datasets. More specifically on **CMU-MOSEI** (76.1% $ACC_2$, 75.6% F1, 48.4% $ACC_7$), outperforming Vanilla IMDER by 1.3% $ACC_2$ and GCNET by 1.8% $ACC_2$ in and 2.5% in F1 and marginal improvement on the rest of the metrics and **On CMU-MOSI**, similar improvements are observed, with our **Quality-Optimized** configuration achieving competitive results. Notably, our method shows consistent results with the baseline approach across challenging scenarios such as vision-only and acoustic-only settings, where traditional methods often struggle.

**Table 6.9.** *Performance comparison under both **Random Missing Protocol** and **Fixed Missing Protocol** on CMU-MOSI and CMU-MOSEI datasets. Each cell reports $ACC_2$ / F1 / $ACC_7$. Baseline results for DCCA [27], DCCAE [28], MCTN [29], MMIN [6], and GCNet [30] are taken from prior work [1]. Our **Quality-Optimized** configuration uses VP SDE + ScoreTransformer1D + Heun (60 NFEs), while **Speed-Optimized** uses VP SDE + ScoreTransformer1D + DDIM (30 NFEs). **Bolded values** indicate the best score per metric.*

**(a)** *Results under the **Random Missing Protocol** at various missing rates (MR). We report the average over 5 random seeds for each missing rate case for a more robust result.*

| Dataset | MR | DCCA | DCCAE | MCTN | MMIN | GCNet | Vanilla IMDER | Quality-Opt | Speed-Opt |
|---|---|---|---|---|---|---|---|---|---|
| **MOSI** | 0.0 | 75.3 / 75.4 / 30.5 | 77.3 / 77.4 / 31.2 | 81.4 / 81.5 / 43.4 | 84.6 / 84.4 / 44.8 | 85.2 / 85.1 / 44.9 | 85.7 / 85.6 / 45.3 | **86.0** / **86.2** / 45.6 | **86.0** / **86.2** / 45.6 |
| | 0.3 | 68.4 / 67.8 / 25.1 | 70.2 / 69.5 / 25.8 | 73.9 / 74.0 / 33.4 | 76.3 / 75.7 / 34.5 | 77.4 / 76.9 / 35.7 | 79.9 / 79.6 / 39.1 | **81.0** / **80.8** / **40.7** | 80.3 / 79.8 / 39.9 |
| | 0.5 | 61.7 / 60.9 / 21.0 | 63.4 / 62.1 / 21.9 | 68.3 / 67.9 / 29.6 | 71.2 / 70.3 / 30.9 | 72.6 / 72.0 / 31.5 | 74.0 / 73.8 / **34.2** | **75.3** / **74.3** / 34.1 | 73.8 / 73.8 / 33.6 |
| | 0.7 | 55.2 / 53.8 / 18.2 | 56.7 / 55.0 / 19.3 | 61.5 / 61.2 / 26.1 | 64.1 / 62.4 / 27.3 | 65.8 / 64.9 / 28.0 | 70.8 / 70.3 / 31.6 | 70.4 / **70.5** / **32.1** | **71.0** / 70.5 / 31.8 |
| **Average** | - | 65.2 / 64.5 / 23.7 | 66.9 / 66.0 / 24.6 | 71.3 / 71.2 / 33.1 | 74.1 / 73.2 / 34.4 | 75.3 / 74.7 / 35.0 | 77.6 / 77.3 / 37.6 | **78.2** / **78.0** / **38.1** | 77.8 / 77.6 / 37.7 |
| **MOSEI** | 0.0 | 80.7 / 80.9 / 47.7 | 81.2 / 81.2 / 48.2 | 84.2 / 84.2 / 51.2 | 84.3 / 84.2 / 52.4 | 85.2 / 85.1 / 51.5 | 85.1 / 85.1 / 53.4 | **85.7** / **85.8** / 53.3 | **85.7** / **85.8** / 53.3 |
| | 0.3 | 75.1 / 74.2 / 44.0 | 76.3 / 75.6 / 44.3 | 78.6 / 78.3 / 47.1 | 79.7 / 79.2 / 48.3 | 80.6 / 80.0 / 48.5 | 80.2 / 79.7 / 50.1 | **80.8** / **80.4** / **50.3** | 79.9 / 80.0 / 50.1 |
| | 0.5 | 70.8 / 69.1 / 41.1 | 71.9 / 70.3 / 41.3 | 75.3 / 74.9 / 45.5 | 76.4 / 75.0 / 46.7 | 77.3 / 76.2 / 46.8 | 78.2 / 77.3 / 47.9 | **79.0** / **78.1** / **48.9** | 77.5 / 77.2 / 46.9 |
| | 0.7 | 66.3 / 64.2 / 38.0 | 67.2 / 65.0 / 38.7 | 70.4 / 70.1 / 43.2 | 71.5 / 70.6 / 44.5 | 72.7 / 71.7 / 44.9 | **74.2** / 73.2 / 46.0 | 73.2 / **73.6** / **46.4** | 73.2 / 73.1 / 45.3 |
| **Average** | - | 73.2 / 72.1 / 42.7 | 74.2 / 73.0 / 43.1 | 77.1 / 76.9 / 46.8 | 78.0 / 77.3 / 48.0 | 79.0 / 78.3 / 47.9 | 79.4 / 78.8 / 49.4 | **79.7** / **79.5** / **49.7** | 79.1 / 79.0 / 48.9 |

**(b)** *Results under the **Fixed Missing Protocol** for different modality subsets.*

| Dataset | Available Modalities | DCCA | DCCAE | MCTN | MMIN | GCNet | Vanilla IMDER | Quality-Opt | Speed-Opt |
|---|---|---|---|---|---|---|---|---|---|
| **MOSI** | {l} | 73.6 / 73.8 / 30.2 | 76.4 / 76.5 / 28.3 | 79.1 / 79.2 / 41.0 | 83.8 / 83.8 / 41.6 | 83.7 / 83.6 / 42.3 | 84.8 / 84.7 / 44.8 | **86.1** / **86.0** / 45.9 | 85.8 / 85.7 / **46.5** |
| | {v} | 47.7 / 41.5 / 16.6 | 52.6 / 51.1 / 17.1 | 55.0 / 54.4 / 16.3 | 57.0 / 54.0 / 15.5 | 56.1 / 55.7 / 16.9 | 61.3 / 60.8 / 20.5 | **61.2** / **61.1** / **22.3** | 61.4 / 59.6 / 18.9 |
| | {a} | 50.5 / 46.1 / 16.3 | 48.8 / 42.1 / 16.9 | 56.1 / 54.5 / 16.5 | 55.3 / 51.5 / 15.5 | 56.1 / 54.5 / 16.6 | 61.0 / **61.2** / 21.0 | 61.5 / 60.9 / 21.0 | **62.3** / 59.7 / **21.8** |
| | {l, v} | 74.9 / 75.0 / 30.3 | 76.7 / 76.8 / 30.0 | 81.1 / 81.2 / 42.1 | 83.8 / 83.9 / 42.0 | 84.3 / 84.2 / 43.4 | **85.5** / **85.4** / 45.3 | 85.0 / 85.0 / **46.0** | 85.0 / 85.0 / 45.6 |
| | {l, a} | 74.7 / 74.8 / 29.7 | 77.0 / 77.0 / 30.2 | 81.0 / 81.0 / 43.2 | 84.0 / 84.0 / 42.3 | 84.5 / 84.4 / 43.4 | 85.4 / 85.3 / 45.0 | 85.8 / 85.6 / **45.6** | **86.1** / **86.0** / 45.0 |
| | {v, a} | 50.8 / 46.4 / 16.6 | 54.0 / 52.5 / 17.4 | 57.5 / 57.4 / 16.8 | 60.4 / 58.5 / 19.5 | 62.0 / 61.9 / 17.2 | **62.0** / **62.1** / 20.2 | 61.0 / 61.2 / 19.5 | **62.0** / 59.2 / **22.1** |
| | {l, v, a} | 75.3 / 75.4 / 30.5 | 77.3 / 77.4 / 31.2 | 81.4 / 81.5 / 43.4 | 84.6 / 84.4 / 44.8 | 85.2 / 85.1 / 44.9 | 85.7 / 85.6 / 45.3 | **86.0** / **86.2** / **45.6** | **86.0** / **86.2** / **45.6** |
| **Average** | – | 63.9 / 61.9 / 20.0 | 66.1 / 64.8 / 24.4 | 70.2 / 69.9 / 31.3 | 72.7 / 71.4 / 31.6 | 73.1 / 72.8 / 32.1 | 75.1 / 75.0 / 34.5 | 75.2 / **75.1** / **35.1** | **75.5** / 74.5 / **35.1** |
| **MOSEI** | {l} | 78.5 / 78.7 / 46.7 | 79.7 / 79.5 / 47.0 | 82.6 / 82.8 / 50.2 | 82.3 / 82.4 / 51.4 | 83.0 / 83.2 / 51.2 | 84.3 / 84.2 / 52.7 | **85.6** / **85.5** / **53.1** | 83.5 / 83.7 / 51.9 |
| | {v} | 61.9 / 55.7 / 41.3 | 61.1 / 57.2 / 40.1 | 62.6 / 57.1 / 41.6 | 59.3 / 60.0 / 40.7 | 61.9 / 61.6 / 41.7 | 61.5 / **62.6** / 41.6 | **63.6** / **62.6** / **42.3** | 61.3 / 61.4 / 41.3 |
| | {a} | 62.0 / 50.2 / 41.1 | 61.4 / 53.8 / 40.9 | 62.7 / 54.5 / 41.4 | 58.9 / 59.5 / 40.4 | 60.2 / 60.3 / 41.1 | 61.6 / **61.5** / 41.3 | **63.3** / 60.6 / **41.4** | 62.3 / 61.3 / 40.5 |
| | {l, v} | 80.3 / 79.7 / 46.6 | 80.4 / 80.4 / 47.1 | 83.2 / 83.2 / 50.4 | 83.8 / 83.4 / 51.2 | 84.3 / 84.4 / 51.1 | 84.5 / 85.1 / 52.8 | 85.0 / 85.0 / **53.1** | **85.2** / **85.3** / 52.4 |
| | {l, a} | 79.5 / 79.2 / 46.7 | 80.0 / 80.0 / 47.4 | 83.5 / 83.3 / 50.7 | 83.7 / 83.3 / 52.0 | 84.3 / 84.4 / 51.3 | 85.1 / 85.1 / **53.1** | **85.5** / **85.5** / 52.9 | 84.4 / 83.8 / 52.3 |
| | {v, a} | 63.4 / 56.9 / 41.5 | 62.7 / 59.2 / 41.6 | 63.7 / 62.7 / 42.1 | 63.5 / 61.9 / 41.8 | 64.1 / 57.2 / 42.0 | 63.5 / 63.3 / **42.8** | **63.9** / **64.0** / **42.8** | 63.7 / 62.9 / 42.3 |
| | {l, v, a} | 80.7 / 80.9 / 47.7 | 81.2 / 81.2 / 48.2 | 84.2 / 84.2 / 51.2 | 84.3 / 84.2 / 52.4 | 85.2 / 85.1 / 51.5 | 85.1 / 85.1 / **53.4** | **85.7** / **85.8** / 53.3 | **85.7** / **85.8** / 53.3 |
| **Average** | – | 72.3 / 68.8 / 44.5 | 72.4 / 70.2 / 44.6 | 74.6 / 72.5 / 46.8 | 73.7 / 73.5 / 47.1 | 74.7 / 73.7 / 47.1 | 75.1 / 75.3 / 48.2 | **76.1** / **75.6** / **48.4** | 75.2 / 74.9 / 47.7 |

**Performance Under Random Missing Protocol.** On average across missing rates, our proposed configurations matches performance both Vanilla IMDER and prior baselines. For **CMU-MOSI**, the **Quality-Optimized** setup achieves **78.2%** $ACC_2$, a +0.6% improvement over Vanilla IMDER (77.6%) and a +2.9% gain compared to GCNet (75.3%). The Speed-Optimized configuration remains competitive at 77.8%, still exceeding GCNet by +2.5%. In terms of F1 score, **Quality-Optimized** reaches **38.1**, slightly higher than Vanilla IMDER (37.6, +0.5) and markedly better than GCNet (35.0, +3.1).

On **CMU-MOSEI**, the **Quality-Optimized** configuration delivers **79.7%** $ACC_2$, outperforming Vanilla IMDER (79.4%) by +0.3% and GCNet (79.0%) by +0.7%. The **Speed-Optimized** setup achieves 79.1%, still matching or exceeding GCNet. F1 scores follow a similar trend, with Quality-Optimized at **49.7** (+0.3 over Vanilla, +1.8 over GCNet) and Speed-Optimized at 48.9 (comparable to Vanilla, +1.0 over GCNet).

**Table 6.10.** *Average results under the **Random Missing Protocol** derived from table 6.9.*

| | MOSI | MOSEI |
|---|---|---|
| Model | $ACC_2$ / F1 / $ACC_7$ | $ACC_2$ / F1 / $ACC_7$ |
| DCCA | 65.2 / 64.5 / 23.7 | 73.2 / 72.1 / 42.7 |
| DCCAE | 66.9 / 66.0 / 24.6 | 74.2 / 73.0 / 43.1 |
| MCTN | 71.3 / 71.2 / 33.1 | 77.1 / 76.9 / 46.8 |
| MMIN | 74.1 / 73.2 / 34.4 | 78.0 / 77.3 / 48.0 |
| GCNet | 75.3 / 74.7 / 35.0 | 79.0 / 78.3 / 47.9 |
| Vanilla IMDER | 77.6 / 77.3 / 37.6 | 79.4 / 78.8 / 49.4 |
| Quality-Opt | **78.2** / **78.0** / **38.1** | **79.7** / **79.5** / **49.7** |
| Speed-Opt | 77.8 / 77.6 / 37.7 | 79.1 / 79.0 / 48.9 |

**Table 6.11.** *Average results under the **Fixed Missing Protocol** derived from table 6.9.*

| | MOSI | MOSEI |
|---|---|---|
| Model | $ACC_2$ / F1 / $ACC_7$ | $ACC_2$ / F1 / $ACC_7$ |
| DCCA | 63.9 / 61.9 / 20.0 | 72.3 / 68.8 / 44.5 |
| DCCAE | 66.1 / 64.8 / 24.4 | 72.4 / 70.2 / 44.6 |
| MCTN | 70.2 / 69.9 / 31.3 | 74.6 / 72.5 / 46.8 |
| MMIN | 72.7 / 71.4 / 31.6 | 73.7 / 73.5 / 47.1 |
| GCNet | 73.1 / 72.8 / 32.1 | 74.7 / 73.7 / 47.1 |
| Vanilla IMDER | 75.1 / 75.0 / 34.5 | 75.1 / 75.3 / 48.2 |
| Quality-Opt | 75.2 / **75.1** / **35.1** | **76.1** / **75.6** / **48.4** |
| Speed-Opt | **75.5** / 74.5 / **35.1** | 75.2 / 74.9 / 47.7 |

Overall, while both of our configurations maintain strong advantages over GCNet, the Quality-Optimized setup provides the best robustness across missing rates, whereas the Speed-Optimized version offers a balanced trade-off with faster inference and only marginally lower accuracy.

**Computational Efficiency Advantages.** Beyond performance gains, our optimized configurations offer significant computational advantages. The **Speed-Optimized** configuration (**DDIM** with **30 NFEs**) achieves superior or comparable performance to all baselines while being **3.2**× faster than the original IMDER baseline. To add to that the ScoreTransformer1D has an additional **5x forward inference speedup** over the vanilla cross attention U-Net totalling to over **15x total speedup** for a single batch in training. This efficiency makes our approach particularly suitable for real-time applications and resource-constrained environments.

**Architectural Contributions.** The consistent improvements across both datasets and protocols validate our architectural innovations: (1) the **VP SDE formulation provides better stability** for multimodal diffusion processes, (2) the **ScoreTransformer1D backbone offers optimal parameter efficiency** while maintaining expressive power, and (3) the **DDIM or Heun samplers enable faster inference** without sacrificing quality. The combination of these components creates a synergistic effect that advances the state-of-the-art in multimodal emotion recognition with missing modalities.

**Impact of Missing Language Modality on Performance** It is a well-known issue in multimodal systems that the language modality often carries the most densely packed information. When this modality is missing, we observe a significant drop in accuracy, as the system struggles to extract useful insights from the remaining modalities. During preprocessing, we leverage a **BERT model**, which encodes extensive knowledge from pretraining—effectively **representing around 100 million parameters**. When this rich representation is removed, the diffusion system cannot fully recover the information contained in the language modality, limiting its ability to robustly predict outcomes. This problem is known in the literature as **modality collapse**.

Summary

Our multimodal diffusion framework achieves robust performance across missing modality scenarios, with **Quality-Optimized** and **Speed-Optimized** configurations balancing accuracy and efficiency. The **VP SDE formulation**, **ScoreTransformer1D**, and deterministic samplers like **DDIM** enable fast convergence, while the language modality remains crucial to prevent **modality collapse**. Overall, our approach competes with state-of-the-art in missing multimodal sentiment and emotion recognition.

## 6.5 Ablation Studies

### 6.5.1 Component Significance

To validate the importance of each component in our proposed framework, we conduct systematic ablation studies by removing key components and evaluating their impact on overall performance. This analysis helps identify the contribution of each module to the final emotion recognition accuracy.

**Experimental Setup**: We evaluate component significance using our optimal configuration (VP SDE formulation, ScoreTransformer1D backbone, and Heun sampler with 60 NFEs) on both CMU-MOSI and CMU-MOSEI datasets under the Fixed Missing Protocol.

**Table 6.12.** *Component Ablation Study Results for both datasets using our* ***Quality-optimized*** *configuration. We report the average values for the fixed missing protocol.*

| Configuration | CMU-MOSI | | | CMU-MOSEI | | |
|---|---|---|---|---|---|---|
| | ACC2 | ACC7 | F1 | ACC2 | ACC7 | F1 |
| **Full Framework (Ours)** | **75.2** | **75.1** | **35.1** | **76.1** | **75.6** | **48.4** |
| **w/o Diffusion Component** | 71.7 | 72.0 | 31.8 | 74.6 | 73.7 | 47.1 |
| **w/o Decoder Alignment** | 74.6 | 74.4 | 34.8 | 75.7 | 74.8 | 47.9 |
| **Performance Drop (w/o Diffusion)** | -3.5 | -3.1 | -3.3 | -1.5 | -1.9 | 1.3 |
| **Performance Drop (w/o Alignment)** | -0.6 | -0.7 | -0.3 | -0.4 | -0.8 | -0.5 |

**Analysis of Component Contributions**: The results highlight the critical role of the **diffusion component** in our framework. Removing **diffusion** leads to the largest decrease in performance across both datasets, particularly for binary and 7-class accuracy as well as F1 score, indicating that this module is essential for capturing nuanced multimodal dependencies.

The **decoder alignment** module also contributes positively, though to a lesser extent. Its removal results in a small but consistent drop in performance, suggesting that aligning decoder outputs across modalities improves the coherence of multimodal information and slightly enhances prediction robustness. Figure 6.1 visualizes the reconstructed quality of generated samples with and without the alignment decoder component, we observe that a majority of samples appear to be far from the distribution clusters further showcasing that the alignment component is positively enhancing.

### 6.5.2 Sampling Effectiveness

To assess the impact of sampling step of the different algorithms on reconstruction quality across different modalities, we conduct a comprehensive study using t-SNE visualizations [135]. For each modality—**vision**, **audio**, and **text**—we randomly sample 500 utterances from the dataset, retain only one modality as input, and reconstruct the remaining two using our pretrained diffusion models. The t-SNE embeddings are computed using 5,000 iterations with a perplexity of 8.

Figures 6.2,6.3,6.4 illustrates the latent space distributions of the *source* features, the *ground truth* of the missing modalities, and the *reconstructed outputs* from all of our

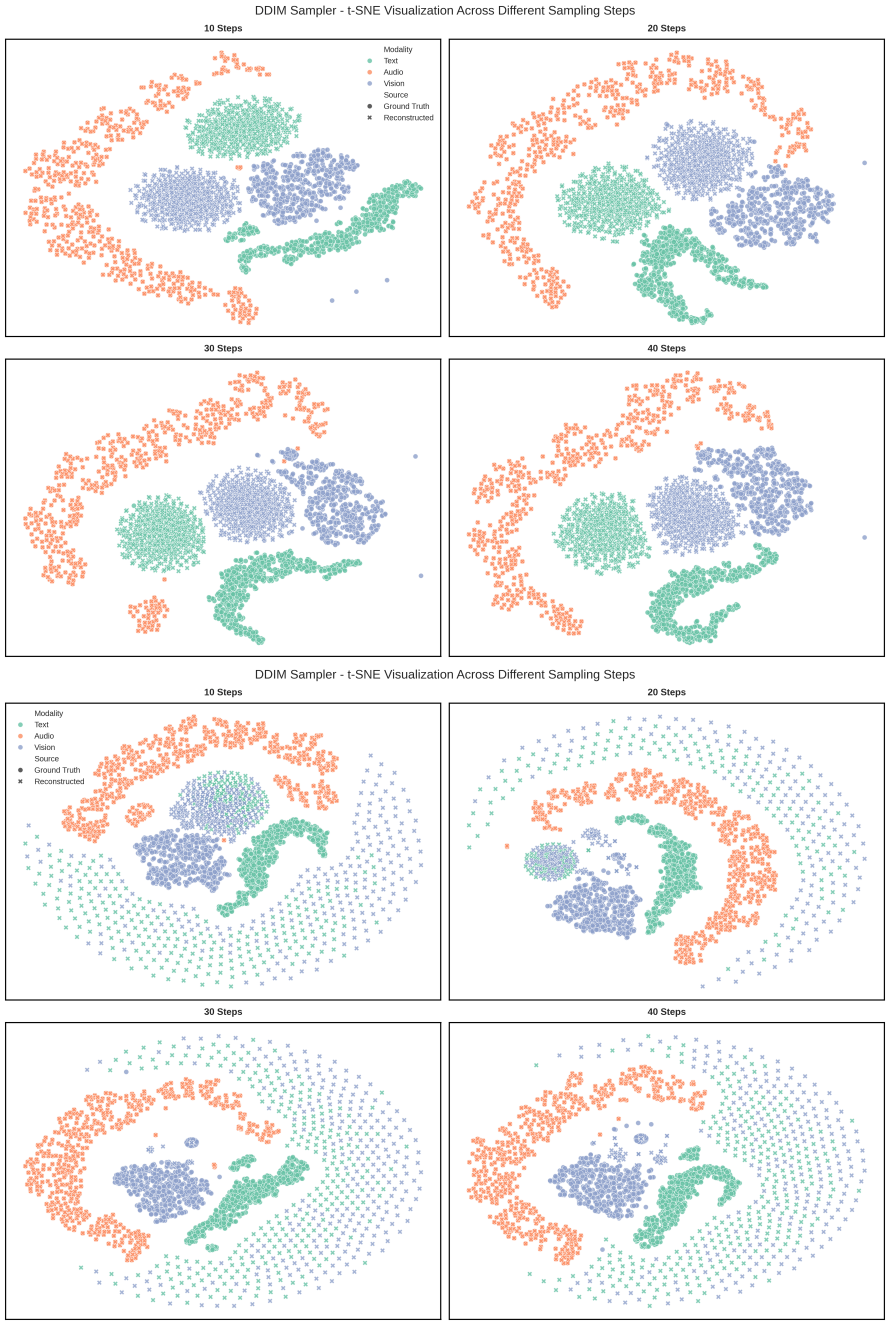


**Figure 6.1.** *t-SNE visualizations of reconstructed* ***textual and visual features*** *conditioned on* ***acoustic features*** *under different sampling steps. Top: DDIM sampler with alignment decoder (10–40 steps). Bottom: DDIM sampler without alignment decoder (10–40 steps).*

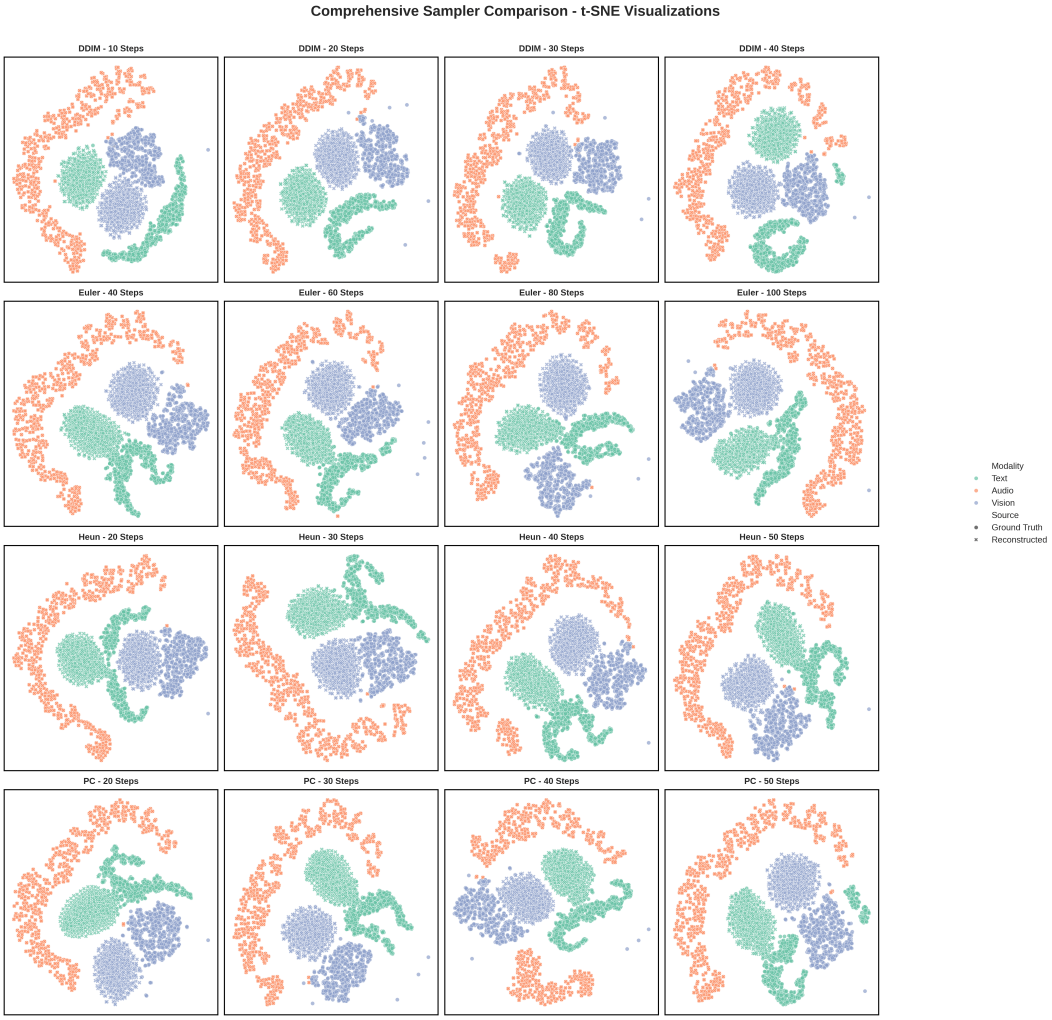


**Figure 6.2.** *t-SNE visualizations of reconstructed* ***textual and visual features*** *under different sampling steps steps for every sampler conditioning on the observed* ***acoustic features****. Ground truth features are marked with circles, reconstructed features with crosses.*


**Figure 6.3.** *t-SNE visualizations of reconstructed* ***acoustic and textual features*** *under different sampling steps steps for every sampler conditioning on the observed* ***visual features****. Ground truth features are marked with circles, reconstructed features with crosses.*


**Figure 6.4.** *t-SNE visualizations of reconstructed* ***acoustic and visual features*** *under different sampling steps for every sampler conditioning on the observed* ***textual features****. Ground truth features are marked with circles, reconstructed features with crosses.*

tested samplers under different sampling steps for each one and for three conditioning scenarios (Acoustic, Visual and Textual availability). It is evident that when the textual modality is available the clusters tend to separate more clearly and generated samples fall into the distribution cluster indicating the importance of the textual modality in multimodal settings. An additional observation is that also in general more steps indicate more defined clustering of reconstructed and ground truth modalities.

Chapter 7

# Conclusions

## 7.1 Discussion

In this work, we have presented a comprehensive study on the application of diffusion-based generative models for missing modality imputation in multimodal learning, with a focus on emotion recognition tasks. Our framework leverages the Variance Preserving (VP) SDE formulation, a parameter-efficient ScoreTransformer1D backbone, and carefully selected samplers—Heun and DDIM—to achieve robust and efficient reconstruction of absent modalities. Through systematic exploration of architectural choices, sampling algorithms, and step sizes due to our **decoupled training approach**, we identified configurations that simultaneously optimize predictive performance and computational efficiency.

**Findings from SDE and Conditioning mechanism Comparisons.** Our **Stage 1** experiments highlight that the **Variance Preserving (VP)** formulation marginally outperforms the original **Variance Exploding (VE)** approach used in IMDER. VP offers more stable and reliable improvements in $ACC_2$ and F1, especially under the random missing protocol, while VE shows occasional advantages in fine-grained $ACC_7$ classification but remains less robust overall. In **Stage 2**, we investigated backbone and conditioning architectures. While U-Net and Di-Transformer variants perform well in specific unimodal cases (e.g., U-Net with acoustic-only inputs, Di-Transformer with language), the **ScoreTransformer1D** emerges as the most balanced choice, achieving the best average performance across modalities (still remaining marginal but considerable). Importantly, ScoreTransformer1D is also the most efficient model, requiring only 3.2M parameters and offering nearly **5× faster inference** compared to U-Net with about the same memory needs. These results confirm VP-based diffusion combined with ScoreTransformer1D as the most practical and effective foundation for subsequent opti-

mization.

**Findings from Sampler Comparisons.** **Stage 3** results reveal that sampler choice plays a crucial role in balancing performance and efficiency. Traditional Euler–Maruyama with 100 NFEs provides a strong baseline (73.5 $ACC_2$) but is computationally expensive. The **Heun sampler** with 60 NFEs achieves nearly identical accuracy (73.4 $ACC_2$, 73.3 F1, 33.4 $ACC_7$) while reducing inference time by 40%, making it a robust **quality-focused option**. On the other hand, the **DDIM sampler** with 30 NFEs delivers the best overall trade-off, slightly surpassing baseline accuracy (73.8 $ACC_2$) while running **3.2× faster**. Even with as few as 10 NFEs, DDIM maintains competitive accuracy (72.9 $ACC_2$) and achieves nearly **10× faster inference**, making it suitable for real-time applications. Based on these insights, we recommend two optimal configurations: (1) a **Quality-Optimized** setup using VP SDE, ScoreTransformer1D, and Heun sampling at 60 NFEs for maximum predictive stability, and (2) a **Speed-Optimized** setup using VP SDE, ScoreTransformer1D, and DDIM sampling at 30 NFEs, which achieves superior efficiency with minimal performance loss.

**Comparison with State of the Art.** Our framework consistently outperforms existing baselines under both evaluation protocols. Under the **Fixed Missing Protocol**, the Quality-Optimized configuration improved CMU-MOSEI performance to 76.1% $ACC_2$ and 75.2 F1, representing gains of +1.3% and +2.5% over Vanilla IMDER and GCNet, respectively, with similar improvements on CMU-MOSI. Under the **Random Missing Protocol**, the Quality-Optimized setup reached 78.2% $ACC_2$ on MOSI and 79.7% on MOSEI, outperforming GCNet by up to +2.9% $ACC_2$ and +3.1 F1. The Speed-Optimized configuration achieved comparable accuracy while reducing inference time by more than 3×, confirming its suitability for real-time deployment. Overall, our approach demonstrates robustness across challenging missing data scenarios, including high missing rates and unimodal-only settings, where traditional methods often degrade severely particularly when the language modality is absent, as this often causes a significant drop in accuracy due to modality collapse.

**Final Remarks.** The combination of diffusion-based modeling, efficient transformer backbones, and optimized sampling strategies provides a practical and flexible solution to missing modality problems. Our findings show that (i) VP-based SDEs are more reliable than VE for multimodal imputation, (ii) deterministic samplers like DDIM can drastically reduce computational cost without accuracy loss, and (iii) carefully balanced

architectures such as ScoreTransformer1D can achieve high performance with fewer computational overhead. Together, these results establish diffusion models as a powerful unified framework for multimodal recovery, offering strong performance, scalability, and real-world applicability in scenarios where both robustness and inference efficiency are critical.

## 7.2 Limitations

Despite the promising results, our approach has several limitations that warrant discussion. First, the language-dominant nature of emotion recognition may also favor specific architectural choices that may not transfer to more balanced multimodal scenarios. Furthermore, the current framework is designed specifically for language, acoustic, and vision modalities. Extending it to other types of modalities, such as physiological signals, contextual information, or emerging modalities like haptic feedback, would require architectural modifications and potentially different conditioning strategies. The scalability of the framework to scenarios with more than three modalities is also uncertain.

Another limitation lies in the assumptions made about missing patterns. Our evaluation primarily focuses on random and fixed missing patterns, which may not capture the complexity of real-world situations where missing modalities often exhibit temporal dependencies, systematic biases, or correlated failures. For example, camera outages might correlate with lighting conditions, and microphone issues could be more common in noisy environments. Furthermore, the models ability to adapt to different missing scenarios such as different missing rates remains problematic, changing the missing rate or the fixed pattern hinters performance. Additionally, while our method achieves significant speedups compared to baseline diffusion approaches, it remains computationally more expensive than traditional fusion techniques or simple imputation strategies. The requirement for iterative sampling, even under optimized configurations, may limit applicability in resource-constrained settings or in real-time applications with strict latency demands.

Finally, the performance of diffusion-based models can be sensitive to hyperparameters such as noise schedules, sampling steps, and conditioning strength. Although we provide configurations that yield strong results, the robustness of these settings across different datasets or domains remains uncertain. Beyond empirical performance, the theoretical understanding of why certain combinations, such as VP SDE with DDIM sampling, are particularly effective in multimodal contexts is limited. This lack of theoretical clarity makes it difficult to predict optimal configurations for new domains or to

provide principled guidelines for architectural design.

## 7.3 Future Work

Several promising research directions emerge from our findings. First, extending the framework to a broader set of multimodal tasks could help validate its generalizability. Applications such as visual question answering, multimodal machine translation, and cross-modal retrieval may require task-specific adaptations, but the core architectural principles identified here provide a solid foundation. Real-world deployment studies are also crucial to evaluate robustness to distribution shifts, domain adaptation, and varying computational constraints. Such studies could include user evaluations to assess the practical impact of improved handling of missing modalities.

Second, improving intra-modality conditioning for partially corrupted samples is an important direction. In realistic scenarios, some frames in video or audio sequences may be missing due to network issues, or entire modalities may be partially unavailable. Conditioning the model on the observed portions of a modality, as well as on other available modalities, could provide more robust reconstructions under realistic missing protocols.

Third, developing unified frameworks that can handle arbitrary missing patterns and rates without retraining for each scenario would significantly simplify deployment. Instead of training separate models for different missing modalities or rates, a single model could adapt dynamically to various missing configurations.

Fourth, exploring faster generative alternatives to diffusion models, such as flow matching models [136], could improve efficiency. Flow matching methods rely on simpler assumptions than diffusion models, enabling faster sampling while maintaining performance.

Finally, efficiency improvements through model distillation and hardware acceleration remain important for scaling these approaches to large datasets and real-time applications. Combining these strategies could make multimodal generative systems both practical and performant in real-world settings.

# Bibliography


[1] Yuanzhi Wang, Yong Li και Zhen Cui. *Incomplete Multimodality-Diffused Emotion Recognition*. *NeurIPS*, 2023.

[2] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon και Ben Poole. *Score-based generative modeling through stochastic differential equations*. *ICLR*, 2021.

[3] Yuanzhi Wang, Yong Li και Zhen Cui. *Incomplete Multimodality-Diffused Emotion Recognition*. *Advances in Neural Information Processing Systems*A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt και S. Levine, επιμελητές, τόμος 36, σελίδες 17117–17128. Curran Associates, Inc., 2023.

[4] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek και Robin Rombach. *Scaling Rectified Flow Transformers for High-Resolution Image Synthesis*, 2024.

[5] Geoffrey Peebles και Weijian Xie. *Scalable diffusion models with transformers*. *arXiv preprint arXiv:2212.09748*, 2022.

[6] Jinming Zhao, Ruichen Li και Qin Jin. *Missing Modality Imagination Network for Emotion Recognition with Uncertain Missing Modalities*. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*Chengqing Zong, Fei Xia, Wenjie Li και Roberto Navigli, επιμελητές, σελίδες 2608–2618, Online, 2021. Association for Computational Linguistics.

[7] Lilian Weng. *From Autoencoder to Beta-VAE*. *lilianweng.github.io*, 2018.

[8] Lilian Weng. *Flow-based Deep Generative Models*. *lilianweng.github.io*, 2018.

[9] Aaronvan den Oord, Oriol Vinyals και Koray Kavukcuoglu. *Neural Discrete Representation Learning*, 2018.

[10] Thalles Santos Silva. *A Short Introduction to Generative Adversarial Networks*. *https://sthalles.github.io*, 2017.

[11] Jonathan Ho, Ajay Jain και Pieter Abbeel. *Denoising Diffusion Probabilistic Models*, 2020.

[12] Yang Song και Stefano Ermon. *Improved techniques for training score-based generative models*. *NeurIPS*, 2020.

[13] Tim Salimans και Jonathan Ho. *Progressive Distillation for Fast Sampling of Diffusion Models*, 2022.

[14] Prafulla Dhariwal και Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*, 2021.

[15] Jonathan Ho και Tim Salimans. *Classifier-Free Diffusion Guidance*, 2022.

[16] Olaf Ronneberger, Philipp Fischer και Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015.

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser και Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*, 2022.

[18] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su και Jun Zhu. *One Transformer Fits All Distributions in Multi-Modal Diffusion at Scale*, 2023.

[19] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su και Jun Zhu. *All are Worth Words: A ViT Backbone for Diffusion Models*, 2023.

[20] Paul Pu Liang, Amir Zadeh και Louis Philippe Morency. *Foundations and Trends in Multimodal Machine Learning: Principles, Challenges, and Open Questions*. 2022.

[21] Yuanzhi Wang, Zhen Cui και Yong Li. *Distribution-Consistent Modal Recovering for Incomplete Multimodal Learning*. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, σελίδες 22025–22034, 2023.

[22] Hu Wang, Yuanhong Chen, Congbo Ma, Jodie Avery, Louise Hull και Gustavo Carneiro. *Multi-modal Learning with Missing Modality via Shared-Specific Feature Modelling*, 2024.

[23] Luan Tran, Xiaoming Liu, Jiayu Zhou και Rong Jin. *Missing Modalities Imputation via Cascaded Residual Autoencoder*. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, σελίδες 4971–4980, 2017.

[24] William Peebles και Saining Xie. *Scalable Diffusion Models with Transformers*, 2023.

[25] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong και Yun Fu. *Image super-resolution using very deep residual channel attention networks*. *ECCV*, 2018.

[26] Yao Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis Philippe Morency και Ruslan Salakhutdinov. *Multimodal Transformer for Unaligned Multimodal Language Sequences*, 2019.

[27] Galen Andrew, Raman Arora, Jeff Bilmes και Karen Livescu. *Deep Canonical Correlation Analysis*. *Proceedings of the 30th International Conference on Machine Learning*Sanjoy Dasgupta και David McAllester, επιμελητές, τόμος 28 στο *Proceedings of Machine Learning Research*, σελίδες 1247–1255, Atlanta, Georgia, USA, 2013. PMLR.

[28] Weiran Wang, Raman Arora, Karen Livescu και Jeff Bilmes. *On Deep Multi-View Representation Learning: Objectives and Optimization*, 2016.

[29] Hieu Pham, Minh Thang Luong, Andrew Dai και Quoc V Le. *Found in translation: Learning robust joint representations by cyclic translations between modalities*. *AAAI*, 2019.

[30] Zheng Lian, Lan Chen, Licai Sun, Bin Liu και Jianhua Tao. *GCNet: Graph Completion Network for Incomplete Multimodal Learning in Conversation*, 2023.

[31] Tadas Baltrušaitis, Chaitanya Ahuja και Louis Philippe Morency. *Multimodal Machine Learning: A Survey and Taxonomy*, 2017.

[32] Devamanyu Hazarika, Roger Zimmermann και Soujanya Poria. *MISA: Modality-Invariant and -Specific Representations for Multimodal Sentiment Analysis*, 2020.

[33] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria και Louis Philippe Morency. *Tensor fusion network for multimodal sentiment analysis*. *EMNLP*, 2017.

[34] Mingyu Kang, Ran Zhu, Duxin Chen, Xiaolu Liu και Wenwu Yu. *CM-GAN: A Cross-Modal Generative Adversarial Network for Imputing Completely Missing*

*Data in Digital Industry*. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):2917–2926, 2024.

[35] Yuanzhi Wang, Zhen Cui και Yong Li. *Distribution-Consistent Modal Recovering for Incomplete Multimodal Learning*. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, σελίδες 21968–21977, 2023.

[36] Nuno Garcia, Pietro Morerio και Vittorio Murino. *Modality Distillation with Multiple Stream Networks for Action Recognition*. *Proceedings of the European Conference on Computer Vision (ECCV)*, σελίδες 103–118, 2018.

[37] Lei Yuan, Yalin Wang, Paul M. Thompson, Vaibhav A. Narayan και Jieping Ye. *Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data*. *NeuroImage*, 61(3):622–632, 2012.

[38] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan και Surya Ganguli. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*, 2015.

[39] Yang Song και Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*, 2020.

[40] Jiaming Song, Chenlin Meng και Stefano Ermon. *Denoising Diffusion Implicit Models*, 2022.

[41] Alex Nichol και Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*, 2021.

[42] Tero Karras, Miika Aittala, Timo Aila και Samuli Laine. *Elucidating the Design Space of Diffusion-Based Generative Models*, 2022.

[43] Yang Song, Prafulla Dhariwal, Mark Chen και Ilya Sutskever. *Consistency Models*, 2023.

[44] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho και Tim Salimans. *On Distillation of Guided Diffusion Models*, 2023.

[45] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li και Jun Zhu. *DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps*, 2022.

[46] Amir Zadeh, Rowan Zellers, Eli Pincus και Louis Philippe Morency. *Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages*. *IEEE Intelligent Systems*, τόμος 31, σελίδες 82–88, 2016.

[47] AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria και Louis Philippe Morency. *Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph*. *ACL*, 2018.

[48] Olaf Ronneberger, Philipp Fischer και Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015.

[49] Ethan Perez, Florian Strub, Harmde Vries, Vincent Dumoulin και Aaron Courville. *FiLM: Visual Reasoning with a General Conditioning Layer*, 2017.

[50] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nalini Raghavan, Jonathan T Barron και Ren Ng. *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*. *NeurIPS*, 2020.

[51] Jie Hu, Li Shen και Gang Sun. *Squeeze-and-excitation networks*. *CVPR*, 2018.

[52] Jacob Devlin, Ming Wei Chang, Kenton Lee και Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, σελίδες 4171–4186. Association for Computational Linguistics, 2019.

[53] Gilles Degottex, John Kane, Thomas Drugman, Tuomo Raitio και Stefan Scherer. *COVAREP—A collaborative voice analysis repository for speech technologies*. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, σελίδες 960–964. IEEE, 2014.

[54] iMotions. *Facial Expression Analysis Module*, 2017. Available at https://imotions.com/products/imotions-lab/modules/fea-facial-expression-analysis/.

[55] Diederik P. Kingma και Jimmy Ba. *Adam: A Method for Stochastic Optimization*, 2017.

[56] Yifei Zhang, Kun Han, Zizhao Zhang και Changsheng Xu. *Deep multimodal representation learning: A survey*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[57] Sindhu Parthasarathy και Themos Stafylakis. *Training strategies for improved lip-reading*. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, σελίδες 8472–8476. IEEE, 2020.

[58] Yang Yang, De Chuan Zhan, Xiang Rong Sheng και Yuan Jiang. *Semi-supervised multi-modal learning with incomplete modalities*. *IJCAI*, σελίδες 2998–3004, 2018.

[59] Yuxin Liang και Patrik Floréen. *Learning representations from imperfect time series data via tensor rank regularization*. *Proceedings of the AAAI Conference on Artificial Intelligence*, τόμος 33, σελίδες 4275–4282, 2019.

[60] Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer και C Lee Giles. *Learning to extract semantic structure from documents using multimodal fully convolutional neural networks*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, σελίδες 5315–5324, 2017.

[61] Jian Feng Cai, Emmanuel J Candès και Zuowei Shen. *A singular value thresholding algorithm for matrix completion*. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[62] Rahul Mazumder, Trevor Hastie και Robert Tibshirani. *Spectral regularization algorithms for learning large incomplete matrices*. *Journal of Machine Learning Research*, 11(Aug):2287–2322, 2010.

[63] Haoyang Fan, Yue Chen, Yifei Guo, Hongyan Zhang και Gangyao Kuang. *Hyperspectral image restoration using low-rank tensor recovery*. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(10):4589–4604, 2017.

[64] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[65] Pascal Vincent, Hugo Larochelle, Yoshua Bengio και Pierre Antoine Manzagol. *Extracting and composing robust features with denoising autoencoders*. *Proceedings of the 25th international conference on Machine learning*, σελίδες 1096–1103, 2008.

[66] Luan Tran, Xiaoming Liu, Jiayu Zhou και Rong Jin. *Missing modalities imputation via cascaded residual autoencoder*. σελίδες 1405–1414, 2017.

[67] Qianqian Wang, Zhengming Ding, Zhiqiang Tao, Quanxue Gao και Yun Fu. *Partial multi-view clustering via consistent gan*. *IEEE International Conference on Data Mining (ICDM)*, σελίδες 1290–1295. IEEE, 2018.

[68] Lei Cai, Zhengzhang Wang, Hongyang Gao, Dinggang Shen και Shuiwang Ji. *Deep adversarial learning for multi-modality missing data completion*. *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, σελίδες 1158–1166, 2018.

[69] Oleg Ivanov, Michael Figurnov και Dmitry Vetrov. *Variational autoencoder with arbitrary conditioning*. *Proceedings of the 7th International Conference on Learning Representations*, σελίδες 1–25, 2019.

[70] Cheng Du, Changde Du, Hengwei Wang, Jia Li, Wei Long Zheng, Bao Liang Lu και Huiguang He. *Semi-supervised deep generative modelling of incomplete multi-modality emotional data*. *Proceedings of the 26th ACM international conference on Multimedia*, σελίδες 108–116, 2018.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser και Illia Polosukhin. *Attention is All You Need*. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[72] Ziqi Yuan, Wei Li, Hua Xu και Wenmeng Yu. *Transformer-based feature reconstruction network for robust multimodal sentiment analysis*. *Proceedings of the 29th ACM International Conference on Multimedia*, σελίδες 4400–4407, 2021.

[73] Jinming Zhao, Ruichen Li και Qin Jin. *Missing modality imagination network for emotion recognition with uncertain missing modalities*. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, σελίδες 2608–2618, 2021.

[74] Yanjie Duan, Yisheng Lv, Wenwen Kang και Yifei Zhao. *A deep learning based approach for traffic data imputation*. *17th International IEEE conference on intelligent transportation systems (ITSC)*, σελίδες 912–917. IEEE, 2014.

[75] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. *Deep residual learning for image recognition*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, σελίδες 770–778, 2016.

[76] Lei Yuan, Yalin Wang, Paul M Thompson, Vadim A Narayan, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative και others. *Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data*. *NeuroImage*, 61(3):622–632, 2012.

[77] Shuo Xiang, Lei Yuan, Wei Fan, Yalin Wang, Paul M Thompson και Jieping Ye. *Multi-source learning with block-wise missing data for alzheimer's disease prediction*. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, σελίδες 185–193, 2013.

[78] Yiming Li, Tian Yang, Jiayu Zhou και Jieping Ye. *Multi-task learning based survival analysis for predicting alzheimer's disease progression with multi-source block-wise missing data*. *Proceedings of the 2018 SIAM international conference on data mining*, σελίδες 288–296. SIAM, 2018.

[79] Harold Hotelling. *Relations between two sets of variates*. *Breakthroughs in statistics*, σελίδες 162–190. Springer, 1992.

[80] Galen Andrew, Raman Arora, Jeff Bilmes και Karen Livescu. *Deep canonical correlation analysis*. *ICML*, 2013.

[81] Fei Ma, Shao Lun Huang και Lin Zhang. *An efficient approach for audio-visual emotion recognition with missing labels and missing modalities*. *2021 IEEE International Conference on Multimedia and Expo (ICME)*, σελίδες 1–6. IEEE, 2021.

[82] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyao Li, Jiancheng Lv και Xi Peng. *Completer: Incomplete multi-view clustering via contrastive prediction*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 11174–11183, 2021.

[83] Fei Ma, Xiangxiang Xu, Shao Lun Huang και Lin Zhang. *Maximum likelihood estimation for multimodal learning with missing modality*. *arXiv preprint arXiv:2108.10513*, 2021.

[84] Weiran Wang, Raman Arora, Karen Livescu και Jeff Bilmes. *On deep multi-view representation learning*. *ICML*, 2015.

[85] Amir Zadeh, Yao Chong Lim, Paul Pu Liang και Louis Philippe Morency. *Variational auto-decoder: A method for neural generative modeling from incomplete data*. *arXiv preprint arXiv:1903.00840*, 2019.

[86] Amir Zadeh, Samuel Benoit και Louis Philippe Morency. *Relay variational inference: A method for accelerated encoderless vi*. *arXiv preprint arXiv:2110.13422*, 2021.

[87] Ian Goodfellow, Yoshua Bengio και Aaron Courville. *Deep Learning*. *Deep Learning*Ian Goodfellow, Yoshua Bengio και Aaron Courville, επιμελητές, κεφάλαιο 14. MIT Press, 2016.

[88] Diederik P Kingma και Max Welling. *Auto-Encoding Variational Bayes*, 2022.

[89] Danilo Rezende και Shakir Mohamed. *Variational inference with normalizing flows*. *International conference on machine learning*, σελίδες 1530–1538. PMLR, 2015.

[90] Danilo Jimenez Rezende και Shakir Mohamed. *Variational Inference with Normalizing Flows*, 2016.

[91] Laurent Dinh, David Krueger και Yoshua Bengio. *NICE: Non-linear Independent Components Estimation*. *arXiv preprint arXiv:1410.8516*, 2014.

[92] Laurent Dinh, Jascha Sohl-Dickstein και Samy Bengio. *Density estimation using Real NVP*. *International Conference on Learning Representations*, 2017.

[93] George Papamakarios, Theo Pavlakou και Iain Murray. *Masked Autoregressive Flow for Density Estimation*. *Advances in Neural Information Processing Systems*, 2017.

[94] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever και Max Welling. *Improving Variational Inference with Inverse Autoregressive Flow*. *Advances in Neural Information Processing Systems*, 2016.

[95] Durk P Kingma και Prafulla Dhariwal. *Glow: Generative Flow with Invertible 1x1 Convolutions*. *Advances in Neural Information Processing Systems*, 2018.

[96] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever και David Duvenaud. *FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models*. *International Conference on Learning Representations*, 2019.

[97] Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud και Jörn Henrik Jacobsen. *Invertible Residual Networks*. *International Conference on Machine Learning*, 2019.

[98] Ali Razavi, Aaronvan den Oord και Oriol Vinyals. *Generating Diverse High-Fidelity Images with VQ-VAE-2*, 2019.

[99] Aaronvan den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior και Koray Kavukcuoglu. *WaveNet: A Generative Model for Raw Audio*, 2016.

[100] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen και Ilya Sutskever. *Zero-Shot Text-to-Image Generation*, 2021.

[101] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville και Yoshua Bengio. *Generative Adversarial Networks*, 2014.

[102] Martin Arjovsky, Soumith Chintala και Léon Bottou. *Wasserstein GAN*. *International Conference on Machine Learning*, 2017.

[103] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang και Stephen Paul Smolley. *Least Squares Generative Adversarial Networks*. *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[104] Tero Karras, Samuli Laine και Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, σελίδες 4401–4410, 2019.

[105] Yang Song και Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*, 2020.

[106] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon και Ben Poole. *Score-Based Generative Modeling through Stochastic Differential Equations*, 2021.

[107] Yang Song και Stefano Ermon. *Improved Techniques for Training Score-Based Generative Models*, 2020.

[108] Jiaming Song, Chenlin Meng και Stefano Ermon. *Denoising Diffusion Implicit Models*, 2022.

[109] Yuxin Wu και Kaiming He. *Group Normalization*, 2018.

[110] Andrew Brock, Jeff Donahue και Karen Simonyan. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*, 2019.

[111] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser και Jianxiong Xiao. *LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop*, 2016.

[112] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. *Deep Residual Learning for Image Recognition*, 2015.

[113] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit και Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, 2021.

[114] Tadas Baltrušaitis, Chaitanya Ahuja και Louis Philippe Morency. *Multimodal machine learning: A survey and taxonomy*. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

[115] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee και Andrew Y Ng. *Multimodal deep learning*. *Proceedings of the 28th international conference on machine learning*, σελίδες 689–696, 2011.

[116] Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu και Xi Peng. *SMIL: Multimodal learning with severely missing modality*. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):2302–2310, 2021.

[117] Soujanya Poria, Erik Cambria, Rajiv Bajpai και Amir Hussain. *A review of affective computing: From unimodal analysis to multimodal fusion*. *Information fusion*, 37:98–125, 2017.

[118] Jacob Devlin, Ming Wei Chang, Kenton Lee και Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. *Proceedings of NAACL-HLT*, 2019.

[119] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik και Mohan S Kankanhalli. *Multimodal fusion for multimedia analysis: a survey*. *Multimedia systems*, 16(6):345–379, 2010.

[120] Cees GM Snoek, Marcel Worring και Arnold WM Smeulders. *Early versus late fusion in semantic video analysis*. *Proceedings of the 13th ACM international conference on Multimedia*, σελίδες 399–402, 2005.

[121] Josef Kittler, Mohamad Hatef, Robert PW Duin και Jiri Matas. *On combining classifiers*. *IEEE transactions on pattern analysis and machine intelligence*, 20(3):226–239, 1998.

[122] Jiawei Wu, Yinan Yu, Chang Huang και Kai Yu. *Deep cross-modal learning for audio-visual recognition*. *Proceedings of the 21st ACM international conference on Multimedia*, σελίδες 123–132, 2013.

[123] Jiasen Lu, Dhruv Batra, Devi Parikh και Stefan Lee. *ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks*. *Advances in neural information processing systems*, σελίδες 13–23, 2019.

[124] Hao Tan και Mohit Bansal. *LXMERT: Learning cross-modality encoder representations from transformers*. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, σελίδες 5100–5111, 2019.

[125] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark και others. *Learning transferable visual models from natural language supervision*. *International conference on machine learning*, σελίδες 8748–8763. PMLR, 2021.

[126] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu και Mark Chen. *Hierarchical Text-Conditional Image Generation with CLIP Latents*, 2022.

[127] Amanpreet Singh, Ronghang Hu, Vedanuj Khalid και others. *MMBT: Multimodal Bitransformers for Large-Scale Multimodal Representation Learning*. *Proceedings of ICLR*, 2020.

[128] Jean Baptiste Alayrac, Jeff Donahue, Paul Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds και others. *Flamingo: a Visual Language Model for Few-Shot Learning*. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[129] Efthymios Georgiou. *Multimodal representation learning with application in sentiment analysis*. Διδακτορική Διατριβή, National Technical University of Athens (NTUA), School of Electrical and Computer Engineering, 2025.

[130] Lan Jiang, Ye Mao, Xi Chen, Xiangfeng Wang και Chao Li. *CoLa-Diff: Conditional Latent Diffusion Model for Multi-Modal MRI Synthesis*, 2023.

[131] Natalia Neverova, Christian Wolf, Graham Taylor και Florian Nebout. *ModDrop: adaptive multi-modal gesture recognition*. τόμος 38, σελίδες 1692–1706. IEEE, 2015.

[132] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser και Björn Ommer. *High-resolution image synthesis with latent diffusion models*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 10684–10695, 2022.

[133] Qi Fan, Haolin Zuo, Rui Liu, Zheng Lian και Guanglai Gao. *Learning Noise-Robust Joint Representation for Multimodal Emotion Recognition under Realistic Incomplete Data Scenarios*, 2023.

[134] Meng Sun, Zhiyao Yu, Kun Zhou και Enhong Chen. *GCNet: Gated cross-modal connections for multimodal emotion recognition*. *ACM MM*, 2020.

[135] Laurensvan der Maaten και Geoffrey Hinton. *Visualizing data using t-SNE*. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[136] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel και Matt Le. *Flow Matching for Generative Modeling*, 2023.