UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Deep Variational Generative Models: Theory and Algorithms

by Zhangkai Wu

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Longbing Cao

University of Technology Sydney
Faculty of Engineering and Information Technology

January 2025

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Zhangkai Wu* declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science* at the *Faculty of Engineering and Information Technology* at the University of Technology Sydney, Australia. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

DATE: 9th November, 2024

PLACE: Sydney, Australia

Abstract

Deep Variational Generative Models (DVGMs) represent a powerful class of generative models that combine variational inference with deep learning architectures. By leveraging the representational strength of deep neural networks and the probabilistic framework of variational inference, DVGMs have advanced the ability to model complex, high-dimensional data distributions, enabling them to effectively handle images, sequences, time-series, and tabular data, thereby extending their impact across machine learning, computer vision, data analysis, and natural language processing. These models, by uniting the strengths of deep learning with Bayesian principles, provide a flexible approach to understanding intricate data structures and have opened new pathways for efficient representation learning and high-quality generation.

Despite their strengths, DVGMs face notable gaps between probabilistic inference and deep generation, raising several key questions: (1) How can DVGMs balance Bayesian inference with the depth required for generative tasks? (2) How can they manage the trade-off between inference-driven representation and data fitting? (3) How can inference assumptions be leveraged to ensure robust generation? (4) How do probabilistic assumptions be designed to generate cross-modality? (5) How can DVGMs achieve consistent inference within dynamic generation processes? These questions underscore the challenges limiting DVGMs' potential in practical applications requiring flexible, reliable, and interpretable data generation.

This thesis systematically studies how to effectively address these challenges, providing both experimental and theoretical support. Given the intricate balance required between inference and generation in DVGMs, it is crucial to integrate information-theoretic principles and adaptive mechanisms to enhance DVGM performance across diverse tasks. Specifically, this thesis proposes five novel methods to tackle these issues. The main ideas include employing information-theoretic approaches to train DVGMs, introducing adaptive balancing mechanisms to dynamically adjust inference and generation based on data characteristics, and designing task-specific DVGM structures tailored for various data types. These innovations aim to strengthen representation disentanglement, improve robustness to noise, and increase scalability, enabling DVGMs to handle high-dimensional, noisy, and time-sensitive data effectively. Through these advancements, the thesis establishes a solid foundation for enhancing DVGMs' applicability in complex, real-world scenarios and provides new directions for future research in generative modeling.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Longbing Cao, for his patient and invaluable guidance throughout my doctoral studies. He taught me not only how to conduct in-depth research but also helped shape my research taste, which will benefit me throughout my career. Through his mentorship, I learned to identify meaningful research questions, develop original algorithms, and refine my technical writing, ultimately growing into an independent researcher. His encouragement to explore advanced research directions, meticulous feedback on my papers, and generosity in sharing his knowledge have been invaluable. I am truly fortunate to have had the opportunity to be mentored by Prof. Longbing Cao during this critical stage of my academic journey.

I am also immensely grateful to Lecturer Xuhui Fan for his invaluable support and mentorship. I owe special thanks to him for guiding me in learning probabilistic generative models. His solid theoretical knowledge and his enthusiasm for exploring new ideas have greatly inspired me. His profound expertise, insightful perspectives, and unwavering encouragement have been a constant source of motivation for me. I am deeply thankful for his commitment to my academic growth.

I would also like to extend my sincere gratitude to the members of my candidature assessment panel, A/Prof. Wei Liu, for their constructive feedback and suggestions that have helped shape my work.

Furthermore, I would like to thank all my colleagues for their support in both my research and daily life, as well as their critical insights into my work: Qi Zhang, Qing Liu, Zhilin Zhao, Wenfeng Hou, Oliver Qi, Hu Cao, James Yang, Hengyu Liu, Hui Chen, Junxian Zhou, Xiang Li, Runze Yang, Yongmin Yoo, Peizhen Li, Matthew Lai, Xiaoming Wu, Xianxun Zhu, Jianbo Zheng, Manqing Dong.

I would also like to extend my heartfelt gratitude to my master's classmates, Yanfang Liu and Jian Zhang, from Nanjing University. Though far away, our discussions on research trends and their genuine passion for pursuing science have made me feel less alone on this journey. I am equally thankful for their support in helping me build mental resilience, offering empathetic comfort during challenging times, and guiding me through one dark moment after another. Their companionship and encouragement have been invaluable in my academic and personal growth.

Finally, I wish to express my deepest gratitude to my family-my parents, my sister and brother-in-law, and my two lively, adorable nieces. My family has always been my greatest source of strength and motivation, nurturing me with values and character that

guide me to this day. I am also deeply grateful to my late grandmother, who raised me with love, taking such good care of me and providing a foundation that has carried me through every challenge. Her unwavering support and presence in my life have been a constant source of comfort and inspiration, and I carry her memory with me in all that I do.

List of Publications

Related to the Thesis :

1. **Zhangkai Wu**, Longbing Cao, Lei Qi. eVAE: Evolutionary variational autoencoder, *IEEE Transactions on Neural Networks and Learning Systems (IEEE-TNNLS)*. [ERA&CORE: A*, JCR Q1]

2. **Zhangkai Wu** and Cao, Longbing. C^{2}VAE: Gaussian Copula-based VAE Differing Disentangled from Coupled Representations with Contrastive Posterior, under review in *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE-TPAMI)*. [ERA&CORE: A*, JCR Q1] (Submitted ID: TPAMI-2023-10-2267)

3. **Zhangkai Wu**, Longbing Cao, Qi Zhang, Junxian Zhou, Hui Chen. Weakly Augmented Variational Autoencoder in Time Series Anomaly Detection. under review in *IEEE Transactions on Neural Networks and Learning Systems (IEEE-TNNLS)*. [ERA&CORE: A*, JCR Q1] (Submitted ID: TNNLS-2024-P-35170)

4. **Zhangkai Wu**, Xuhui Fan, Jin Li, Zhilin Zhao, Hui Chen, Longbing Cao. ParamReL: Learning Parameter Space Representation via Progressively Encoding Bayesian Flow Networks. *31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2025)*. [ERA&CORE: A*]

5. **Zhangkai Wu**, Xuhui Fan, Longbing Cao. ProgDiffusion: Efficient Unconditional Sampling with Progressive Semantic Representations for Diffusion Autoencoders, *31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2025)*. [ERA&CORE: A*]

6. **Zhangkai Wu**, Xuhui Fan, Hongyu Wu, Longbing Cao. SCoT: Straight Consistent Trajectory for Pre-Trained Diffusion Model Distillations, under reviewer in NeurIPS 2025.[ERA&CORE: A*]

7. **Zhangkai Wu**, Xuhui Fan, Zhongyuan Xie, Kaize Shi, Zhidong Li, Longbing Cao. FAME: Fairness-aware Attention-modulated Video Editing, under reviewer in AAAI 2026.[ERA&CORE: A*]

8. **Zhangkai Wu**, Xuhui Fan, Zhongyuan Xie, Kaize Shi, Longbing Cao. VALA: Learning Latent Anchors for Training-Free and Temporally Consistent Video Editing, under reviewer in AAAI 2026.[ERA&CORE: A*]

OTHERS :

9. Hui Chen, Xuhui Fan, **Zhangkai Wu**, Longbing Cao. FigBO: A Generalized Acquisition Function Framework with Look-Ahead Capability for Bayesian Optimization, under review in Machine Learning. [JCR Q1]

10. Hui Chen, Hengyu Liu, **Zhangkai Wu**, Xuhui Fan, Longbing Cao. FedSI: Federated Subnetwork Inference for Efficient Uncertainty Quantification. *IEEE Transactions on Neural Networks and Learning Systems (IEEE-TNNLS)*. [ERA&CORE: A*, JCR Q1]

11. Hongyu Wu, Xuhui Fan, **Zhangkai Wu**, Longbing Cao. Nested AutoRegressive Models, under review in AAAI 2026.[ERA&CORE: A*]

12. Xuhui Fan, **Zhangkai Wu**, Hongyu Wu, Longbing Cao. A Survey on Pre-Trained Diffusion Model Distillations, under review in AAAI 2026.[ERA&CORE: A*]

Table of Contents

List of Figures

Figure **Page**

LIST OF TABLES

Chapter 1

Introduction

In this chapter, we briefly introduce the definition of deep variational generative model, related challenges and questions, thesis contributions, and finally show the framework of the entire thesis.

1.1 Background

Modern machine learning systems increasingly rely on generative models to understand, simulate, and manipulate complex data. However, several fundamental challenges remain unresolved: how to balance expressive data modeling with principled uncertainty estimation, how to achieve robust generation under limited supervision, and how to design models that generalize across modalities and evolving data distributions. These issues are especially critical in real-world scenarios such as medical diagnosis, where models must not only predict outcomes from heterogeneous clinical data (e.g., images, text, and signals), but also provide reliable uncertainty estimates to support decision-making. Similarly, in applications like language generation or scientific discovery, generating plausible yet controllable outputs under data sparsity or structural ambiguity remains a persistent challenge.

To address these problems, deep variational learning has emerged as a powerful framework that combines the strengths of variational inference with the representational capacity of deep neural networks. This approach enables scalable, uncertainty-aware modeling of complex data, offering a principled yet flexible foundation for tasks such as generative modeling, anomaly detection, and semi-supervised learning. Within this paradigm, Deep Variational Generative Models (DVGM) have been developed to extend conventional architectures like VAEs, aiming to tackle problems such as hierarchical representation, dynamic inference, and multimodal generation with enhanced fidelity and interpretability.

The development of deep generative models has undergone significant advancements, evolving from foundational models like Variational Autoencoders (VAEs) [76] to advanced frameworks such as Diffusion Models (DMs) [62, 131–134, 136], Bayesian Flow Networks (BFNs) [54], and Flow Matching [5, 95, 96]. These models have demonstrated exceptional capabilities in modelling high-dimensional data, showing robust performance across a wide range of applications, including machine learning, data analysis, and computer vision. Deep generative models are now widely used in tasks such as density estimation, modelling sequential and tabular data, image data editing for computer vision, and representation learning. Each of these advancements has contributed to improved accuracy and interpretability in complex data distributions, thus enhancing the capacity of generative models to deliver impactful results across various domains.

1.2 Motivation

DVGMs are integral to modern data modeling, enabling powerful inference and generation capabilities across high-dimensional and complex data domains. However, achieving a seamless integration of inference and generation within DVGMs remains challenging. In particular, DVGMs face inherent trade-offs in balancing these dual tasks, with notable gaps between amortized inference and the capabilities of deep learning frameworks. Additionally, DVGMs encounter difficulties in reliably capturing intricate data representations and generating accurate samples under varied data conditions. These issues limit the robustness and versatility of DVGMs in practical applications, such as anomaly detection, structured image editing, and time series analysis, where models must be flexible, adaptable, and highly accurate. Thus, there is a pressing need to advance DVGM architectures to better address the intricate dynamics between inference and generation.

To advance the theoretical and practical potential of DVGMs, this thesis identifies five overarching challenges that are pivotal to enhancing their effectiveness:

1. how to establish a flexible framework within DVGMs to dynamically balance inference and generation tasks,

2. how to develop reliable mechanisms for DVGMs to accurately represent and disentangle latent factors in complex image data,

3. how to improve DVGM robustness when inferring representations in time series data environments with sparse or noisy signals,

4. how to enhance the adaptability of DVGMs in capturing diverse latent structures in discrete and continuous data for more reliable inferences across varied conditions,

5. how to facilitate the generation of meaningful, structured representations in image data that support both interpretability and utility in downstream tasks, and how to enable fine-grained, adaptable control in generation processes for image data to meet specific semantic demands within the latent space.

This thesis offers an in-depth exploration of these challenges, proposing innovative solutions aimed at strengthening the integration of inference and generation in DVGMs and extending their applicability across a broader range of complex, real-world tasks.

1.3 Research Questions and Objectives

This thesis aims to develop novel theoretical and algorithmic frameworks to improve the balance of inference and generation in Deep Variational Generative Models (DVGMs), with a particular focus on enhancing adaptability, robustness, and semantic quality across a range of complex data types. The research is guided by the following questions and corresponding objectives:

RQ1. How can evolutionary mechanisms balance inference and generation in DVGMs? Balancing inference and generation is critical to effective DVGM training. Existing approaches rely on static hyperparameters, often leading to KL vanishing and suboptimal representations. To address this, we propose an *evolutionary VAE (eVAE)* that applies variational genetic algorithms to dynamically fine-tune this balance, enabling the model to adapt to different data structures without manual intervention.

RQ2. How can DVGM calibrate inference to separate disentangled and coupled representations? Many existing models suffer from entangled latent factors, reducing interpretability and stability. To tackle this, we introduce *Contrastive Copula VAE (C^{2}VAE)*, which employs contrastive learning and a self-supervised classifier to refine disentangled representations and mitigate coupling.

RQ3. How can weak augmentation improve inference robustness in DVGMs for anomaly detection? Sparse and noisy anomalies in time series data challenge DVGM inference. To overcome this, we develop a *Weakly Augmented VAE (WAVAE)* that integrates weak augmentation and contrastive learning to enhance the model,Äôs anomaly sensitivity and reconstruction accuracy.

RQ4. How can DVGMs enhance inference in complex parameter spaces for better generation? Modeling high-level semantics from both continuous and discrete data requires flexible latent structures. We address this by proposing *ParamReL*, a framework that extracts semantic representations directly from parameter spaces using progressive encoding, improving generative accuracy and adaptability.

RQ5. How can progressive inference facilitate low-dimensional generation in diffusion models? Diffusion models often struggle with compact latent representations. To solve this, we introduce *ProgDiffusion*, a diffusion-based model incorporating a self-encoding mechanism to generate timestep-specific semantic representations for efficient, structure-aware generation.

1.4 Research Innovations

This thesis aims to advance Deep Variational Generative Models (DVGMs) by enhancing their adaptability and robustness across diverse data types and application scenarios. The main contributions of this study are summarized as follows:

Innovation 1.

This study pioneers an evolutionary mechanism within DVGMs, enabling a dynamic balance between inference and generation that is adaptable throughout training. This innovation increases the adaptability of DVGMs to various data structures and tasks by automatically adjusting model trade-offs, thereby improving their robustness across different application contexts without manual hyperparameter tuning.

Innovation 2.

To address the limitations in handling complex image data, this research introduces a Contrastive Copula VAE (C^{2}VAE) that utilizes contrastive learning to separate disentangled from coupled latent factors. This approach strengthens the inference process within DVGMs, improving their capacity to manage high-dimensional image data by enhancing the stability and clarity of latent representations, which is essential for image-based applications.

Innovation 3.

This study introduces a Weakly Augmented VAE (WAVAE) specifically designed to enhance inference robustness in time series anomaly detection. By incorporating weak augmentation and self-supervised learning, WAVAE demonstrates an advanced ability to handle noisy and sparse signals in time series data, making it more adaptable to real-world anomaly detection scenarios where data quality and density vary.

Innovation 4.

A novel framework, ParamReL, is developed to extract meaningful semantics from complex parameter spaces, enabling DVGMs to process both discrete and continuous data effectively. This framework advances DVGM adaptability by allowing progressive encoding directly in parameter space, ensuring accurate and context-aware generation that supports a broader range of data types and applications.

Innovation 5.

This thesis proposes a Progressive Diffusion model, ProgDiffusion, which leverages progressive inference to generate low-dimensional semantic representations in diffusion models. By aligning latent changes over time, this approach allows DVGMs to generate structured and efficient representations, enhancing their suitability for applications requiring compact representations, such as data synthesis and feature extraction.

1.5 Research Contributions

This thesis makes four primary contributions to the advancement of Deep Variational Generative Models (DVGM), addressing theoretical and practical challenges related to inference, generation, and generalization across data types and architectures.

Contribution 1. A Generalized Framework for DVGM Design.

Despite the success of VAEs and related models, existing DVGM research often focuses on narrow architectural variants without a unifying structure for handling complex data distributions. This thesis proposes a comprehensive and generalizable DVGM framework that unifies variational inference with deep generative modeling, integrating various mechanisms such as latent diffusion, copula-based representation, and Bayesian flow. This framework provides a foundation for systematically designing and comparing DVGMs under a consistent set of principles, bridging gaps between separately developed methods and enabling scalable modeling across tasks.

Contribution 2. Novel Inference-Centered DVGM Architectures.

Effective variational inference in deep models remains a core challenge, particularly in dynamic or structurally complex data scenarios. To address this, several novel models are proposed:

1) *eVAE*, an evolutionary variational autoencoder, introduces a genetic optimization mechanism that adaptively balances the trade-off between inference and generation, overcoming issues like KL vanishing and enhancing disentangled representation learning.

2) *ProgDiffusion* rethinks diffusion-based generation by embedding a progressive self-encoding mechanism into the latent space, enabling low-dimensional and semantically consistent generation, especially in high-dimensional domains.

3) *ParamReL* extends inference capabilities to the parameter space, enabling DVGMs to directly learn semantic representations from model parameters and improve generalization across both discrete and continuous data domains.

Contribution 3. Cross-Domain Application and Data Adaptability.

Most DVGM models are benchmarked on narrow data types (e.g., images), limiting their applicability. This thesis systematically adapts the proposed models to multiple domains:

1) For image data, C^2*VAE* introduces a contrastive copula-based design that disentangles latent factors while mitigating coupling, improving interpretability in vision applications.

2) For time series anomaly detection, *WAVAE* incorporates weak augmentation and contrastive learning to strengthen inference robustness under sparse and noisy anomaly conditions.

3) For discrete and textual data, both *ParamReL* and *eVAE* are adapted to handle structural irregularities, demonstrating the DVGM framework,Äôs versatility beyond continuous domains.

Contribution 4. Practical Scalability and Downstream Utility.

A growing concern in modern AI is scaling generative models while maintaining interpretability and control. This thesis explores how the proposed DVGM architectures can scale to larger models and be adapted for downstream tasks:

1) Design strategies are introduced to extend DVGMs to deeper networks and broader datasets while preserving inference stability.

2) Applied studies,Äîsuch as image editing via region control and time series anomaly detection,Äîdemonstrate how the proposed methods improve downstream performance, offering practical benefits in real-world AI systems.

1.6 Research Significance

The theoretical and practical significance of this thesis is summarized as follows:

1.6.1 Theoretical Significance

This thesis makes significant contributions to the theoretical foundation of DVGMs by advancing model adaptability across various inference and generation tasks. The theoretical innovations in this research have strong potential to guide future studies in generative modeling, especially in the fields of structured representation learning, anomaly detection, and flexible image editing.

The findings on evolutionary mechanisms in DVGMs provide a new framework for balancing inference and generation in generative models, introducing a theoretical basis for dynamic adaptability in generative tasks. This framework encourages future researchers to explore more adaptive balancing methods, enhancing DVGM performance in diverse data environments.

This thesis also introduces novel inference mechanisms tailored to different DVGM architectures, including VAEs, diffusion models, and Bayesian flow networks. By examining the theoretical properties of contrastive and progressive inference methods, this research lays a foundation for improved representation disentanglement and low-dimensional generation, expanding the capabilities of DVGMs in data synthesis and latent structure learning.

Further, this research offers a structured approach to enhancing DVGM inference across data types, such as visual, time series, and discrete data. Theoretical advancements in handling these diverse data types with tailored models provide a foundation for more robust DVGMs, inspiring further studies on data-type-specific generative modeling.

1.6.2 Practical Significance

The practical significance of this thesis lies in its potential to address the increasing demand for adaptable generative models across a variety of real-world applications. The proposed DVGM frameworks and models are validated on real-world datasets, ensuring that the methods developed in this research can be applied directly to practical problems in fields such as computer vision, natural language processing, and anomaly detection.

In particular, the new DVGM models for visual data, time series anomaly detection, and flexible image editing extend the applicability of generative models in areas where data diversity and noise present significant challenges. Practitioners can leverage these DVGM frameworks to improve performance in applications like medical imaging, predictive maintenance, and personalized content generation, where model robustness and adaptability are critical.

Moreover, this thesis demonstrates how DVGM architectures can be scaled for large-model design and adapted for specific downstream tasks. The findings offer a practical roadmap for deploying DVGMs in complex real-world scenarios, emphasizing the models' flexibility, scalability, and effectiveness in supporting a broad spectrum of machine learning applications.

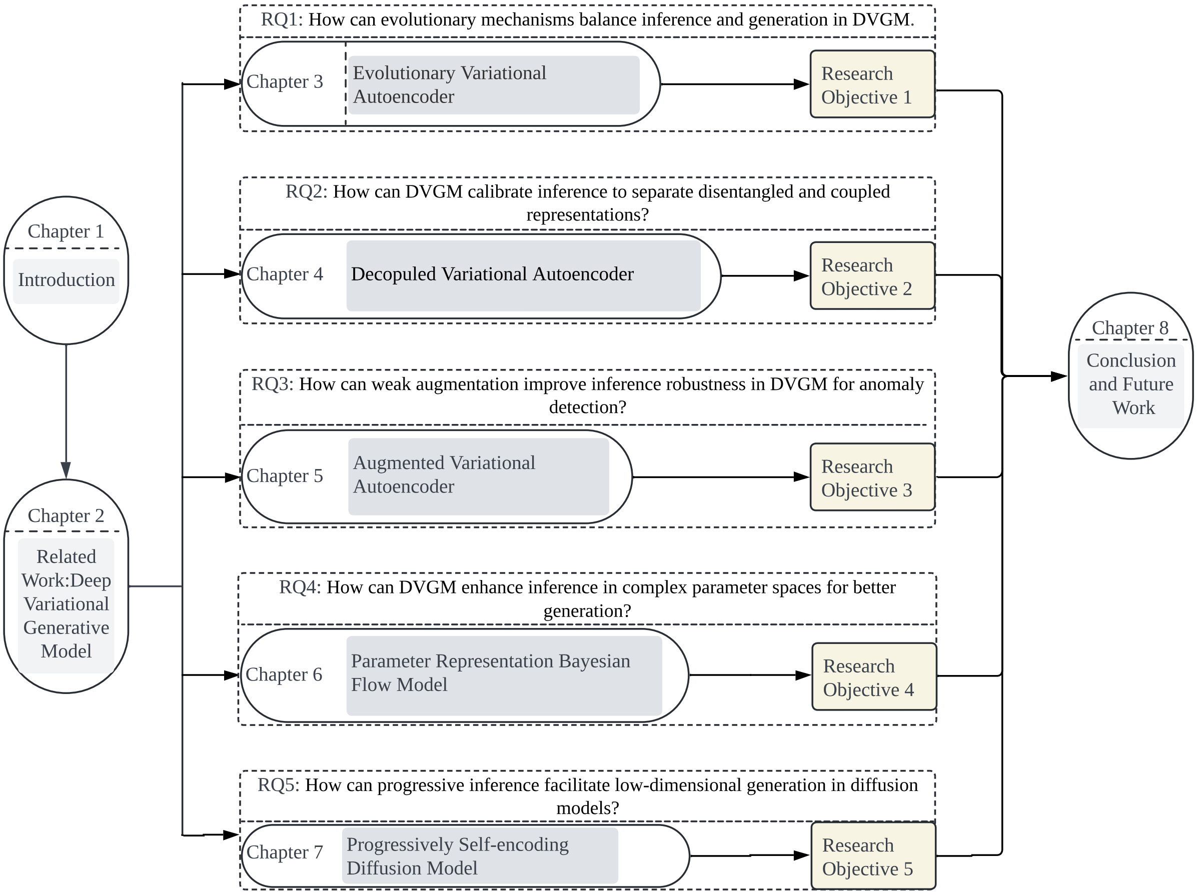

RQ1: How can evolutionary mechanisms balance inference and generation in DVGM.
Chapter 3
Evolutionary Variational Autoencoder
Research Objective 1
RQ2: How can DVGM calibrate inference to separate disentangled and coupled representations?
Chapter 4
Decopuled Variational Autoencoder
Research Objective 2
RQ3: How can weak augmentation improve inference robustness in DVGM for anomaly detection?
Chapter 5
Augmented Variational Autoencoder
Research Objective 3
RQ4: How can DVGM enhance inference in complex parameter spaces for better generation?
Chapter 6
Parameter Representation Bayesian Flow Model
Research Objective 4
RQ5: How can progressive inference facilitate low-dimensional generation in diffusion models?
Chapter 7
Progressively Self-encoding Diffusion Model
Research Objective 5
Chapter 1
Introduction
Chapter 2
Related Work:Deep Variational Generative Model
Chapter 8
Conclusion and Future Work

Figure 1.1: Thesis structure

1.7 Thesis Structure

The structure of this thesis is shown in Figure 1.1, and the chapters are organized as follows:

CHAPTER 1 provides an introduction to the research background, motivation, research questions, objectives, and contributions of this thesis.

CHAPTER 2 reviews related work on Deep Variational Generative Models (DVGM), discussing existing methodologies, applications, and limitations, thereby establishing the foundation for the contributions in this thesis.

CHAPTER 3 presents eVAE: Evolutionary Variational Autoencoder, a novel DVGM that applies evolutionary mechanisms to dynamically balance inference and generation. This chapter addresses **RQ1** and aims to achieve **RO1** by enhancing model adaptability without the need for static hyperparameter tuning.

CHAPTER 4 introduces C2VAE: Gaussian Copula-based VAE, which differentiates disentangled from coupled representations through a contrastive posterior. This model is designed to improve representation stability in complex image data, addressing **RQ2** to achieve **RO2**.

CHAPTER 5 presents the Weakly Augmented Variational Autoencoder (WAVAE) for anomaly detection in time series data. By integrating self-supervised learning, WAVAE strengthens inference robustness and improves sensitivity to anomalies. This chapter addresses **RQ3** to achieve **RO3**.

CHAPTER 6 introduces ParamReL: Learning Parameter Space Representation via Progressively Encoding Bayesian Flow Networks. This model is designed to enhance inference across complex parameter spaces, supporting DVGM applications in both discrete and continuous data domains. This chapter addresses **RQ4** to achieve **RO4**.

CHAPTER 7 proposes ProgDiffusion: Progressively Self-encoding Diffusion Models, a novel DVGM that facilitates low-dimensional and timestep-specific semantic generation. This approach enhances DVGM efficiency and aligns latent representations with diffusion processes, addressing **RQ5** to achieve **RO5**.

CHAPTER 8 summarizes this thesis's findings, discusses the research's implications, and suggests potential directions for future work.

1.8 Notation and Description of Symbols

In the development of Deep Variational Generative Models (DVGM), a clear and consistent notation system is essential for understanding the mathematical formulations and conceptual elements. **Table 1.1** provides a comprehensive summary of the symbols and their corresponding descriptions used throughout this work. The symbols are categorized into six main groups:

- **Losses and Divergences**: This category includes various loss functions ($\mathscr{L}$) and divergence measures ($\alpha^{\mathtt{D}},\beta^{\mathtt{D}},\gamma^{\mathtt{D}}$) that are critical for optimizing the DVGM framework. These metrics help balance the reconstruction accuracy, latent representation quality, and model robustness.
- **Data and Latent Variables**: These symbols ($\mathbf{x},\mathbf{z}$, etc.) represent the input data, its variations (e.g., $\mathbf{x}_t,\mathbf{z}_{\mathrm{a}}$), and corresponding latent variables. They are foundational for describing how data is processed and represented within the generative model.
- **Statistical Parameters and Functions**: This group (μ,σ,ρ, etc.) defines statistical properties such as mean, variance, and correlation, which are used for modeling distributions and quantifying uncertainty in the latent space.
- **Distributions and Functions**: Key probability distributions ($p_{\mathtt{I}},q,T$) and mathematical functions (Ψ,TC) are included here, highlighting their role in variational inference and generative processes.
- **Evolutionary Parameters and Strategies**: This section introduces evolutionary computation concepts ($\mathtt{chromosomes}$,$\mathscr{E}$,Pr_c,Pr_m), which are integrated with the variational generative framework. These parameters and strategies enhance the exploration of solution spaces and the model's adaptability to complex tasks.
- **Miscellaneous Symbols**: These symbols (N,B,f,g, etc.) capture additional concepts, including data batch properties, dimensionality, and specific encoding or decoding functions. They ensure a complete and versatile representation of the DVGM process.

This detailed breakdown of notations facilitates a structured understanding of the theoretical framework and its practical implementations. By standardizing these symbols, the presented model maintains clarity and consistency, enabling easier comprehension and replication of the proposed methods.

Table 1.1: Symbols and their descriptions in DVGM.

Notions	**Descriptions**
Losses and Divergences	

Continued on next page

Notions		Descriptions	
$\mathscr{L}$	Loss function	$\mathscr{L}_{\mathrm{R}}$	Reconstruction loss
$\mathscr{L}^{\mathrm{r}}$	Raw loss	$\mathscr{L}^{\mathrm{a}}$	Augmented loss
$\mathscr{L}_{\mathtt{I}}$	Inference loss	α	Coefficient of total correlation
$\alpha^{\mathtt{D}}$	α-divergence	β	Coefficient of KL divergence
$\beta(t)$	Noise schedule	$\beta^{\mathtt{D}}$	β-divergence
γ	Coefficient of mutual information	$\gamma^{\mathtt{D}}$	γ-divergence
Data and Latent Variables			
$\mathbf{x}$	Data/input variable	$\hat{\mathbf{x}}$	Reconstructed data/input variable
$\mathbf{x}_t$	t-times data/input variable	$\mathbf{y}$	Labels of sample
$\mathbf{z}$	Latent variable	$\mathbf{z}_{\mathrm{r}}$	Raw latent variable
$\mathbf{z}_{\mathtt{a}}$	Augmented latent variable	$\mathbf{z}_{\mathrm{t}}$	t-times latent variable
$\mathbf{z}_{\mathtt{c}}$	Coupled representation	$\mathbf{z}_i, \mathbf{z}_+^i$	Positive sample representations
$\mathbf{z}_-^{ij}$	Negative sample representation		
Statistical Parameters and Functions			
μ	Mean value	$\mu_{\mathtt{c}}$	Coupled mean variable
σ	Variance value	$\sigma_{\mathtt{c}}$	Coupled variance value
Σ	Covariance matrix	ρ	Correlation matrix
ν	Degree of freedom	τ	Temperature parameter
Distributions and Functions			
$p_{\mathtt{I}}$	Input distribution	$p_{\mathtt{O}}$	Output distribution
q	Posterior distribution	Ψ	Classifier
I	Mutual information	TC	Total correlation
T	Time step	T^{c}	Cumulative distribution function
Evolutionary Parameters and Strategies			
$\texttt{chromosome}$	Chromosome variables	$\{\beta_l\}$	Beta coefficients
c	Information bottleneck	$\mathscr{E}$	Variational evolutionary learner
$f^{\mathtt{Fit}}$	Fitness function	Pr_c	Crossover rate
Pr_m	Mutation rate	$\mathscr{R}$	Evolving distribution

Continued on next page

Notions		**Descriptions**	
$\mathscr{M}$	Variational mutation strategy		
Miscellaneous Symbols			
N	Number of samples	B	Batch samples
K	Number of negative samples	D	Dimensions of input
f	Encoder function	g	Decoder function
F	Multivariate cumulative distribution	F^{-1}	Inverse of cumulative distribution
V	Value function	C	Copula function
r	Density ratio	h	Bayesian update function
u_t	Hyper feature at t-times	H	Entropy
ϵ	Noise variable $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$	η	Threshold value of anomaly score
φ	Noise model parameterized by φ	$\eta^{\texttt{eVAE}}$	Hyperparameter of SBX

CHAPTER 2

DEEP VARIATIONAL GENERATIVE MODEL

Chapter Overview. This chapter introduces the foundations and advancements of Deep Variational Generative Models (DVGM), a family of models that combine the strengths of variational inference and deep learning. The chapter is structured as follows:

- Section 2.1 provides a high-level overview of DVGMs and their representative architectures, including VAEs, diffusion models, and Bayesian Flow Networks.
- Subsequent sections examine key technical components: trade-offs between inference and generation, representation learning, latent space modeling, and cross-modality adaptability.
- We also review the application of DVGMs in time-series modeling, anomaly detection, and self-supervised learning.
- The chapter concludes with a discussion on research gaps in current DVGM approaches, motivating the novel methods proposed in the following chapters.

2.1 Deep Variational Generative Model Overview

Generative AI has rapidly emerged as a transformative area in machine learning, providing frameworks for data generation and representation that mimic real-world complexity [53, 84, 106, 115, 136]. Within this domain, Deep Variational Learning (DVL) [54, 62, 76] represents a powerful integration of deep learning and variational inference techniques. This fusion benefits from deep networks' representational power and the ability of variational inference to handle uncertainty. DVL thus enables models like the Variational Autoencoder (VAE) [120, 130, 145, 146], Flow-based Models [16, 122], and Diffusion Models [39, 123, 131] to model complex distributions. These models can be applied across diverse tasks, from image generation to anomaly detection, thanks to their flexibility and robustness.

2.1.1 Variational Autoencoders

The VAE is a foundational model in DVGM, designed to address limitations in autoencoders such as the lack of smooth representation spaces [140]. Through learning continuous and smooth distribution representations $p(x)$ over latent variables $\boldsymbol{z}$, VAEs enable effective encoding and decoding that preserves data structure. The encoder, parameterized by $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$, and the decoder, which approximates $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$, allow VAEs to reconstruct data and generate new samples by sampling from the learned distribution. This framework, leveraging the Stochastic Gradient Variational Bayes (SGVB) estimator and the reparameterization trick, enables efficient optimization of both generative and inference parameters, leading to tractable gradients. The VAE objective function, typically optimized via ELBO, balances data reconstruction and KL divergence, making it suitable for generative tasks where capturing latent structure is critical [45, 52].

VAEs mitigate autoencoder issues like sparse representation [140] by learning continuous and smooth representation distribution $p(x)$, $x \in \mathscr{X}$ from observations $\mathscr{X}$ over latent variables $\boldsymbol{z}$. After learning an encoding distribution $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ in encoding neural networks, VAEs apply variational inference to approximate the posterior distribution $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$. Learning tasks such as reconstructed and generated outputs can then be sampled from this learned distribution in a generative process. With the SGVB estimator and reparameterization trick, the gradients become tractable, and the generative parameters θ and inference parameters ϕ are learnable. The objectives of VAEs can be converted to ELBO with the expectation over empirical distribution p_{data} of the data towards both reconstruction $\mathscr{L}_{\text{R}}$ and inference $\mathscr{L}_{\text{I}}$ [45, 52].

$$\begin{aligned}\mathscr{L}_{ELBO} &= E_{x \sim p_{\text{data}}} \Big[E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} [\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})] \\ &\quad - \mathrm{KL}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \| p(\boldsymbol{z})\right)\Big] \\ &= E_{x \sim p_{\text{data}}} \mathscr{L}_{\text{R}} + \mathscr{L}_{\text{I}}\end{aligned} \tag{2.1}$$

2.1.2 Diffusion Models

Diffusion models are a recent addition to the DVGM family, excelling in generating high-quality, diverse data by modeling data transformations as a sequence of denoising steps [39, 123, 131]. These models leverage stochastic processes to iteratively transform noise into data, resulting in clear, sharp images or structured data outputs. Unlike other generative models, diffusion models are particularly suitable for tasks requiring high fidelity in generated samples, such as image generation and speech synthesis. By optimizing for different noise levels and introducing regularization techniques, diffusion models achieve both flexibility and precision, handling high-dimensional data effectively.
DDPM Denoising Diffusion Probabilistic Models (DDPM) [62] introduce diffusion and denoising processes to generate high-quality samples. Formally, let $\mathbf{x}_0$ denote the original data and $\mathbf{x}_T$ denote the pure Gaussian noise. The forward diffusion process is defined as a series of diffusion steps $t \in \{1, \ldots, T\}$, where noise is added at each diffusion step t:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{2.2}$$

β_t is a variance schedule that controls the amount of noise added at each step. The goal of the diffusion process is to gradually transform data $\mathbf{x}_0$ into noise $\mathbf{x}_T$ through this iterative procedure.

The reverse process aims to revert the noisy data $\mathbf{x}_T$ back to the original data $\mathbf{x}_0$ by step-by-step denoising. This is achieved by learning a series of denoising steps that approximate the reverse transitions:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}), \tag{2.3}$$

where $\mu_{\boldsymbol{\theta}}(\cdot,\cdot)$ is a neural network parameterized by $\boldsymbol{\theta}$ to predict the mean of the reverse process, and σ_t^2 is the variance. The entire reverse process can be optimized by minimizing the KL-divergence of $\mathrm{KL}[q(\mathbf{x}_0, \mathbf{x}_{1:T}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x}_0, \mathbf{x}_{1:T})]$. This objective encourages the model to learn accurate reverse transitions, thereby enabling the generation of high-quality samples from pure noises.

DDIM [131] accelerates sampling process of DDPM by designing a deterministic sampling step as:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}}\frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}, \mathbf{0}), \tag{2.4}$$

which keeps the form $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I})$. Since Markov property is not required in the diffusion process, DDIM may use a subset of denoising timesteps to speed up the sampling procedure.

Diffusion autoencoders (DAE) Diffusion models are incapable of obtaining meaningful semantic representations through its training process since the intermediate latents $\{\mathbf{x}_t\}_t$ have the same size of observations $\mathbf{x}_0$. Thus, diffusion autoencoder (DAE) methods [118, 150] use an encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_0)$ to first encode observations $\mathbf{x}_0$ into a semantic representation $\mathbf{z}$ and then insert $\mathbf{z}$ into the denoising step of the reverse process as $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z})$. The entire reverse process can be optimized by minimizing the KL-divergence of $\mathrm{KL}\left[q(\mathbf{x}_0, \mathbf{x}_{1:T}|\mathbf{z})q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_0) \parallel p_{\boldsymbol{\theta}}(\mathbf{x}_0, \mathbf{x}_{1:T})p(\mathbf{z})\right]$.

However, $\mathbf{z}$ captures *static* semantic representations but not align well with the dynamic intermediate latents $\{\mathbf{x}_t\}_t$. More importantly, it would be almost infeasible to utilize the trained encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_0)$ to generate samples since sample $\mathbf{x}_0$ is unknown in the sample generation task.

2.1.3 Bayesian Flow Networks

Bayesian Flow Networks enhance flow-based models by integrating Bayesian principles to model data uncertainties and structural dependencies [54]. In these networks, the transformation between data distributions is governed by a series of invertible mappings, allowing for precise control over data representations. Bayesian Flow Networks are especially useful in fields where the data exhibits complex, structured dependencies, such as in genomics or time-series forecasting. The Bayesian component allows the model to capture both the uncertainty inherent in the data and the latent dependencies between variables, providing an interpretable framework that can adapt to dynamic data structures.

Bayesian Flow Networks (BFNs) [54, 135, 171] serve as deep generative models with a primary objective to learn an output distribution for generating observations. The distribution's parameters are learned by a neural network, which takes the posterior parameters of observations of inputs. Here, we try to understand BFNs from an alternative parameter perspective since these (posterior) parameters play a key role in BFNs. BFNs involves concepts such as input distribution, sender distribution and receiver distribution, making it less accessible to readers unfamiliar with BFNs.

Figure 2.1 shows T steps of training and sample generation in BFNs, similar to diffusion models [62, 131]. To train BFNs, we minimize the divergence between the ground-truth data distribution and the evolving output distributions over T steps. At each step $t \in \{T, \ldots, 1\}$, an intermediate (posterior) parameter $\boldsymbol{\theta}_t$ is first updated using a Bayesian update function $h(\cdot)$ as $\boldsymbol{\theta}_t = h(\boldsymbol{\theta}_{t+1}, \mathbf{x}_{t+1})$, where $\mathbf{x}_{t+1}$ is the observation at step $t+1$. $\boldsymbol{\theta}_t$ is then fed into a neural network $\psi(\cdot)$ to form the parameters of output distribution, i.e., a decoder $p_{\mathrm{O}}(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t)$, for model training. After training, these intermediate output distributions can be employed to simulate observations during the sample generation process, replacing the actual observations at each step t.

By working in the parameter space, BFNs can uniformly model continuous, discrete, and discretized observations. For example, BFNs can use the mean of Gaussian distributions as parameter $\boldsymbol{\theta}$ to model continuous data or use the event probabilities of categorical distributions as $\boldsymbol{\theta}$ to study discrete data (see detailed settings for distributions in Table 2.1). However, BFNs cannot produce meaningful latent semantics capturing high-level concepts in the mixed-type observations, such as hair colors in portrait images.

Table 2.1: Examples of detailed distribution formats in BFNs. $\boldsymbol{\theta}_{t+1} = \{\mu_{t+1}, \rho_{t+1}^{-1}\})$. cate: categorical distribution.

Data type	$p_{\mathrm{I}}(\mathbf{x}_t|\boldsymbol{\theta}_{t+1})$	$p_{\mathrm{S}}(\widehat{x}_t|\mathbf{x}_t;\alpha_t)$	$\boldsymbol{\theta}_t = h(\boldsymbol{\theta}_{t+1}, \widehat{x}_t, \alpha_t)$
Continuous data	$\mathcal{N}(\mathbf{x}_t;\mu_{t+1},\rho_{t+1}^{-1})$	$\mathcal{N}(\widehat{x}_t;\mathbf{x},\alpha_t^{-1})$	$\mu_t = \frac{\alpha_t\widehat{x}_t+\rho_{t+1}\mu_{t+1}}{\alpha_t+\rho_{t+1}}$
Discrete data	$\mathrm{Cat}(\mathbf{x}_t;\frac{1}{K}\cdot\mathbf{1})$	$\mathcal{N}(\widehat{x}_t;\alpha_t K\mathbf{e}_{\mathbf{x}_t}-\alpha_t,\alpha_t K\mathbf{I})$	$\boldsymbol{\theta}_t = \frac{e^{\widehat{x}_t}\boldsymbol{\theta}_{t+1}}{\sum_k e^{\mathbf{x}_{t-1,k}}\theta_{t+1,k}}$
Data type	$p_{\mathrm{O}}(\mathbf{x}_t|\boldsymbol{\theta}_t)$	$p_{\mathrm{R}}(\widehat{x}_t|\psi(\boldsymbol{\theta}_t),\alpha_t)$	
Continuous data	$\delta(\mathbf{x}_t-\psi(\boldsymbol{\theta}_t))$	$\mathcal{N}(\widehat{x}_t;\psi(\boldsymbol{\theta}_t),\alpha_t^{-1})$	
Discrete data	$\mathrm{Cat}(\mathrm{softmax}(\psi(\boldsymbol{\theta}_t)))$	$\sum_k p_O(k;\psi(\boldsymbol{\theta}_t))\mathcal{N}(\widehat{x}_t;\alpha_t K\mathbf{e}_k-\alpha_t,\alpha_t K\mathbf{I})$	

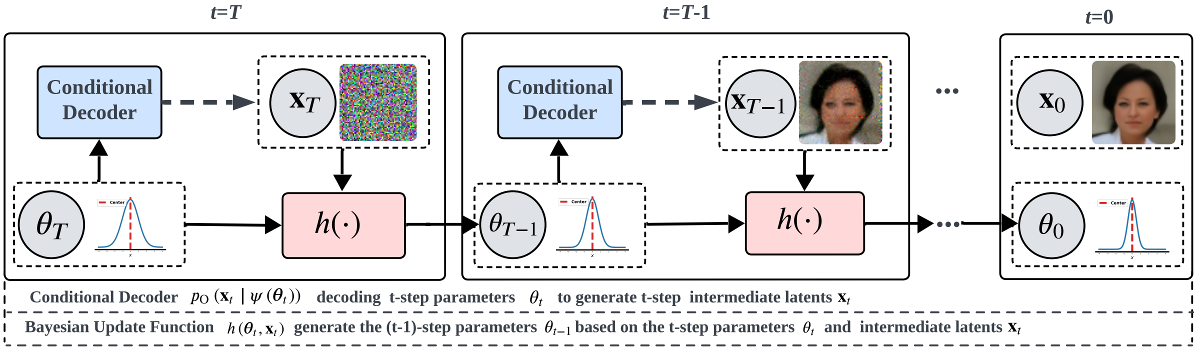

t=T
t=T-1
t=0
Conditional Decoder
$\mathbf{x}_T$
θ_T
$h(\cdot)$
Conditional Decoder
$\mathbf{x}_{T-1}$
θ_{T-1}
$h(\cdot)$
$\mathbf{x}_0$
θ_0
Center
Conditional Decoder $p_{\mathrm{O}}(\mathbf{x}_t \mid \psi(\theta_t))$ decoding t-step parameters θ_t to generate t-step intermediate latents $\mathbf{x}_t$
Bayesian Update Function $h(\theta_t, \mathbf{x}_t)$ generate the (t-1)-step parameters θ_{t-1} based on the t-step parameters θ_t and intermediate latents $\mathbf{x}_t$

Figure 2.1: Our alternative understanding of Bayesian Flow Networks (BFNs). Each step consists of a conditional decoder $p_{\mathrm{O}}(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t))$ (in blue rectangle) and a Bayesian update function $h(\cdot)$ (in peach rectangle). In training BFNs, the dashed arrows between the conditional decoder and $\{\mathbf{x}_t\}_{t=1}^T$ are non-existent, as $\{\mathbf{x}_t\}_{t=1}^T$ refers to observations. During sample generation, these dashed arrows become solid, representing that the decoder generates $\mathbf{x}_t$ as part of the sampling process.

2.2 Applications of DVGM on Multimodal Data

The DVGM framework is highly adaptable to various data modalities, making it a versatile tool for modeling structured and unstructured data. By capturing complex latent spaces and managing uncertainty, DVGM enables sophisticated representations across diverse applications.

2.2.1 Trade-off between Generation and Inference

To solve Evidence Lower Bound (ELBO), the inference model $q_\phi(z|x)$ can be trained jointly by maximizing the ELBO to acquire reasonable compression for task fitting. However, a weak capacity of the decoder and the variety of data could make the expressive posterior favor task fitting rather than optimal inference [188]. For example, in variational language generation, the decoder built on autoregressive models such as LSTM and PixelCNN can generate language samples by the autoregressive property rather than the posterior-based latent variables [157]. The VAE degenerates to an autoregressive model where the KL divergence between posterior and prior reaches zero quickly during training. This results in KL vanishing and poor generalization in test for the lack of diversity. The approaches of learning orthogonal transformation of priors with the same distribution by the decoder [71, 107] may sacrifice accurate inference in optimal representation and generalization of fitting the data. Other research attributes the training conflict to the inherent property of bound optimization. For example, under a solid factorial assumption about the posterior distribution [98], i.e.,

$$p(\mathbf{x},\mathbf{z}) = p(\mathbf{x} \mid \mathbf{z}) \prod_i p(z_i), \tag{2.5}$$

the ELBO constraining the variational samples favors the data fitting [18] but fails to maximize the probability mass on log-likelihood. In addition, the vanilla VAE optimizer strengthens the disjointness between $q_\phi(z|x_i)$, i.e., $\mu_i \to \infty, \sigma_i \to 0^+$, to separate the log-likelihood concentrated on each sample, resulting in maximizing the mass of joint distribution [188].

One way to address the above issues is to tighten the log-likelihood lower bound for correct variational approximation in posterior [7, 40, 81, 124, 148], prior [77, 141] and decomposition of ELBO [45, 188] under some mild assumptions. For example, β-VAE adds the hyper-parameter β to weigh the $\mathscr{L}_{KL}$ term. Then, ELBO minimizes $\mathscr{L}_R$ to the convergence of data fitting collectively with the regularization $\mathscr{L}_I$ by varying β:

$$\begin{aligned}\mathscr{L}_{ELBO} =& E_{x\sim p_{data}}\Big[E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}\mid\mathbf{z})] \\ & -\beta D_{KL}\left(q_\phi(\mathbf{z}\mid\mathbf{x})\|p(\mathbf{z})\right)\Big] \\ =& E_{x\sim p_{\text{data}}}\left[\mathscr{L}_{\text{R}}+\beta\mathscr{L}_{\text{I}}\right].\end{aligned} \tag{2.6}$$

β-VAE introduces some fundamental limitations, which trigger various follow-up research. InfoVAE introduces a scaling parameter λ on the KL-term and converts the objective to [188]:

$$\begin{aligned}\mathscr{L}_{ELBO} =& \alpha I_q(\mathbf{x};\mathbf{z})-D_{KL}\left(q_\phi(\mathbf{z}\mid\mathbf{x})\|p(\mathbf{z})\right) \\ & -E_{q(z)}\left[D_{KL}\left(q_\phi(\mathbf{z}\mid\mathbf{x})\|p_\theta(\mathbf{x}\|\mathbf{z})\right)\right],\end{aligned} \tag{2.7}$$

where I_q is the mutual information with weight α and $\alpha+\lambda-1=0$. This linear tuning on the KL shows limitation to dynamic uncertainty.

Further, conditional VAE (CVAE) introduces an initial guess as a conditional variable into the objective function for multimodal data. The SA-VAE involves a cyclical annealing schedule to split the training to multiple cycles starting at $\beta = 0$ and progressively increases β until $\beta = 1$ to reduce the KL vanishing [49]. In ControlVAE, the PID control compares the KL divergence with a set point, with their difference as feedback to the controller to tune the hyperparameter $\beta(t)$ [127]. ControlVAE thus optimizes KL dynamically but is constrained by the PID controller which follows a separate tuning mechanism from the VAE itself.

The existing work leaves gaps for building an approximate weight allocation between reconstruction and inference, tuning external hypberparameters within the VAE working mechanism and handling these issues in a dynamic manner over an evolutionary learning process. Our eVAE addresses these gaps by incorporating the variational genetic learning into balancing inference and generation and evolutionarily involving their effect into adjusting the VAE learning behaviors toward better uncertain tradeoff learning.

2.2.2 Representation Learning

DVGM provides a robust framework for representation learning by capturing complex data structures through latent space transformations. By training on disentangled representations and leveraging contrastive methods, DVGM learns independent and semantically meaningful features, making it ideal for applications such as image classification, object detection, and semantic segmentation [15]. Techniques like β-VAE, InfoVAE, and FactorVAE add constraints that encourage disentanglement, allowing the DVGM to isolate independent factors within the data. This disentanglement enables more interpretable features, which are crucial in applications where feature interpretability is essential for downstream tasks.

Unsupervised disentangled learning in VAEs aims to learn hierarchical distribution dependencies between hidden features toward inducing hidden units independently discriminative to generative factor variances, thus capturing those explanatory features in the hidden space [15]. This requires meeting a factorizable and diagonal assumption on estimating posterior distributions in VAEs [18, 76] to generate decoupled features by stochastic variational inference. To eliminate the entanglement between hidden features, the TC and dual total correlation (DTC) are incorporated into evidence lower bound (ELBO) under the factorization assumption. Specifically, penalizing the TC and DTC terms aims to regularize the posterior estimation toward discarding those dependent feature pairs or clusters, respectively. Accordingly, the recent research focuses on accurately estimating these TC terms. For example, β-TCVAE [27] derives a decomposed ELBO by the Monte Carlo (MC) estimation iteratively over samples. HFVAE [45] constructs an MC-based estimator by partially stratified sampling. These methods suffer from the MC-based scalability issue and inductive bias (such as relating to the batch size). Further, FactorVAE [74] involves an adversarial mechanism to train a density ratio-based ELBO. GCAE [178] captures dependencies in feature groups by specifying discriminators on specific DTC terms. In contrast, C^{2}VAE involves a new attempt for disentangled learning to differ disentangled from coupled features and then their representations.

Contrastive learning enables self-supervision. One typical example is to contrast similar with dissimilar data points by a triplet loss to encode and discriminate semantic features in a hypothesis space for representation learning [57]. Another recent topic is to train conditional generative models in a contrastive manner to exploit the correlations between data samples, which could be of various types. cVAE [1] learns a foreground reconstruction by eliminating the background information among dependent feature pairs. C-VAE [35] learns a latent variable indicator by a minority/majority loss to address the class imbalance in downstream tasks. ContrastVAE [151, 163] aggregates the posterior from two different views of comments for a sequential recommendation. NCP-VAE [9] trains an optimal prior for sampling with a contrastive loss in an adversarial way. These studies focus on reconstruction for specific learning tasks, but limited work contributes to inference accuracy in VAEs. In contrast, C^{2}VAE makes the first attempt to learn and differ strongly vs weakly coupled features for contrastive disentangled-coupled representation disentanglement.

Copula functions [168] are introduced to deep neural networks for VDL, including VAEs, variational LSTM (VLSTM) [165], where copula learns the dependencies between hidden features. Copula-based VAEs and VLSTM integrate copula dependence modeling into variational inference to improve autoencoders and LSTMs. CopulaVAE [148] replaces the collapsible ELBO with a Gaussian copula-based posterior to avoid the KL vanishing in language modeling. Copula VLSTM [165] learns dependence degrees and structures between hidden features for leveraging LSTM for sequential forecasting. [149] adopts a Gaussian copula to model the correlations between discrete latent variables for a conditional generation from a Bernoulli posterior. [125, 155] integrate a copula function into LSTM to model the dependence for forecasting. Instead, C^2VAE integrates copula representations into contrastive classification to downplay those coupled features for improved disentangled representation learning.

2.2.3 Time-Series Learning

Time-series data present unique challenges due to their temporal dependencies and structural patterns. In this context, DVGM models such as Variational Recurrent Neural Networks (VRNN) and time-series-based VAEs capture sequential dependencies by integrating meta-prior learning and recurrent structures [34, 137]. These models enable anomaly detection, trend forecasting, and behavior prediction by capturing the latent distributions over time, which is crucial for robust time-series modeling. Recent extensions incorporate recurrent and attention-based structures, enhancing DVGM's ability to model long-term dependencies and contextual variability, which are essential for applications in finance, healthcare, and climate modeling.

2.2.4 Implicit Data Fitting by Non-probabilistic Generative Models

Non-probabilistic generative models for Time Series Anomaly Detection (TSAD) aim to reconstruct data robustly. Prior studies concentrate on optimizing this reconstruction process to match the characteristics of time-series data via the design of deep network embeddings. Specifically, [25] and [73] implemented an autoencoder (AE) framework, deploying symmetric encoder-decoder structures and assembling one to multiple CNN-based encoder-decoders for the reconstruction of sequence data. Furthermore, [190] utilized a generative adversarial network (GAN)-based reconstruction for anomaly detection, implicitly fitting a likelihood function for normal data through an adversarial mechanism.

2.2.5 Explicit Data Fitting by Probabilistic Generative Models

Unlike AE-based models that learn an encoding-decoding process for a dataset, VAE-based models excel in identifying continuous representations within a low-dimensional space. These representations, characterized by their smooth and continuous nature in the hidden space, are essential for preserving probabilistic properties during sampling. Consequently, VAEs can reconstruct samples with increased sharpness and interpretability, outperforming their AE-based counterparts. In contrast to GANs, VAEs explicitly model data likelihood distribution and provide additional constraints on the posterior distribution based on a preset prior, making them more suitable for modelling data in dynamic areas and designing end-to-end anomaly detectors. For instance, [93] utilized a Gaussian mixture model (GMM) as data likelihood distribution, and [164] employs a dynamical prior over time.

Issues in VAE based TSAD: VAEs tend to sacrifice representation [187] for data fitting. In that case, the induced latent hole will lead to the lack of robustness. At the same time, the modeling failure in learning the likelihood of the sequence data exacerbates its robustness issues. Specifically, VAE-based anomaly detection typically employs a convolutional neural network (CNN) architecture for data encoding. While effective for image data, this approach often fails to capture the temporal characteristics of time-series data, such as seasonality, periodicity, and frequency domain features, through CNN encoding filters. The shallow fully connected networks (FCN) are employed in VAEs as substitutes. As a result, the naive structure cannot capture varying dependencies, and compared to image data, the sequences in training are relatively small. Due to the modeling and data issues, these generative models cannot converge to optimal.

Advances in VAE-based TSAD: To remedy the above issues, the traditional variational framework has been upsurged by integrating the *meta-prior* into generative modelling. For instance, a variational recurrent neural network (VRNN) establishes a model for the VAE inference, prior estimation, and reconstruction processes by capturing the temporal dependencies between intermediate variable $\boldsymbol{h}$ in the deterministic model and input variables $\boldsymbol{x}$ in the recurrent neural network. This approach and its variants [34, 72, 137] effectively utilize VAE to learn and model the latent distribution of data while maintaining temporal dependencies in the recurrent neural network. On the other hand, the variational representation can be designed. [89] utilizes prototype-based approaches to define latent representations for multivariate time series (MTS) and learn a robust likelihood distribution of normal data.

2.2.6 Self-supervised Learning on Time Series Data in Deterministic and Generative Models

In deterministic models, augmenting time-series data or their representations, combined with specific self-supervised algorithms, can provide a sufficient depth for training in downstream tasks. For instance, in prediction tasks, [156] encoded time-series segments in both time and frequency domains to obtain positive and negative sample pairs, using contrastive learning to capture the seasonal-trend representation of time-series data. [64] constructed positive pairs with multi-granularity time-series segments and corresponding latent variable representations, enhancing fine-grained information for prediction by maximizing mutual information.

In classification tasks, [86] formed pairwise representations of global and local input series, obtaining informational gains through adversarial learning. For anomaly detection tasks, [169] acquired spatio-temporal dependent representations suitable for downstream tasks. [180] proposed a multi-layer representation learning framework to obtain consistent, contextual representations of overlapping segments, designing a contrastive loss by decomposing overlapping subsequences in both instance and temporal dimensions to obtain positive and negative sample pairs. Additionally, [173] employed a dual bilinear process at the encoding level to capture positive and negative samples of time sequences, thereby capturing both long and short-term dependencies.

In contrast, Self-Supervised Learning (SSL) based on time-series generative models typically focuses on data and representation augmentation as a generative approach. For instance, AE-based methods such as those presented in [29, 129, 152, 184] leverage the AE architecture for data augmentation. Similarly, diffusion-based approaches [6, 90, 154] employ diffusion processes to augment time-series data and representations.

2.2.7 Latent Space Learning

DVGM models provide flexible latent space structures, which can be adapted to capture hierarchical, structured, and disentangled representations for both continuous and discrete data. Hierarchical VAEs, for example, allow for the learning of nested latent spaces, supporting applications where complex dependencies exist, such as in natural language processing and hierarchical clustering. Through architectures like VQ-VAE and Ladder VAE, DVGM can capture varying levels of abstraction within the data, which is essential for tasks requiring multi-level feature extraction, such as language modeling, speech synthesis, and bioinformatics [75, 130].

Generative representation learning models can be considered as a variant of Latent Variable Models (LVMs). Latent Variable Models [46] which aim at learning the joint distribution $p(\mathbf{x},\mathbf{z})$ over data $\mathbf{x}$ and latent variables $\mathbf{z}$ present efficient ways for uncovering hidden semantics. In LVMs, the joint distribution $p(\mathbf{x},\mathbf{z})$ is usually decomposed as: $p(\mathbf{x},\mathbf{z}) = p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})$, where $p(\mathbf{z})$ represents prior knowledge for inference [143], thus facilitating learning the conditional distribution $p(\mathbf{x} \mid \mathbf{z})$. Among LVMs, Variational AutoEncoders (VAEs) [76] and diffusion models [62, 131] are two representative approaches [82].

In VAEs, latent variables $\mathbf{z}$ is obtained through an *encoder network* $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, whereas observations are reconstructed through a *decoder network* $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, with $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ being the encoder and decoder parameters.

The dimensions of $\mathbf{z}$ are usually much smaller than those of $\mathbf{x}$, denoted as $|\mathbf{z}| \ll |\mathbf{x}|$, such that redundant information is effectively removed and the most semantically meaningful factors are abstracted [100]. VAEs are popular for downstream tasks like disentanglement [44, 61, 66, 174], classification [138, 142], and clustering [68, 167].

On the other hand, diffusion models [62, 131] first use T diffusion steps to transform observation $\mathbf{x}$ into a white noise $\mathbf{x}_T$ and then use T denoising steps to reconstruct the observation. Diffusion models have obtained impressive performance in the fidelity and diversity of generation tasks. However, they might be unable to obtain meaningful latent semantics since the dimensions of $\mathbf{x}$ and $\mathbf{x}_T$ are the same as $|\mathbf{x}| = |\mathbf{x}_T|$. [118, 150] have attempted to integrate a decodable auxiliary variable $\mathbf{z}$ to enable diffusion models to obtain low-dimensional latent semantics. However, they have not overcome issues like the slow training speed inherent to the diffusion and reverse processes.

2.2.8 Discrete and Continuous Data Modeling

DVGM models have shown effectiveness in handling mixed data types, such as discrete and continuous data. Recent advances incorporate copula-based modeling and discrete encodings, which facilitate learning in non-smooth latent spaces. For discrete data, models like VQ-VAE employ codebook structures to encode discrete representations, supporting applications in text generation and symbolic reasoning. For continuous data, hierarchical structures like NVAE and ParamReL improve posterior approximation and enhance generative capacity [130, 148]. This adaptability allows DVGM to handle applications in natural language processing, medical diagnosis, and other domains where both discrete and continuous variables coexist.

Recent advances have demonstrated that diffusion models [62, 131] are capable of generating high-quality data. Nonetheless, compared to the autoencoder framework, the intermediate outputs in diffusion stages are high-dimensional and lack smoothness, making them unsuitable for representation learning. Contemporary research focuses on encoding a conditional latent space to acquire low-dimensional semantic representations. However, those observations-based models [118, 150], such as VAEs and diffusion models, exhibit limitations when applied to discrete data.

Deep hierarchical VAEs have seen progress in capturing latent dependence structures for encoding an expressive posterior, statistically or semantically. VQVAE-based [120, 146] models have local-to-global features-based explanatory hierarchies at the image level, forming a codebook-based discrete posterior. In [130, 141], recursive latent structures in multi-layer networks form an aggregated posterior. NVAE [145] demonstrates that depth-wise hierarchies encoded by residual networks can approximate the posterior precisely despite using shallow networks. Unlike the observation-based encoder, where the information flow between input and latent is maximized in encoding-decoding pipelines in the sample space, ParamReL uses progressive encoders in the parameter space to capture the dynamic semantics.

Pre-trained diffusion models [123], [14] have shown that the upsampling features from a U-Net can capture semantic information useful for downstream tasks. This discovery has sparked increasing research in leveraging these upsampling features of pre-trained diffusion models across various applications, including classification [108, 159], semantic segmentation [14, 189], panoptic segmentation [166], semantic correspondence [59, 101, 139, 182], and image editing [60, 144]. In most of these approaches, identifying the optimal denoising step and upsampling layer is crucial for achieving high predictive performance. These approaches do not suggest fundamental changes to model architectures or training methodologies, leaving the specific architectural components and techniques for learning useful semantic representations unclear. ParamReL uses these discoveries to construct efficient self-encoders.

Chapter Summary

This chapter provided a comprehensive overview of Deep Variational Generative Models, covering foundational architectures such as VAEs, diffusion models, and Bayesian Flow

Networks, as well as their applications across diverse data types and tasks. Through our analysis, several key limitations were identified:

- Existing models often struggle to maintain a stable trade-off between inference accuracy and generative expressiveness, especially in dynamic or high-dimensional settings.
- Representation learning remains constrained by challenges in disentanglement, information collapse, and the lack of semantic interpretability in intermediate latents.
- Current approaches inadequately support joint modeling of discrete and continuous variables, limiting DVGM adaptability in real-world, heterogeneous data environments.
- Most architectures are optimized for specific tasks or modalities, lacking a generalizable and modular framework to unify design across domains.

These challenges motivate the development of novel, inference-centered DVGM models introduced in the next chapters, which aim to improve adaptability, robustness, and semantic expressiveness in both generative and downstream tasks.

Chapter 3

Evolutionary Variational Autoencoder

The Variational Autoencoder (VAE) framework provides a powerful approach for learning latent representations, but challenges such as KL-vanishing and imbalanced inference-generation dynamics limit its performance. To address these issues, we propose evolutionary variants, aiming to answer **RQ1**: *How can evolutionary mechanisms balance inference and generation in DVGM?*

To achieve **RO1**, we introduce the **Evolutionary VAE (eVAE)**, which employs variational genetic algorithms to dynamically optimize the information bottleneck, mitigating KL-vanishing and enhancing representation disentanglement. The following sections detail the design of eVAE, highlighting how evolutionary strategies, such as probabilistic chromosome selection and simulated binary crossover, are integrated to achieve a balance between compression and reconstruction.

3.1 eVAE: Evolutionary variational autoencoder

Variational Autoencoders (VAEs) [76] have attracted considerable interest for their capacity to learn continuous and smooth distributions from observations, integrating probabilistic modeling and deep neural learning principles. They offer substantial advantages in incorporating prior knowledge, mapping inputs to probabilistic representations, and approximating the likelihood of outputs. The incorporation of a Stochastic Gradient Variational Bayes (SGVB) estimator [76] within VAEs allows the model to learn a structured probabilistic latent space for more representative attributes in a hidden space. VAEs have been successfully applied in various fields, including time series forecasting [47], out-of-domain detection in images [58, 109, 119, 162], image generation with spiking signals, and text generation by language modeling [185]. Beyond generative tasks, VAEs are used extensively in representation learning tasks, such as disentanglement [126, 176], classification [69, 128], clustering [172], and manifold learning [2, 31]. However, VAEs still encounter challenges, particularly in finding an optimal trade-off between representation compression and generation accuracy.

Theoretically, the bound optimization of variational inference in VAEs replaces the log-likelihood function with a surrogate function optimized by gradient descent. In practice, the Evidence Lower Bound (ELBO) is unable to fully approximate the conditional likelihood due to a persistent gap between posterior and prior distributions, thereby failing to achieve a balance between representation robustness and reconstruction quality. Specifically, a weak KL divergence can lead to KL vanishing, whereas a strong KL divergence may result in an unfavorable likelihood. Additionally, this trade-off is sensitive to the disjointed nature of posterior distributions, data characteristics, and network architectures.

To address these issues, various techniques have been proposed. The first approach focuses on adjusting term balance in objective functions. Examples include β-VAEs [20] incorporating a hyperparameter β, InfoVAE [188] adding a scaling parameter to the KL term, SA-VAE [49] using a cyclical annealing schedule to progressively increase β to reduce KL vanishing, and ControlVAE [127] implementing a proportional-integral-derivative (PID) control to dynamically tune hyperparameters. However, these methods partially address only specific objectives, failing to resolve balance issues in dynamic settings.

From an information bottleneck perspective, VAEs function as lossy information compressors, where adjusting the KL divergence within a range controls the information bottleneck, which flows from representing latent variables to reconstructing samples. This adjustment supports a trade-off between compression and reconstruction [8, 20]. Inspired by this perspective, we propose a novel framework called *evolutionary VAE* (eVAE), which dynamically tunes the optimal state of this information bottleneck across iterations to better align information flow. eVAE integrates variational evolutionary learning with variational information bottleneck concepts in VAEs to facilitate optimal exploration and a well-balanced trade-off between representation compression and generation accuracy.

The integration of evolutionary learning with VAEs is an emerging research area.

This work pioneers this integration by proposing a *variational genetic algorithm* that optimizes VAE objectives and model exploration through an evolving framework. Additionally, eVAE dynamically refines the VAE inference process (the KL terms) using evolutionary and probabilistic techniques, promoting a stable convergence towards balanced inference and generation. To prevent premature convergence, eVAE introduces probabilistic chromosome selection for a smooth search space. To avoid exhaustive random search, simulated binary crossover [36] and Cauchy-distributed mutation guide training towards stable convergence. Overall, eVAE represents the first evolutionary VAE framework, unconstrained by VAE architecture, input settings, or ELBO objective function. By combining variational encoding and decoding, information bottleneck principles, and evolutionary learning within a deep neural framework, eVAE improves model disentanglement and reconstruction loss, effectively addressing the issue of KL vanishing.

3.2 The eVAE Model

The eVAE model addresses the challenges mentioned by incorporating variational genetic learning to balance inference and generation, adjusting VAE learning dynamics towards a better trade-off without complex hyperparameter tuning. The eVAE model also tunes the inference-generation balance in Eq. (2.1) by jointly addressing issues in Eqs. (2.6) and (2.7). The framework of eVAE, illustrated in Figure. 3.1, integrates variational evolutionary learning into deep learning to improve VAE balance.

3.2.1 Notation and Problem Setup

The input variable is denoted by $\mathbf{x} \in \mathscr{X} \subseteq \mathbb{R}^{d_{\mathbf{x}}}$, and the latent variable by $\mathbf{z} \in \mathscr{Z} \subseteq \mathbb{R}^{d_{\mathbf{z}}}$ where $d_{\mathbf{x}} \gg d_{\mathbf{z}}$. The goal is to achieve optimal inference capability, i.e., deriving $\mathbf{z}$ for downstream tasks, and robust reconstruction, i.e., generating $\hat{\mathbf{x}}$ from $\mathbf{x}$ using the decoder. During each iteration, the VAE optimizes a surrogate loss in Eq. (2.6) regulated by a factor β. We adopt an evolutionary algorithm-based training method termed *evolving inner-outer-joint training*, where the value of β is determined iteratively to balance inference and reconstruction.

At the start of training, a set of candidate values, $\{\beta\}$, representing l chromosomes, is generated. For iteration t, model performance is evaluated by its distortion $\mathscr{L}_I$ and rate $\mathscr{L}_R$. These values are then passed to the variational evolutionary learner $\mathscr{E}$. Operated

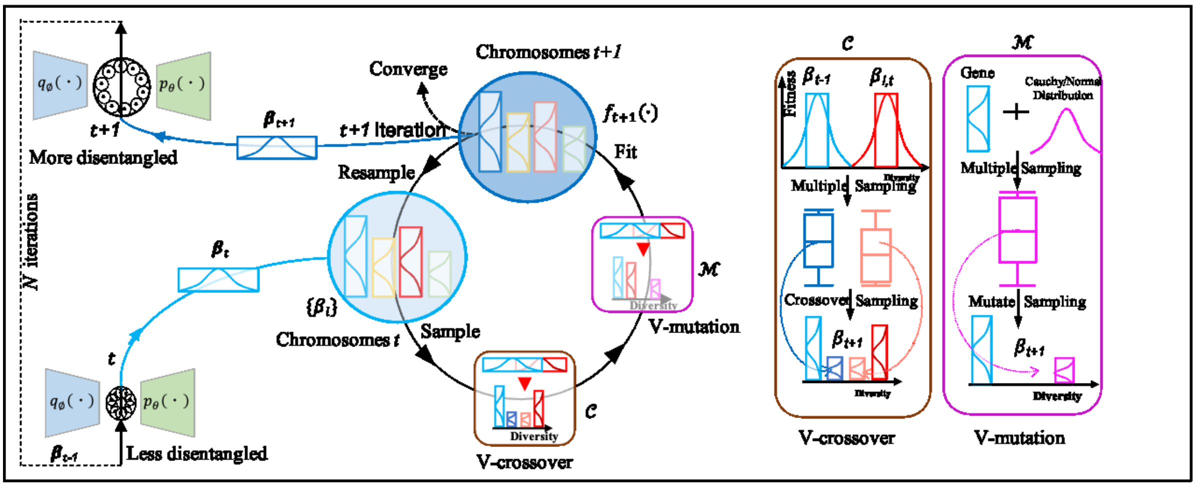


Figure 3.1: The framework of eVAE. The VAE results inform chromosome sampling. The genes are then updated through variational V-crossover and V-mutation. The evolved results at t are evaluated for retraining, abandonment, or convergence at $t+1$.

under crossover and mutation probabilities, denoted Pr_c and Pr_m, the learner performs genetic operations on chromosomes. These adjusted genes are integrated into the model for the subsequent iteration, guiding the model toward a trade-off between inference and reconstruction.

3.2.2 eVAE - Evolving Inner-outer-joint Training

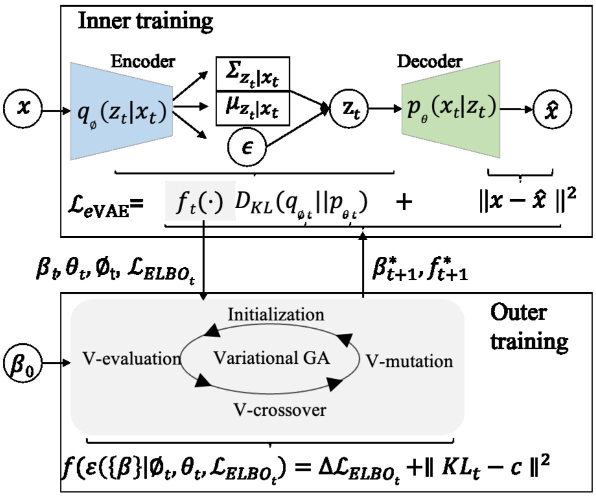

Inner training
Encoder
x
$q_\phi(z_t|x_t)$
$\Sigma_{z_t|x_t}$
$\mu_{z_t|x_t}$
ϵ
z_t
Decoder
$p_\theta(x_t|z_t)$
$\hat{x}$
$\mathcal{L}_{eVAE} = f_t(\cdot)\ D_{KL}(q_{\phi_t}||p_{\theta_t}) + ||x-\hat{x}||^2$
$\beta_t, \theta_t, \phi_t, \mathcal{L}_{ELBO_t}$
β^*_{t+1}, f^*_{t+1}
β_0
Initialization
V-evaluation
Variational GA
V-mutation
V-crossover
Outer training
$f(\varepsilon(\{\beta\}|\phi_t, \theta_t, \mathcal{L}_{ELBO_t}) = \Delta\mathcal{L}_{ELBO_t} + || KL_t - c ||^2$

Figure 3.2: eVAE - inner-outer joint evolutionary training process. The upper part illustrates VAE training at time t, while the lower part shows outer training by variational genetic algorithm. Optimized results are fed back to VAE for further training.

The eVAE framework follows an evolving inner-outer-joint learning process, where both VAE and its evolutionary parameters are optimized iteratively. Figure. 3.2 shows this process. For input $\mathbf{x}_t \in \mathscr{X}$, a VAE model $\mathscr{V}$ is initialized with β_0 and trained to optimize the decoder $p_\theta(\mathbf{x}_t|\mathbf{z}_t)$ and encoder $q_\phi(\mathbf{z}_t|\mathbf{x}_t)$ with a prior $p(\mathbf{z})$. The initial objective function is defined as:

$$\mathscr{L}_{ELBO_t^{inner}} = E_{\mathbf{x}_t \sim p_{\mathscr{X}}} \left[\mathscr{L}_{\mathrm{R}}(\mathbf{x}_t, \mathbf{z}_t) + \beta \mathscr{L}_{\mathrm{I}}(\mathbf{x}_t, \mathbf{z}_t)\right].$$

In the next step, an outer variational evolutionary learner $\mathscr{E}$ updates the weight in an outer process. In this work, $\mathscr{E}$ is implemented using a *variational genetic algorithm* (VGA) for variational evolution. $\mathscr{E}$ samples a chromosome β_t from an evolving distribution $\mathscr{R}$ and evolves it through variational crossover and mutation, generating a new chromosome β_{t+1} for the next outer iteration.

The updated parameter β_{t+1} is used in the VAE model $\mathscr{V}_t$ for the following training iteration. We store the VAE state at time t and calculate the fitness value of β_{t+1} using the following objective:

$$\begin{aligned} \mathscr{L}_{ELBO_{t+1}^{\text{outer}}} =& E_{\mathbf{x}_{t+1} \sim p_{\mathscr{X}}} \big[\mathscr{L}_{\mathrm{R}}(\mathbf{x}_{t+1}, \mathbf{z}_{t+1}) \\ & + \beta \mathscr{L}_{\mathrm{I}}(\mathbf{x}_{t+1}, \mathbf{z}_{t+1})\big]. \end{aligned} \tag{3.1}$$

Next, the state of the VAE at time t is reloaded, and β_{t+1} is updated by the chromosome with the highest fitness value, β^*_{t+1}, from the candidate group. Consequently, $\mathscr{V}_{t+1}$ and its parameters ϕ_{t+1}, θ_{t+1}, and β_{t+1} are updated for the $t+1$ iteration. The VGA $\mathscr{E}$ repeats this process until convergence.

This joint inner-outer training approach optimizes the balance between reconstruction and inference (e.g., minimizing $\mathscr{L}_R$ with reconstructed $\hat{\mathbf{x}}_t$ and optimizing $\mathscr{L}_I$) over time, iterating in a generative process until the VAE model converges.

Thus, the eVAE objective function becomes:

$$\mathscr{L}_{eVAE} = \min_{\theta_t, \phi_t} \textstyle\sum_{t=1}^{N} \mathscr{L}_{ELBO_{t+1}} \left(\theta_t, \phi_t; f\left(\mathscr{E}(\phi_t, \theta_t, \{\beta\}, \mathscr{L}_{ELBO_t})\right), \mathbf{x}_t\right). \tag{3.2}$$

Optimizing β^* for the balance between representation and generation yields:

$$\beta^* \sim f\left(\mathscr{E}\left(\{\beta\} \mid \phi_t, \theta_t, \mathscr{L}_{ELBO_t}\right)\right). \tag{3.3}$$

The overall ELBO of eVAE is:

$$\begin{aligned} \mathscr{L}_{eVAE} &= E_{\mathbf{z} \sim q_\phi(\mathbf{z}_t|\mathbf{x}_t)} \log p_\theta(\mathbf{x}_t \mid \mathbf{z}_t) + f_{\beta_t \sim \mathscr{R}}(\beta_t, \mathscr{E}) D_{KL}(q_\phi(\mathbf{z}_t \mid \mathbf{x}_t) \| p(\mathbf{z})) \\ &= E_{\mathbf{x}_{t+1} \sim p_{\mathscr{X}}}[\mathscr{L}_R + f(\beta)\mathscr{L}_I + \mathscr{E}(\Delta \mathscr{L}_{ELBO})]. \end{aligned} \tag{3.4}$$

3.2.3 Variational Evolution in eVAE

The *variational evolutionary learner* $\mathscr{E}$ in eVAE optimizes parameters using a Variational Genetic Algorithm (VGA), including crossover and mutation operations. This approach circumvents typical problems of premature convergence and random search.

3.2.3.1 Variational Genetic Algorithm (VGA)

eVAE uses VGA for external parameter optimization while conducting internal gradient-based optimization. VGA consists of several stages: initialization VGA consists of several stages: initialization, variational crossover (V-crossover), variational mutation (V-mutation), and variational evaluation (V-evaluation), as shown in the bottom part of Figure. 3.2. Each step is outlined below.

Chromosome selection: Chromosomes in VGA are embedded as continuous variables sampled from an evolving distribution $\mathscr{R}$. A candidate group of L chromosomes, $\{\beta_l\} = \{\beta_1, \ldots, \beta_L\}$, is maintained, where each chromosome is associated with a fitness value, $f^{\texttt{Fit}}$. This allows chromosome-fitness pairs to evolve in VGA, producing offspring selected based on fitness across generations:

$$\{\beta, f^{\texttt{Fit}}\} = \{(\beta_1, f_1^{\texttt{Fit}}), \ldots, (\beta_L, f_L^{\texttt{Fit}})\}. \tag{3.5}$$

V-crossover: The top-performing chromosomes at time t undergo crossover to generate new genes, increasing genetic variety. Using Simulated Binary Crossover (SBX) [36], eVAE identifies the strongest chromosomes, $\beta_{t,father}$ and $\beta_{t,mother}$, and generates candidate offspring, β_{t+1}, for the next iteration:

$$\mathscr{C}: \begin{cases} \beta_{t+1,child1} = \frac{1}{2}[(1+r_c)\beta_{t,father} + (1-r_c)\beta_{t,mother}]; \\ \qquad\qquad \text{or} \\ \beta_{t+1,child2} = \frac{1}{2}[(1-r_c)\beta_{t,father} + (1+r_c)\beta_{t,mother}], \end{cases} \tag{3.6}$$

where r_c is the crossover rate, drawn from a probability density function $P_c(r_c)$:

$$P_c(r_c) = \begin{cases} 0.5(\eta^{\texttt{eVAE}}+1) r_c^{\eta^{\texttt{eVAE}}}, & \text{if } r_c \le 1; \\ 0.5(\eta^{\texttt{eVAE}}+1) \frac{1}{r_c^{\eta^{\texttt{eVAE}}+2}}, & \text{otherwise.} \end{cases} \tag{3.7}$$

Samples for r_c are drawn from:

$$r_c = \begin{cases} (2u)^{\frac{1}{\eta^{\texttt{eVAE}}+1}}, & \text{if } u \le 0.5; \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta^{\texttt{eVAE}}+1}}, & \text{otherwise} \end{cases} \tag{3.8}$$

where u is a random variable, and $\eta^{\texttt{eVAE}}$ is a hyper-parameter that influences the similarity between offspring and their parents. A larger $\eta^{\texttt{eVAE}}$ yields offspring closer to parents, while a smaller $\eta^{\texttt{eVAE}}$ results in greater variation. The new offspring, β_{t+1}, is selected based on which option of $\mathscr{C}$ better satisfies Eq. equation 3.11.

The V-crossover procedure is outlined in Algorithm 1.

Algorithm 1 `VGACrossover`: Simulated Binary Crossover in Variational Genetic Algorithm

```
Require: Strongest chromosomes at $t$-iteration, $\beta_{t,father}$, $\beta_{t,mother}$; scaling hyperparameter $\eta^{\texttt{eVAE}}$
Ensure: Updated candidate group $\{\beta_l\}^{t+1}$ at $t+1$-iteration
Generate a random variable $u$ between 0 and 1
if $u \leq 0.5$ then
    $r_c = (2u)^{\frac{1}{\eta^{\texttt{eVAE}}+1}}$                ▷ Generate mutation rate $r_c$
else
    $r_c = (\frac{1}{2(1-u)})^{\frac{1}{\eta^{\texttt{eVAE}}+1}}$
end if
function SBXCROSSOVER($\beta_{t,father}, \beta_{t,mother}, r_c$)
$\beta_{t+1,child1} = 0.5\left[(1+r_c)\beta_{t,father} + (1-r_c)\beta_{t,mother}\right]$
$\beta_{t+1,child2} = 0.5\left[(1-r_c)\beta_{t,father} + (1+r_c)\beta_{t,mother}\right]$
Replace the two chromosomes with lowest fitness in $\{\beta_l\}^t$ with $\beta_{t+1,child1}$ and $\beta_{t+1,child2}$
end function
return $\{\beta_l\}^{t+1}$
```

V-mutation: Offspring generated through crossover undergo further mutation to enhance genetic diversity. Variational mutation strategy $\mathscr{M}$ diversifies offspring by modifying β_{t+1} from crossover or β_t from the current generation. Chromosome $\beta_{t,m}$ from the group mutates with probability:

$$\mathscr{M} : \beta_{t+1} = \beta_{t,m} + r_m, \tag{3.9}$$

where r_m is sampled from a Cauchy distribution P_m:

$$P_m = \frac{1}{\pi}\frac{1}{1+\beta_{t,m}^2}. \tag{3.10}$$

Algorithm 2 details the mutation process.

3.2.3.2 V-evaluation & VGA Fitness Function

Chromosomes updated through V-crossover and V-mutation undergo evaluation to determine if they advance to the next generation. Within the VAE framework, the following

Algorithm 2 `VGAMutation`: Cauchy-based Mutation in Variational Genetic Algorithm

```
Require: Selected chromosome $\beta_{t,m}$ at $t$-iteration from group $\{\beta_l\}^t$
Ensure: Updated group $\{\beta_l\}^{t+1}$ at $t+1$-iteration
function CauchyMutation($\beta_{t,m}$)
    Generate a Cauchy distribution based on $\beta_{t,m}$
    $P_m = \frac{1}{\pi}\frac{1}{1+\beta_{t,m}^2}$
    Sample from $P_m$
    $\beta_{t+1} = \beta_{t,m} + r_m$        ▷ Generate mutated chromosome and update the group
end function
return $\{\beta_l\}^{t+1}$
```

heuristic fitness function $f^{\texttt{Fit}}$ guides chromosome evolution to align with VAE objectives:

$$f_{t+1}^{\texttt{Fit}} = \Delta\mathscr{L}_{ELBO_{t+1}} + ||KL_{t+1}(\beta_{t+1}) - c||, \tag{3.11}$$

where:

$$\Delta\mathscr{L}_{ELBO_{t+1}} = \mathscr{L}_{ELBO_{t+1}}(\beta_{t+1}) - \mathscr{L}_{ELBO_t}(\beta_t), \tag{3.12}$$

represents the ELBO change after applying the evolved β. The task-specific information bottleneck c ensures bound optimization of eVAE through evolutionary adjustments.

V-evaluation and fitness values direct eVAE towards convergence, balancing reconstruction and inference with evolutionary parameterization. Chromosomes with higher fitness, forming pairs such as $\{\beta_{t+1}^*, f_{t+1}^*\}$, proceed in the optimization. The inner-outer-joint eVAE process is summarized in Algorithm 3.

3.3 Theoretical Analysis

To illustrate the eVAE framework, we consider β-VAE as a base model to develop an evolutionary VAE. Here, we analyze the impact of eVAE on parameter adjustment, the trade-off between reconstruction and regularization, and the training performance in β-VAE.

To understand how eVAE achieves effective compression in encoding and decoding, we revisit ELBO from an information bottleneck perspective. Using rate-distortion theory, we define an optimization bound for β-VAE. This leads to three distinct lower bounds: experiment-specific (e.g., β-VAE), KL-specific (e.g., ControlVAE), and iteration-specific (e.g., eVAE). The experiment-specific bound adjusts the ratio β_B between reconstruction and representation loss as a fixed hyperparameter [20]. KL-specific approaches, such

Algorithm 3 `EIOTraining`: Evolving Inner-outer-joint Training

```
Require: Crossover rate $Pr_c$, mutation rate $Pr_m$, batch data $\mathbf{x}$
Ensure: Optimal chromosome pair $\{\beta^*, f^*\}$
Initialize parameters of decoder $\theta$, encoder $\phi$, and chromosome group $\{\beta_l\} = \{\beta_1, \dots, \beta_L\}$
while $t < T$ do
    Sample $Pr_t$ from $N(0,1)$                                ▷ Probability to evolve
    if $Pr_t \leq Pr_m$ then
        Save current parameters $\theta_t$, $\phi_t$
        Generate $\beta_{t+1}$ using V-mutation $\mathscr{M}(\beta_t)$
        Evaluate $\beta_{t+1}$ using fitness function $f(\mathscr{E}(\phi_t, \theta_t, \{\beta\}, \mathscr{L}_{ELBO_t}))$
    else if $Pr_m < Pr_t \leq Pr_c$ then
        Save parameters $\theta_t$, $\phi_t$
        Generate $\beta_{t+1}$ using V-crossover $\mathscr{C}(\beta_{t,father}, \beta_{t,mother})$
        Evaluate $\beta_{t+1}$ using fitness function
    else
        Select the strongest pair $\{\beta^*, f^*\}$
        Update VAE parameters $\theta_{t+1}, \phi_{t+1} \leftarrow \mathscr{L}_{eVAE}(\theta_t, \phi_t, \beta^* | \mathbf{x}_t)$
    end if
end while
return $\{\beta^*, f^*\}$
```

as PID-based ControlVAE [127], employ nonlinear controllers for KL divergence to set β_{KL} dynamically. In contrast, eVAE aims to dynamically adjust β through a variational evolutionary learner $\mathscr{E}(\beta_t)$ over iterations.

The trade-offs achieved by eVAE across iterations balance reconstruction and representation, optimizing β-VAE parameters and achieving both theoretical and empirical efficiency, as shown in Figure. 3.3. In particular, the early convergence of disentangled representation impacts reconstruction, while unstable rate optimization impacts disentanglement [126]. Only the eVAE model, guided by an iteration-specific lower bound, maintains a stable balance, achieving optimal models with low reconstruction error and high mutual information gap (MIG) metric.

Specifically, β-VAE tunes β by setting a task-relevant ratio between inference loss $\mathscr{L}_{\mathrm{I}}$ and generation loss $\mathscr{L}_{\mathrm{R}}$, aiming to maximize the likelihood at a constant threshold, B, given by:

$$\max_{\phi,\theta} E_{\mathbf{x}\sim p_{\text{data}}} \mathscr{L}_R, \quad \text{s.t. } \beta \mathscr{L}_I < B. \tag{3.13}$$

This process can be framed through the Information Bottleneck (IB) theory [7], where $\beta\mathscr{L}_I$ acts as a bottleneck, compressing latent capacity of $\mathbf{Z}$ to represent $\hat{\mathbf{X}}$. In this context,

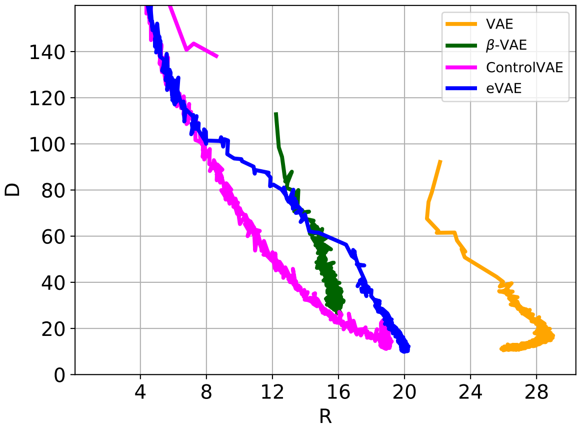


Figure 3.3: Information plane with $R-D$ curves of VAE, β-VAE, ControlVAE, and eVAE on dSprites. Values (R,D) are averaged over five restarts per iteration.

IB optimizes $I(\mathbf{Z},\hat{\mathbf{X}})$ to derive a concise representation:

$$\max I(\mathbf{Z},\hat{\mathbf{X}}), \quad \text{s.t. } I(\mathbf{Z},\hat{\mathbf{X}}) \leq I_B. \tag{3.14}$$

Theorem 3.1. *For β-VAE hyperparameter β_B, the experiment-specific lower bound is:*

$$\mathcal{L}_{\beta-VAE}(\theta,\phi) = -D - \beta_B R \leq -\beta_B I - D,$$

where R denotes compression rate, D distortion, and β_B is an experiment-specific constant. VAEs therefore achieve a rate-distortion trade-off.

Proof. Using VIB, ELBO is bounded by distortion $D = -E_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})]$ and rate $R = E_{q_\phi(\mathbf{x},\mathbf{z})}[D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))]$, where $I(\mathbf{X},\mathbf{Z})$ is bounded above by $I \leq R$:

$$\begin{aligned} R &= \int q(\mathbf{x})\mathrm{d}\mathbf{x} \int q_\theta(\mathbf{z} \mid \mathbf{x}) \log \frac{q_\theta(\mathbf{z} \mid \mathbf{x})}{p(\mathbf{z})} \mathrm{d}\mathbf{z} \\ &= I(\mathbf{X},\mathbf{Z}) + TC(\mathbf{Z}) \\ &\geq I(\mathbf{X},\mathbf{Z}) \end{aligned}$$

and lower bound $I \geq H - D$:

$$\begin{aligned} I &= \int\int p(\mathbf{x},\mathbf{z}) \log \frac{p(\mathbf{x} \mid \mathbf{z})}{p(\mathbf{x})} \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} \\ &\geq \int\int p(\mathbf{x},\mathbf{z}) \log \frac{q(\mathbf{x} \mid \mathbf{z})}{p(\mathbf{x})} \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} \\ &= \int\int p(\mathbf{x},\mathbf{z}) \log q(\mathbf{x} \mid \mathbf{z}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} - \int p(\mathbf{x}) \log p(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &= \int\int p(\mathbf{x},\mathbf{z}) \log q(\mathbf{x} \mid \mathbf{z}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} + H(\mathbf{X}) \\ &= H - D. \end{aligned}$$

Thus, the information bottleneck is bounded by

$$H - D \leq I \leq R,$$

where H represents data entropy.

In β-VAE, β_B imposes a compression constraint, producing:

$$\begin{aligned}
&H - D \leq I \leq R,\\
&H - D \leq \beta_B I \leq \beta_B R,\\
&H \leq \beta_B I + D \leq \beta_B R + D.
\end{aligned}$$

The bound simplifies to:

$$\mathscr{L}_{\beta-VAE}(\theta,\phi) = -D - \beta_B R \leq -\beta_B I - D.$$

■

ControlVAE is a β-VAE variant tuning β_{KL} to achieve a specific KL level through PID control:

Theorem 3.2. *For ControlVAE's KL-specific factor β_{KL}, the lower bound is:*

$$\mathscr{L}_{ControlVAE}(\theta,\phi) = -D - \beta_{KL} R \leq -\beta_{KL} I - D.$$

eVAE extends the β-VAE framework by evolving $\beta I(\mathbf{X},\mathbf{Z})$ through a variational evolutionary learner $\mathscr{E}$. Following outer iteration t, β_t evolves based on task and training phase, guided by fitness function f^{Fit}:

Theorem 3.3. *eVAE establishes an iterative lower bound for the inner training phase at iteration $t+1$:*

$$\mathscr{L}_{eVAE} \leq -\mathscr{E}(\beta_t) I_t - \mathscr{E}(\beta_t) D_t,$$

where the iteration-specific lower bound balances task alignment in $-\mathscr{E}(\beta_t)D_t$ with inference quality in $\mathscr{E}(\beta_t)I_t$.

Proof. From Eq. equation 3.4 and Eq. equation 3.11, the eVAE loss at iteration t is:

$$\begin{aligned}
\mathscr{L}_{eVAE} &= E_{\mathbf{z}\sim q_\phi(\mathbf{z}_t|\mathbf{x}_t)} \log p_\theta(\mathbf{x}_t \mid \mathbf{z}_t) +\\
&\quad f_{\beta_t \sim \mathscr{R}}(\beta_t, \mathscr{E}) D_{KL}(q_\phi(\mathbf{z}_t \mid \mathbf{x}_t) \| p(\mathbf{z}))\\
&= E_{\mathbf{x}_{t+1}\sim p_{\mathscr{X}}}[\mathscr{L}_R + f(\beta_t)\mathscr{L}_I + \mathscr{E}(\Delta \mathscr{L}_{ELBO})].
\end{aligned}$$

Per VIB theory, R_t sets the upper bound of the information bottleneck $I_t(\mathbf{X}_t, \mathbf{Z}_t)$:

$$R_t = I_t(\mathbf{X}_t, \mathbf{Z}_t) + TC(\mathbf{Z}_t).$$

The lower bound D_t is given by:

$$I_t \geq H - D_t.$$

Thus, eVAE optimization for a dynamic lower bound is:

$$\mathscr{L}_{eVAE} \leq -\mathscr{E}(\beta_t)D_t - \mathscr{E}(\beta_t)I_t.$$

■

We further compare the effects of eVAE against baseline VAE models (β-VAE and ControlVAE) using rate-distortion (R-D) curves in Figure. 3.3. Standard VAE (in yellow) prioritizes representation learning over minimizing empirical error at the initial stage, optimizing the rate only after a quarter of the iterations. In β-VAE (in green), the experiment-specific lower bound $(-\beta_B I - D)$ restricts the achievable distortion. ControlVAE (in pink) initially minimizes distortion for data fitting, and subsequently optimizes the rate to obtain a smooth representation through a KL-specific bound $(-\beta_{KL} I - D)$. However, directly adjusting β through the PID controller can cause fluctuating trade-offs between inference capacity and reconstruction quality. In contrast, eVAE produces an iteration-specific lower bound $(-\mathscr{E}(\beta_t)I_t - \mathscr{E}(\beta_t)D_t)$ to maintain a stable balance between minimizing distortion and controlling the rate. In disentangled representation learning, early convergence hinders reconstruction, while unstable R-D optimization reduces disentanglement [126]. Only the iteration-specific lower bound from eVAE avoids balance fluctuation, reaching optimal model performance with low reconstruction error and high mutual information gap (MIG) metric.

3.4 Experiments

We evaluate eVAE in three primary tasks: disentangled representation learning, image generation, and language modeling. Each task is set up following established baselines [127], ensuring consistency in encoder-decoder network architectures and optimizers.

3.4.1 Dataset and Baselines

The three tasks and their respective datasets are outlined below:

Disentangled representation learning aims to learn independent latent variables to generate images. We evaluate eVAE on the dSprites dataset[1], a collection of 737,280 binary images generated by five factors: shape (square, ellipse, heart), scale, orientation, and x and y positions. Baselines include β-VAE, ControlVAE, and DynamicVAE, with parameters kept consistent across models. For fairness, we use $\beta = 1$ and $\beta = 4$ in β-VAE, and for ControlVAE and DynamicVAE, a setpoint of KL = 19.

Image generation involves reconstructing imagery samples from given data points. For this task, we use the CelebA dataset (cropped version) [97], which consists of 202,599 RGB images of celebrity faces. The baselines used for comparison are β-VAE and ControlVAE.

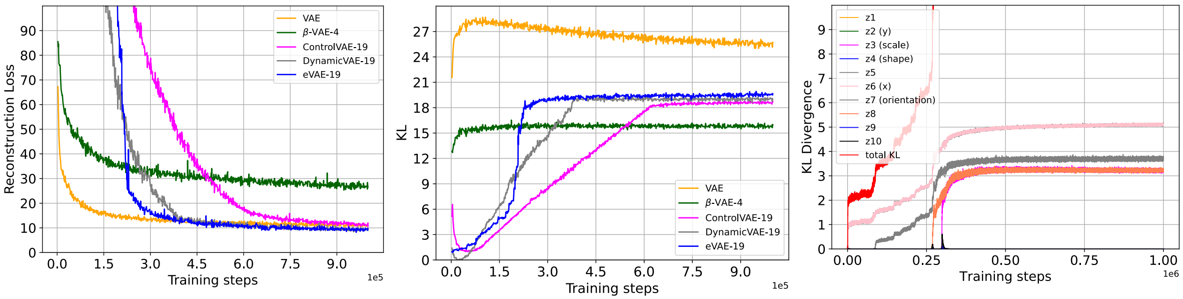
Language modeling performs word-level text generation. We use the Penn Treebank dataset (PTB) [103], an English corpus. To illustrate the impact of KL vanishing, we compare eVAE with cost annealing (Cost-10k) [17], cyclical annealing (Cyc-8) [49], and PID control (PID-3) [127]. For fair comparison, eVAE is tuned to match the KL setpoint used in ControlVAE.

3.4.2 Performance in Disentangled Representation Learning

In this task, we evaluate reconstruction quality using disentangled features. Figure. 3.4(a) shows that eVAE achieves the lowest reconstruction loss, reaching 9.2, compared to ControlVAE at the same KL divergence target (KL = 19). Figures. 3.4(a) and (b) also demonstrate that eVAE presents a more stable training curve than other VAEs due to its VGA fitness-guided dynamic weighting, unlike the direct β adjustments in β-VAE and ControlVAE. Figure. 3.4(c) illustrates the disentanglement achieved across each generative factor.

The disentanglement effectiveness is further quantified using the MIG [26] and dimension-wise MIG scores. Table 3.1 summarizes the results, where eVAE outperforms β-VAE and other baselines.

[1] dSprites: a dataset for disentanglement testing. Available at: https://github.com/deepmind/dsprites-dataset/



(a) Reconstruction Loss

(b) KL

(c) Element-wise KL Divergence

Figure 3.4: Learning curves on dSprites. (a, b) show that eVAE achieves the lowest reconstruction loss compared to VAE ($\beta = 1$), β-VAE ($\beta = 4$), ControlVAE, and DynamicVAE at a fixed KL point KL = 19. (c) displays the element-wise KL divergence across iterations, with eVAE maintaining stable KL values for each dimension (factor): y position (z2), scale (z3), shape (z4), x position (z6), and orientation (z7).

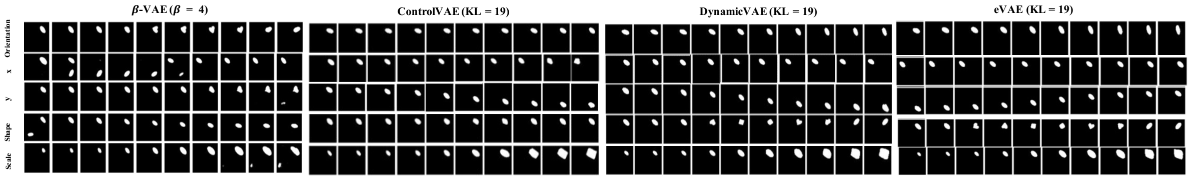


Figure 3.5: Latent traversal on dSprites using ellipse shapes. Each row represents a different latent factor while keeping others fixed. The first column shows the seed image for initialization. The remaining columns display images generated by manipulating the latent dimension z over the range [-3, 3].

Table 3.1: Disentangled representation performance on dSprites.

Metrics/Models	**β-VAE ($\beta = 4$)**	**ControlVAE**	**DynamicVAE**	**eVAE**
pos. x	0.0359	**0.7697**	0.7662	0.7286
pos. y	0.0243	0.7458	**0.7500**	0.7180
Shape	0.0116	0.0777	0.1276	**0.1449**
Scale	0.1507	0.6412	0.6591	**0.6605**
Orientation	0.0039	0.0961	0.1123	**0.1261**
MIG	0.1741	0.4492	0.4689	**0.4723**

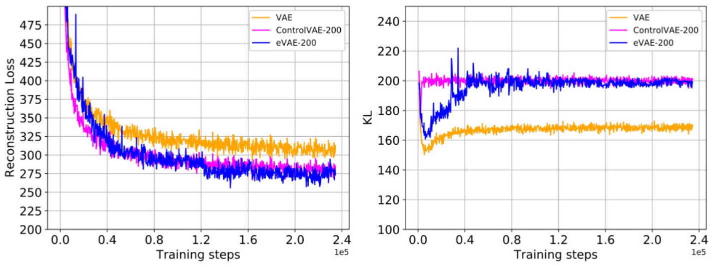
Table 3.2: Generation performance on CelebA.

Metrics/Models	VAE	ControlVAE	eVAE
FID	58.71 ± 0.207	55.79 ± 0.257	**54.06 ± 0.201**
SSIM	0.675 ± 0.0001	0.688 ± 0.0002	**0.692 ± 0.0001**

3.4.3 Performance in Image Generation

We evaluate eVAE for image generation using the CelebA dataset. For fair comparison, we set the KL target to 200, aligning with the best reconstruction quality in [127]. eVAE is tuned to achieve this target similar to ControlVAE. Figure. 3.6(a) shows that eVAE achieves the lowest reconstruction error, and the dynamic evolution of β allows the KL divergence to reach the desired target, as shown in Figure. 3.6(b).

During testing, eVAE generates new samples from the prior by discarding the encoder pathway [76]. For latent traverse, we plot latent codes from -3 to 3, covering a broader range.



(a) Reconstruction Loss

(b) KL

Figure 3.6: Performance comparison of different VAEs for image generation on CelebA.

3.4.4 Performance in Language Modeling

To demonstrate eVAE's effectiveness in preventing KL vanishing, we plot KL, reconstruction loss, and KL weight over iterations. The evolution of β reveals eVAE's automatic tuning capability and indicates that cost annealing (Cost-10k) suffers from KL vanishing. Cyclical annealing (Cyc-8), PID, and VGA achieve nonzero KL divergence, with eVAE showing the lowest reconstruction loss at 70 in word generation.

3.4.5 Sensitivity Analysis

Setpoint: The setpoint in Eq. equation 3.11 implicitly guides the model towards convergence at a target value c for specific representation learning. Setpoints vary based on differences in benchmark datasets and backbone models. In this section, we present a principle for selecting this hyperparameter guided by information theory, followed by experimental results on disentangled tasks.

Based on Shannon's source coding theorem [55] for bit compression, a D-ary block code of length n with size M has a compression limit:

$$H_D(\mathbf{X}) > \limsup_{n\to\infty} \frac{1}{n} \log_D M_n.$$

Simplifying for binary code ($D = 2$) with dataset size L, the information bottleneck c satisfies:

$$c \geq H(\mathbf{X}) > \log_2 L.$$

In practice, the KL setpoint should exceed $\log_2 L$ by a small margin. For dSprites, setting KL to 19 yields the best disentanglement (Table 3.3), while for CelebA, a KL of around 190 ensures optimal image generation (Table 3.4). In the language modeling task (Figure. 3.8), a KL setpoint above 2 prevents KL vanishing, maintaining stability during training.

Table 3.3: Sensitivity analysis of setpoints, crossover rates, and mutation rates in dSprites.

Metrics/Models	eVAE					
	(KL = 30)	(KL = 15)	(0.04, 0.001)	(0.04, 0.002)	(0.03, 0.001)	(0.03, 0.002)
MIG	0.121	0.1872	0.4723	0.4715	0.4662	0.4709
Distortion	29.5	15.1	9.9	9.9	9.6	10.2

Table 3.4: Sensitivity analysis of setpoints and crossover/mutation rates on CelebA.

Metrics/Models	eVAE					
	(KL = 190)	(KL = 210)	(0.12, 0.005)	(0.09, 0.005)	(0.1, 0.006)	(0.1, 0.004)
FID	58.32 ± 0.101	57.89 ± 0.205	58.07 ± 0.183	58.32 ± 0.198	57.95 ± 0.201	58.11 ± 0.189
Distortion	208	196	209	201	199	202

Crossover and Mutation Rates: In VGA, large crossover or mutation rates can prevent convergence, while overly small rates risk reducing diversity and causing premature convergence. Experimentally, a crossover rate around 0.03 with a mutation rate an order of magnitude smaller yields stable performance in the disentanglement task, as shown

Table 3.5: Sensitivity analysis of setpoints and crossover/mutation rates on CelebA.

Metrics/Models	**eVAE**					
	(KL = 190)	**(KL = 210)**	**(0.12, 0.005)**	**(0.09, 0.005)**	**(0.1, 0.006)**	**(0.1, 0.004)**
FID	58.32 ± 0.101	57.89 ± 0.205	58.07 ± 0.183	58.32 ± 0.198	57.95 ± 0.201	58.11 ± 0.189
Distortion	208	196	209	201	199	202

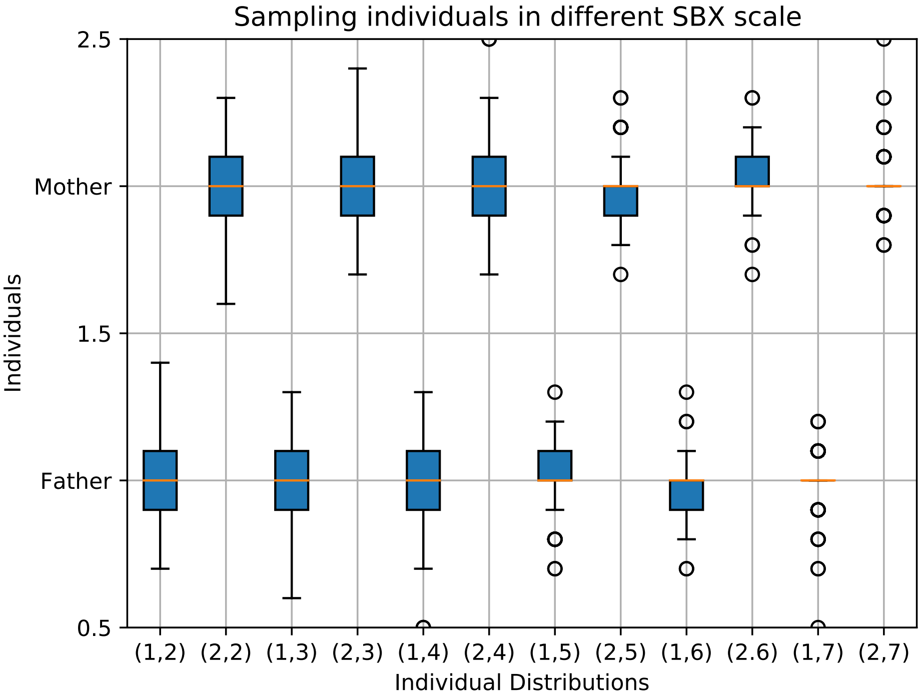


Figure 3.7: Box plot of generated distributions from a fixed parent pair (Father = 1, Mother = 2), varying $\eta^{\texttt{eVAE}}$ values for SBX. Higher $\eta^{\texttt{eVAE}}$ values reduce offspring diversity.

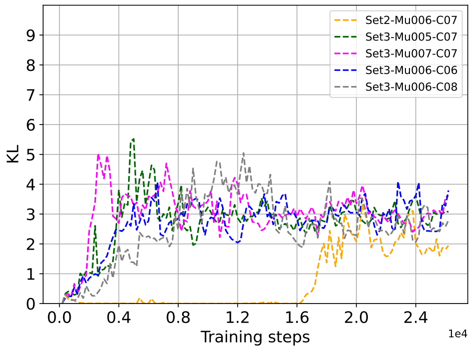


Figure 3.8: Numerical analysis of eVAE's impact on KL vanishing in PTB. The label 'Set-Mu-C' denotes setpoint, mutation, and crossover rates. Results show setpoints below 3 exacerbate KL vanishing.

in Table 3.3. For image generation, a crossover rate of around 10% and mutation rate of 0.5% achieve optimal performance (Table 3.4). For language generation a crossover rate of 7% and mutation rate of 0.6% prevents KL vanishing.

Individual Generation: Bit-wise crossover and mutation are unsuitable for generating sufficient diversity in deep learning models. To address this, VGA's simulated binary crossover (SBX) operator generates float 64 chromosomes. The SBX hyperparameter $\eta^{\texttt{eVAE}}$ controls sampling scale, with larger $\eta^{\texttt{eVAE}}$ values producing offspring closer to parent values (Figure. 4.2). While task-specific tuning is ideal, experimentally $\eta^{\texttt{eVAE}} = 5$ offers stability across tasks.

3.5 Summary of this Chapter

In this chapter, we propose eVAE to solve the the imbalance between representation inference and task fitting caused by surrogate loss in VAEs. We make the first attempt to introduce an *evolutionary variational autoencoder* (eVAE), building on the variational information bottleneck (VIB) theory and integrative evolutionary neural learning. eVAE integrates a variational genetic algorithm into VAE with variational evolutionary operators, including variational mutation, crossover, and evolution. Its training mechanism synergistically and dynamically addresses and updates the learning trade-off uncertainty in the evidence lower bound without additional constraints and hyperparameter tuning. Furthermore, eVAE presents an evolutionary paradigm to tune critical factors of VAEs and addresses the premature convergence and random search problem in integrating evolutionary optimization into deep learning. Experiments show that eVAE addresses the KL-vanishing problem for text generation with low reconstruction loss, generates all disentangled factors with sharp images, and improves image generation quality. eVAE achieves better disentanglement, generation performance, and generation-inference balance than its competitors.

CHAPTER 4

DECOUPLED VARIATIONAL AUTOENCODER

Variational deep learning has enabled robust representation learning, yet challenges in modeling hidden feature dependencies and achieving disentangled representations persist. Addressing these gaps leads to **RQ2**: *How can DVGM calibrate inference to separate disentangled and coupled representations?* To fulfill **RO2**, we introduce the **Contrastive Copula VAE (C^{2}VAE)**, which leverages neural copula functions and contrastive learning to address these challenges.

The subsequent sections detail how C^{2}VAE refines unsupervised disentangled learning by separating coupled and factorized representations, demonstrating its advantages over traditional TC-based approaches in improving inference stability and disentanglement quality.

4.1 Gaussian Copula-based VAE Differing Disentangled from Coupled Representations with Contrastive Posterior

In recent years, integrating stochastic variational inference into deep neural networks (DNNs) has formed a new learning paradigm - variational deep learning (VDL, or deep variational learning). VDL jointly characterizes dependencies between hidden neural features and between their distributions, going beyond deep neural principles and synergizing analytical statistical principles. Variational autoencoders (VAEs) represent a typical milestone for VDL, which transforms point-based autoencoders into process-oriented VAE for VDL. Various VAEs have been proposed in recent years to robustly fit the likelihoods of diverse data, such as tabular data [3, 4, 35, 110, 170], images [130, 146], and sequences [42, 70]. By estimating the likelihood over all data points, a VAE learns a smooth representation space under certain manifold hypotheses. It characterizes variational low-dimensional distributions corresponding to the input feature space and produces analytical results leveraging deep features and relations learned by DNNs. Consequently, VAEs further enhance representation learning for more challenging learning tasks such as out-of-domain detection [24, 91], image processing, time series anomaly detection [23, 94], multi-task learning [138], domain adaptation [67, 147], and continual learning [37, 177]. However, a significant gap remains in VAEs, i.e., exploring the distribution dependency between hidden features of DNNs, which has shown beneficial for leveraging stochastic factor interactions and downstream tasks [148, 165].

On the other hand, to enable more explainable variational reconstruction, a recent interest and challenge in VAE studies is to enable their unsupervised disentangled learning. Disentangled learning has been widely explored in supervised representation learning and classification [15] to learn single hidden units sensitive to single generative factor change but invariant to the variances of other factors. However, unsupervised disentangled learning in VAEs is more challenging with limited progress made. A common approach involves the total correlation (TC) to remedy the insufficient expressive posterior in the surrogate loss of vanilla VAEs. TC is a variant of mutual information to quantify the redundancy in multivariate dimensions [51]. For VAEs, TC is incorporated into their evidence lower bounds (ELBO) to induce factorized variational distributions with a $TC(\boldsymbol{Z})$ loss capturing the divergence between estimated posterior $q(\boldsymbol{Z})$ and prior $p(\boldsymbol{Z})$ over hidden features $\boldsymbol{Z} \in \mathscr{R}^d$:

$$\begin{aligned} TC(\boldsymbol{Z}) &= TC(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_d) \\ &= \mathbb{E}_{q(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_d)} \left[\log \frac{q(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_d)}{p(\boldsymbol{z}_1) p(\boldsymbol{z}_2) \ldots p(\boldsymbol{z}_d)} \right] \\ &= D_{KL}\big(q(\boldsymbol{Z}) || p(\boldsymbol{Z})\big). \end{aligned} \tag{4.1}$$

However, factorizing the prior, i.e., $p(\boldsymbol{Z}) := \prod_{j=1}^{d} p\left(\boldsymbol{z}_j\right)$ involves strong IID assumption between hidden features $\{\boldsymbol{z}_j\}$ [21]. Further, enforcing TC does not guarantee to capture dependent structures by the posterior distribution, no matter what the estimator is, by either mutual information estimators [13, 27, 45, 80, 138] or density ratio tricks [74, 178]. This is because the dependencies between hidden features may vary, where some are more strongly coupled than others, resulting in more (we call explicit) vs less (implicit) explanatory hidden features. For example, high cholesterol may be more affiliated with dietary habits and exercises than with age and gender. While the TC-based factorization ensures the independence between features, more explanatory (explicit) features may still be coupled with other less explanatory (implicit) ones in the hidden feature space. Hence, the TC factorization only guarantees the independence between those disentangled explicit features but ignores the dependencies in the entire hidden space. This forms another important gap in the existing VAE theories.

This work addresses the aforementioned gaps in modeling distribution dependency in the hidden neural space and further differentiates strongly coupled hidden features from weakly coupled features to improve unsupervised disentangled representations. To this end, we build a contrastive copula variational autoencoder (C^{2}VAE). First, as copula functions have been demonstrated powerful in learning high-dimensional dependence [111, 168], a neural Gaussian copula function learns the dependence between hidden features and identifies coupled representations [165]. Then, a self-supervised contrastive classification mechanism contrasts the disentangled factorized representations with these coupled representations sampled from a neural Gaussian copula function. Further, C^{2}VAE filters those strongly dependent hidden features captured by the copula function and induces an optimal posterior distribution characterizing more factorizable hidden features for improved disentangled representations.

We evaluate C^{2}VAE on four synthetic and natural image datasets: two grayscale (dSprites, SmallNORB) and two colored (3D Shapes, 3D Cars). It demonstrates the effect of the C^{2}VAE design and mechanisms in outperforming the existing TC-based models in terms of four disentanglement performance measures based on intervention, prediction, and information.

4.2 The C^{2}VAE Model

4.2.1 Preliminary

We introduce factorized posterior estimation, copula-coupled representation learning, and contrastive disentangled learning. These form the key constituents of our C^{2}VAE.

As shown in Figure. 4.1, the encoder output in C^{2}VAE is converted to two sets of representations: (1) the neural disentangled posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ as a multivariate Gaussian with a diagonal covariance structure; and (2) a copula coupled representation

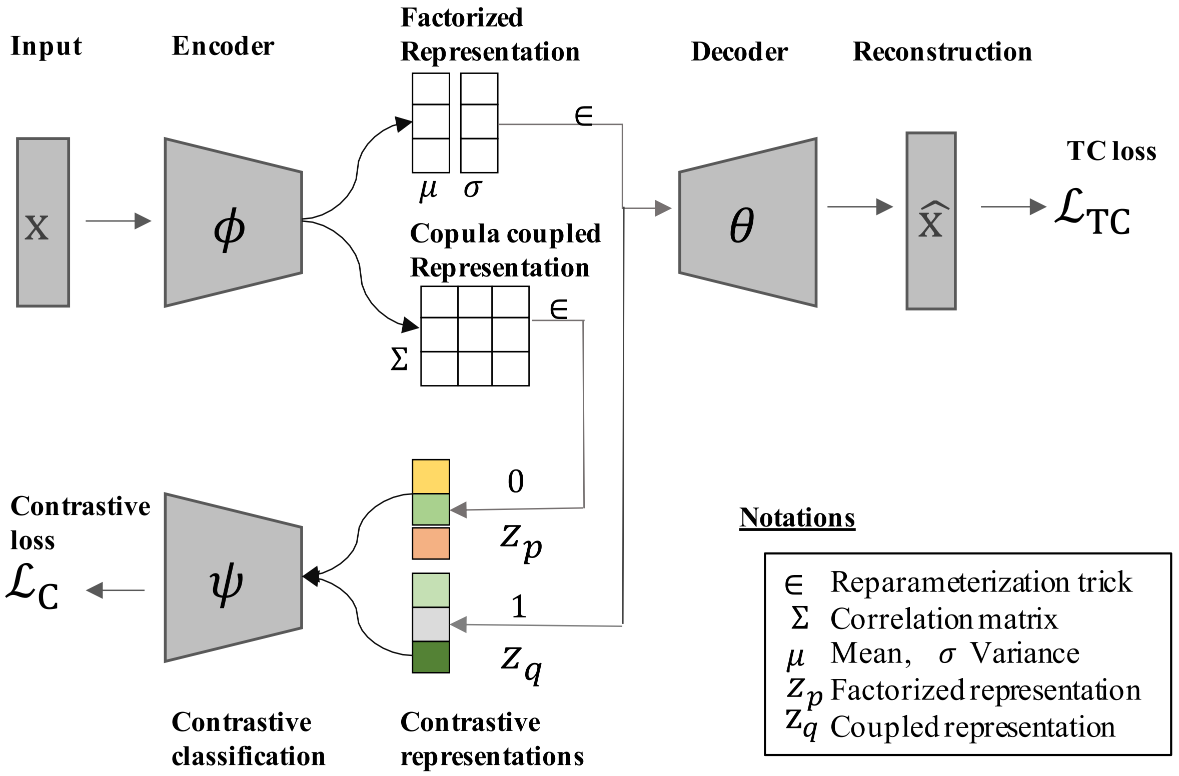

Input
Encoder
Factorized Representation
Decoder
Reconstruction
x
ϕ
μ
σ
∈
θ
x̂
TC loss
$\mathcal{L}_{TC}$
Copula coupled Representation
Σ
∈
Contrastive loss
$\mathcal{L}_C$
ψ
0
z_p
1
z_q
Contrastive classification
Contrastive representations
Notations
∈ Reparameterization trick
Σ Correlation matrix
μ Mean, σ Variance
z_p Factorized representation
z_q Coupled representation

Figure 4.1: C^{2}VAE: The architecture and contrastive learning of disentangled representations for contrastive copula VAE. $\mathscr{L}_{TC}$ optimizes disentangled factorized representations, $\mathscr{L}_C$ enhances the disentanglement by distinguishing factorized representations from coupled representations.

by a new encoder branch as a covariance encoder, which shares the same framework as the posterior encoder. This auxiliary encoder parameterized by ψ captures the dependence between hidden variables by learning the neural copula function. Copula learns the dependence coefficient matrix Σ. These two sets of representations share the dimension of hidden variables and learn their respective representations parameterized by mean μ_c and coefficient matrix Σ, respectively. All the notions can be found in Table 1.1.

4.2.2 Factorized Posterior Estimation for Disentangled Representations

VAE [76] is a generative model: $p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$ over data $\boldsymbol{x}$ and hidden features $\boldsymbol{z}$ learned in a deep manner. By sampling from the prior $p(\boldsymbol{z})$ of hidden features, the generative distribution $p(\boldsymbol{z}|\boldsymbol{x})$ can be approximated by a variational distribution $q(\boldsymbol{z}|\boldsymbol{x})$.

Further, to incorporate this generative learning into the autoencoder framework, a surrogate loss below is derived from approaching the reconstruction $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ by a decoder parameterized by θ to the inference $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ by an encoder parameterized by ϕ. The VAE learning process can be denoted as:

$$\begin{aligned}\log p(\boldsymbol{x}) &\geq \mathbb{E}_{q(\boldsymbol{z})} \log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\\ &\geq \mathbb{E}_{q(\boldsymbol{z})} \log p(\boldsymbol{x} \mid \boldsymbol{z}) + \mathbb{E}_{q(\boldsymbol{z})} \log \frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}\\ &\geq \mathbb{E}_{q(\boldsymbol{z})} \log p(\boldsymbol{x} \mid \boldsymbol{z}) - D_{KL}\big(q(\boldsymbol{z})||p(\boldsymbol{z})\big).\end{aligned}$$

When trained by a stochastic gradient variational Bayes (SGVB) estimator, VAE optimizes:

$$\mathscr{L}_{ELBO} := \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\Big[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\Big] - D_{KL}\big(q_\phi(\boldsymbol{z} \mid \boldsymbol{x})||p(\boldsymbol{z})\big). \tag{4.2}$$

VAE reconstructs samples by optimizing the likelihood function $\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\Big[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\Big]$ and learns a low-dimensional representation under a manifold hypothesis by regularizing $D_{KL}\big(q_\phi(\boldsymbol{z} \mid \boldsymbol{x})||p(\boldsymbol{z})\big)$.

To learn disentangled representations by VAEs for explanatory hidden generative factors, under the factorizable assumption, the posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is estimated by decomposing it into several independent and identically distributed (IID) conjugate distributions. Then, we convert the ELBO in Eq. (4.2) to a TC-based ELBO as follows:

$$\begin{aligned}\mathscr{L}_{\text{TC}} :=& \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\Big[\log p(\boldsymbol{x}|\boldsymbol{z}) - D_{KL}\big(q(\boldsymbol{z} \mid \boldsymbol{x})||\bar{q}(\boldsymbol{z} \mid \boldsymbol{x})\big)\\ &-D_{KL}\big(q(\boldsymbol{z})||\bar{q}(\boldsymbol{z})\big)\Big]\\ =& LL(\boldsymbol{x} \mid \boldsymbol{z}) - I(\boldsymbol{x},\boldsymbol{z}) - TC(\boldsymbol{z}),\end{aligned}$$

where $\bar{q}(\boldsymbol{z}) := \prod_{j=1}^{d} q\left(\boldsymbol{z}_j\right)$, $LL(\boldsymbol{x}|\boldsymbol{z})$ is the log-likelihood of data samples, $I(\boldsymbol{x},\boldsymbol{z})$ is the mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$. The TC term, i.e., the density ratio of two distributions $q(\boldsymbol{z}), \bar{q}(\boldsymbol{z})$, is estimated by the density ratio trick, $TC(\boldsymbol{z}) := D_{KL}\big(q(\boldsymbol{z})||\bar{q}(\boldsymbol{z})\big)$.

It is not feasible to compute this expectation (i.e., integral) analytically. Generally, using Monte Carlo methods to explicitly determine the density ratio results in a significant computational load, especially ill-suited for deep learning. The challenge deepens when this density ratio $r(\mathbf{z}) := \frac{q(\boldsymbol{z})}{\bar{q}(\boldsymbol{z})}$ is inaccessible due to the incalculability of either $q(\boldsymbol{z})$, $\bar{q}(\boldsymbol{z})$, or both. Under these circumstances, we pivot towards techniques designed for density ratio estimation. This involves harnessing the strong association between density ratio estimation and probabilistic classification. Specifically, we employ a discriminator, denoted as $\Psi(\mathbf{z})$, parameterized by ψ and use an adversarial learning paradigm to approximate the desired density ratio distribution. In this scenario, our approximation for $TC(\mathbf{z})$ is transformed into a classification problem for the discriminator Ψ, i.e.,

$$TC(\mathbf{z}) \sim r^*(\mathbf{z}) \sim \Psi_\psi(\mathbf{z}). \tag{4.3}$$

To derive the Ψ based density ratio item $r^*(\mathbf{z})$ in optimizing the $TC(\mathbf{z})$ item in Eq. (4.2.2), the density ratio trick is enacted as follows: We assume y to be the binary label indicating the origin of $\boldsymbol{z}$ from either $q(\boldsymbol{z})$ when $y = 1$ or $\bar{q}(\boldsymbol{z})$ when $y = 0$, and $\mathscr{P}(\boldsymbol{z}|y)$ signifies the ensuing conditional distribution:

$$\mathscr{P}(\mathbf{z} \mid y) := \begin{cases} q(\mathbf{z}) & (y = 1) \\ \bar{q}(\mathbf{z}) & (y = 0) \end{cases}.$$

According to the Bayes' theorem, the density ratio $r(\mathbf{z})$ is rewritten as:

$$\begin{aligned} r(\mathbf{z}) &= \frac{q(\mathbf{z})}{\bar{q}(\mathbf{z})} = \frac{\mathscr{P}(\mathbf{z} \mid y = 1)}{\mathscr{P}(\mathbf{z} \mid y = 0)} \\ &= \left(\frac{\mathscr{P}(y = 1 \mid \mathbf{z})\mathscr{P}(\mathbf{z})}{\mathscr{P}(y = 1)}\right)\left(\frac{\mathscr{P}(y = 0 \mid \mathbf{z})\mathscr{P}(\mathbf{z})}{\mathscr{P}(y = 0)}\right)^{-1} \\ &= \frac{\mathscr{P}(y = 0)}{\mathscr{P}(y = 1)}\frac{\mathscr{P}(y = 1 \mid \mathbf{z})}{\mathscr{P}(y = 0 \mid \mathbf{z})}. \end{aligned}$$

Then, We approximate the ratio of marginal densities using the proportion of sample sizes:

$$\frac{\mathscr{P}(y = 0)}{\mathscr{P}(y = 1)} \approx \frac{N_q}{N_{\bar{q}} + N_q}\left(\frac{N_{\bar{q}}}{N_{\bar{q}} + N_q}\right)^{-1} = \frac{N_q}{N_{\bar{q}}},$$

where N_q and $N_{\bar{q}}$ represent the number of samples drawn from $q(\mathbf{z})$ and $\bar{q}(\mathbf{z})$, respectively. For computational simplicity, we assume $N_q = N_{\bar{q}}$ so that we can prepare the same number of samples from the two distributions. Under this assumption, the density ratio $r(\mathbf{z})$ can be reformulated in terms of pseudo-label class probabilities,

$$r(\mathbf{z}) = \frac{\mathscr{P}(y = 1 \mid \mathbf{z})}{\mathscr{P}(y = 0 \mid \mathbf{z})}.$$

In fact, by introducing the pseudo-labels, we reduce the problem's complexity to a binary classification task. This implies that one distribution $\mathscr{P}(y = 0 \mid \mathbf{z})$ can be represented in terms of another $1 - \mathscr{P}(y = 1 \mid \mathbf{z})$. Consequently, we can derive a density ratio expression involving only pseudo-label class probabilities $\mathscr{P}(y = 1 \mid \mathbf{z})$. Namely,

$$\begin{aligned} r(\mathbf{z}) = \frac{\mathscr{P}(y = 1 \mid \mathbf{z})}{\mathscr{P}(y = 0 \mid \mathbf{z})} &= \frac{\mathscr{P}(y = 1 \mid \mathbf{z})}{1 - \mathscr{P}(y = 1 \mid \mathbf{z})} \\ &= \exp\left[\log\frac{\mathscr{P}(y = 1 \mid \mathbf{z})}{1 - \mathscr{P}(y = 1 \mid \mathbf{z})}\right] \\ &\approx \exp\left[\sigma^{-1}\big(\mathscr{P}(y = 1 \mid \mathbf{z})\big)\right]. \end{aligned}$$

where σ^{-1} is the logit function given by $\sigma^{-1}(\rho) = \log\left(\frac{\rho}{1-\rho}\right)$.

In fact, the pseudo-label class probabilities $\mathscr{P}(y = 1 \mid \mathbf{z})$ can be approximated using a discriminator $\Psi(\mathbf{z})$ parameterized by ψ. This discriminator accepts samples from both $q(\mathbf{z})$ and $\bar{q}(\mathbf{z})$, subsequently outputting a probability within the range $[0,1]$, indicating the likelihood of the samples originated from $q(\mathbf{z})$. Thus, we term $\Psi_{\psi}(\mathbf{z})$ as a probabilistic classifier. Therefore, we can use a function of probabilistic classifier $\Psi_{\psi}(\mathbf{z})$ as a proxy to acquire an estimator of the density ratio $r(\mathbf{z})$. Namely,

$$\begin{aligned} r(\mathbf{z}) &= \exp\left[\sigma^{-1}(\Psi(\mathbf{z}))\right] \\ &\approx \exp\left[\sigma^{-1}\big(\mathscr{P}(y = 1 \mid \mathbf{z})\big)\right] = r^*(\mathbf{z}). \end{aligned}$$

Hence, if we have a probabilistic binary classifier that outputs the probability of y with input $\boldsymbol{z}$, we can estimate the density ratio:

$$\begin{aligned} TC(\boldsymbol{z}) = D_{KL}\big(q(\boldsymbol{z})||\bar{q}(\boldsymbol{z})\big) &= \mathbb{E}_{q(\boldsymbol{z})} \log \frac{q(\boldsymbol{z})}{\bar{q}(\boldsymbol{z})} \\ &\approx \log \frac{q(\boldsymbol{z})}{\bar{q}(\boldsymbol{z})} \\ &= \log \frac{\mathscr{P}(y = 1 \mid \boldsymbol{z})}{\mathscr{P}(y = 0 \mid \boldsymbol{z})} \\ &= \log \frac{\mathscr{P}(y = 1 \mid \boldsymbol{z})}{1 - \mathscr{P}(y = 1 \mid \boldsymbol{z})} \\ &= \log \frac{\Psi(\boldsymbol{z})}{1 - \Psi(\boldsymbol{z})}, \end{aligned}$$

where $\Psi(\boldsymbol{z})$ is a classifier.

Subsequently, we incorporate the classifier-based density ratio into TC-based ELBO in Eq. (4.2.2) to acquire the surrogate loss:

$$\begin{aligned} \mathscr{L}_{\mathrm{TC}} :=& \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\Big[\log p(\boldsymbol{x}|\boldsymbol{z}) - D_{KL}\big(q(\boldsymbol{z} \mid \boldsymbol{x})||\bar{q}(\boldsymbol{z} \mid \boldsymbol{x})\big) \\ & - D_{KL}\big(q(\boldsymbol{z})||\bar{q}(\boldsymbol{z})\big)\Big], \\ =& LL(\boldsymbol{x} \mid \boldsymbol{z}) - I(\boldsymbol{x}, \boldsymbol{z}) - TC(\boldsymbol{z}) \\ =& \mathscr{L}_{\mathrm{ELBO}} - \gamma \mathbb{E}_{q(\boldsymbol{z})}\left[\log \frac{\Psi(\boldsymbol{z})}{1 - \Psi(\boldsymbol{z})}\right], \end{aligned}$$

where the TC term quantifies the dependencies between d-dimensional hidden variables, regularized by the degree hyperparameter γ.

Figure. 4.2 illustrates this TC-based decomposition of the vanilla ELBO in Eq. (4.2.2). The TC-based ELBO $\mathscr{L}_{\mathrm{TC}}$ is a loose bound to ensure the independence between factors $\boldsymbol{z}$ in the factorized posterior. It avoids a correlation structure between hidden variables toward disentangled representations.

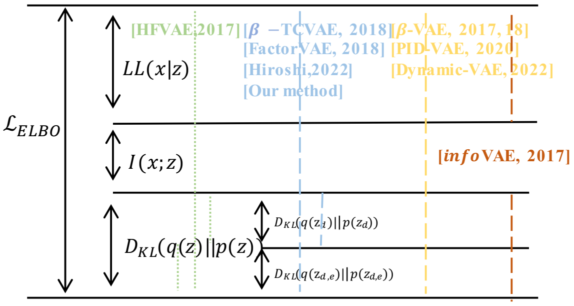


Figure 4.2: The element-wise decomposition of ELBO under the factorizable assumption based on information theory. Under the comparison of objectives in VAEs, we can conclude that TC-based factorization, e.g., [61, 74, 138], provides a tighter bound than other methods, e.g., [45].

4.2.3 Learning Coupled Representations by Copula

Typically, we employ adversarial mechanisms to train the discriminator-based density ratio estimation $r^*(\boldsymbol{z})$. The specific approach involves two main steps: firstly, we maximize the generator's ability in representation learning, denoted as $G = \{Encoder, Decoder\}$ based on Eq. (4.2.2), i.e.,

$$\max_{G} V(G) = H\big(q(\boldsymbol{z}),1\big), H\big(\bar{q}(\boldsymbol{z}),0\big) - \mathscr{L}_{\text{ELBO}}. \tag{4.4}$$

where V is a value function and D denotes the classifier which classifies whether sample $\boldsymbol{z}$ comes from disentangled distribution $q(\boldsymbol{z})$ or $\bar{q}(\boldsymbol{z})$.

Subsequently, we invert the pseudo-labels to induce adversarial, thereby enhancing the discriminator's capability to discern whether the sample originated from the disentangled distribution $q(\boldsymbol{z})$ or not, i.e.,

$$\max_{D} V(D) = H\big(q(\boldsymbol{z}),0\big), H\big(\bar{q}(\boldsymbol{z}),1\big). \tag{4.5}$$

During this second adversarial phase, while samples from the disentangled distribution can be easily drawn (sampled from the optimized encoder distribution), those from another distribution $\bar{q}(\boldsymbol{z})$ are often generated based on samples generated from $q(\boldsymbol{z})$ under the independence testing assumption [74]. Diverging from previous methodologies, we leverage the Gaussian copula function to generate samples that entangle each other from a copula distribution, denoted as $\boldsymbol{z}_c \sim \bar{q}_c(\boldsymbol{z})$. This methodology enables the discriminator to dissociate interrelated features, thereby empowering the model to identify more orthogonal representational spaces.

Specifically, we learn the coupled representations $\boldsymbol{z}_c$ in Eq. (4.9). A Gaussian copula $C(\cdot)$ captures the joint dependencies (with matrix Σ) between hidden features $\boldsymbol{z}$ of the learned posterior distribution after the encoding with parameters ϕ, as shown in Figure. 4.1. This identifies those coupled samples, which can be treated as drawn from the joint distribution $p(\boldsymbol{z};\mu_c,\sigma_c)$ with the reparameterization trick to enable the stochastic latent variables $\boldsymbol{z}$ to be represented by a deterministic function with parameters μ_c,σ_c.

To model the joint dependence between multivariates, copula learns a joint distribution over marginal distributions whose univariate marginal distributions are given as $F_d(\boldsymbol{z}_d)$ for variable $\boldsymbol{z}_d$. As the Gaussian copula fits most of the multivariate applications, we assume $\boldsymbol{z}_d \sim \text{Uniform}(0,1)$. Under Sklar's theorem [111], there exists a joint copula function $C(\cdot)$ which captures the dependencies between variables given the cumulative distribution function of multiple variables $\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_d$. Their multivariate cumulative distribution F can be modeled by copula over marginal distributions as:

$$F(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_d) = C\big(F_1(\boldsymbol{z}_1),\ldots,F_d(\boldsymbol{z}_d)\big), \tag{4.6}$$

Gaussian copula is elliptical whose marginal distribution $F(\boldsymbol{z})$ is subject to an elliptical family. With $u_i = F_i(\boldsymbol{z}_i)$, we can obtain the copula density function c by:

$$c(u_1,\ldots,u_d) = F\big(F^{-1}(\boldsymbol{z}_1),F^{-1}(\boldsymbol{z}_2),\ldots,F^{-1}(\boldsymbol{z}_d)\big), \tag{4.7}$$

where $F^{-1}(z)$ is the inverse cumulative distribution function of marginal Gaussian distribution F and the copula function c is a multivariate density normal distribution parameterized with mean μ_c and covariance matrix Σ.

When imposing a dependence assumption on latent representations, subject to a diagonal multivariate Gaussian distribution with mean μ_c and variance σ_c, a Gaussian copula joint distribution with covariance matrix Σ is sampled in neural settings and by a differentiable reparameterization. Here, we adopt the Cholesky-based parameterization of coefficient matrices to induce the latent samples. The Cholesky parameterization [148] is for the joint distribution of Gaussian copula, which factorizes a correlation matrix into a triangular matrix and its transposition for sampling the copula function directly in a high dimensional space. To ensure the numerical stability, i.e., the matrix needs to be positive definite, having all diagonal elements to be 1, we learn the components separately: $\Sigma = \mathbf{w} \cdot \mathbf{I} + \mathbf{v}\mathbf{v}^{\mathbf{T}}$, which is defined as:

$$
\begin{aligned}
\Sigma = & \begin{bmatrix} 1 & & Softplus(\Sigma;\phi_C) \\ & \ddots & \\ Softplus(\Sigma;\phi_C) & & 1 \end{bmatrix} + \\
& \begin{bmatrix} 1 & & Tanh(\Sigma;\phi_C) \\ & \ddots & \\ Tanh(\Sigma;\phi_C) & & 1 \end{bmatrix} \\
& \begin{bmatrix} 1 & & Tanh(\Sigma;\phi_C) \\ & \ddots & \\ Tanh(\Sigma;\phi_C) & & 1 \end{bmatrix}^T \\
= & \, \mathbf{w} \cdot \mathbf{I} + \mathbf{v}\mathbf{v}^{\mathbf{T}}.
\end{aligned}
\tag{4.8}
$$

The decomposition generates the positive definite covariance $\Sigma = LL^T$ for reparameterization. By sampling from the uniform distribution, we acquire the coupled representations: $\boldsymbol{z}_c = \boldsymbol{\mu}_c + \boldsymbol{\sigma}_c \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, maintaining the dependencies between individual dimensions.

Algorithm 4 shows the process of representation sampling. Different from the low-rank representation in [148, 149], we generate the coefficient matrix directly and replace the ReLU function by the Softplus function to ensure the positive definite property of the triangular matrix L.

Algorithm 4 Coupled representation learning with Gaussian copula

```
**Input:** Factorized mean $\boldsymbol{\mu}_d$ and covariance $\Sigma$
**Output:** Coupled representation $\boldsymbol{z}_c$
$\Sigma \leftarrow q_\phi(\boldsymbol{z}|\boldsymbol{x})$
$\mathbf{w} \leftarrow \text{Softplus}(\mathbf{W_1} \cdot \Sigma + \mathbf{b_1})$
$\mathbf{v} \leftarrow \text{Tanh}(\mathbf{W_2} \cdot \Sigma + \mathbf{b_2})$
$\Sigma \leftarrow \mathbf{w} \cdot I + \mathbf{v}\mathbf{v}^T$
$\mathbf{L} \leftarrow \text{CholeskyFactorization}(\Sigma)$
$\boldsymbol{z}_c \leftarrow \boldsymbol{\mu}_c + \mathbf{L} \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$
```

Consequently, with the coupled representations learned, we can apply the contrastive learning in Section 4.2.4 to distinguish the discrepancy over the factorized representation $\boldsymbol{z}_q$ and this coupled representation $\boldsymbol{z}_p$ following the contrastive learning framework in Eq. (4.9). This will make the learned posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ more factorizable.

4.2.4 Contrastive Learning to Enhance Disentangled Representations

Although different strategies are available to estimate the TC term in Eq. (4.2.2) with factorized factors in a DNN setting, there is no theoretical guarantee to acquire the optimal posterior for disentangled learning. This is attributed to the difficulty in modeling a heterogeneous and hierarchical posterior distribution while the TC-based ELBO decomposition is IID. In contrast, statistically, it is easier to model the correlation structure in the low-dimensional factorized factors.

Accordingly, to address the incorrect amortized inference and reconstruction error of the modified bound in Eq. (4.2.2) for disentanglement, the optimal posterior can be approximated in a contrastive way: we can learn an unsupervised classifier Ψ parameterized by ψ to distinguish the aforementioned disentangled representation $\boldsymbol{z}_d$ from the coupled representation $\boldsymbol{z}_c$ learned from the entire hidden space as discussed in Section 4.2.3. First, with these two representations $\boldsymbol{z}_d$ and $\boldsymbol{z}_c$, we define their (1) strongly independent (positive) pair $\big(\boldsymbol{z}_d, q(\boldsymbol{z})\big)$, where $\boldsymbol{z}_d$ can be treated as drawn from a (similar) target distribution $q(\boldsymbol{z})$, denoted as $H\big(q(\boldsymbol{z}),1\big)$ with a pseudo label 1 indicating that the learning representation is drawable from the target distribution; and (2) strongly dependent negative pair $(\boldsymbol{z}_c, \bar{q}(\boldsymbol{z}))$, where $\boldsymbol{z}_c$ is drawn from a dissimilar distribution $\bar{q}(\boldsymbol{z})$, denoted by $H\big(\bar{q}(\boldsymbol{z}),0\big)$ with a pseudo label 0. Then, we learn the classifier Ψ to determine whether the representation comes from the target or a dissimilar distribution with a contrastive loss $\mathscr{L}_C$:

$$\begin{aligned}\mathscr{L}_C &= H\big(q(\boldsymbol{z}),1\big) + H\big(\bar{q}(\boldsymbol{z}),0\big) \\ &= \frac{1}{N}\sum_{n=1}^{N}\Big[\ln\big(\sigma(\Psi_\psi(\boldsymbol{z}_d^n))\big) + \ln\big(1-\sigma(\Psi_\psi(\boldsymbol{z}_c^n))\big)\Big].\end{aligned} \tag{4.9}$$

where N is the number of samples. We train Ψ with the pseudo labels for $\Psi_\psi(\boldsymbol{z}_d^n)$ over disentangled posterior $\boldsymbol{z}_d$ and $\Psi_\psi(\boldsymbol{z}_c^n)$ over coupled representations $\boldsymbol{z}_c$. By minimizing $\mathscr{L}_C$, consequently, to enhance disentanglement, the contrastive loss and classifier Ψ ensure that the latent variables inferred by the encoder discard those features drawn from the similar distribution, i.e., retaining those independent features from the dissimilar distribution.

4.2.5 The C^{2}VAE Algorithm

We build C^{2}VAE as follows, with its architecture and information flow shown in Figure. 4.1. Given data $\mathscr{D} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\}$, we first learn its posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ per the factorization assumption. By applying the reparameterization trick, we train the TC-based ELBO with a factorized posterior $q(\boldsymbol{z})$. Then, the optimal posterior $q^*(\boldsymbol{z}|\boldsymbol{x})$ is trained in iterations that discard those dependent features. Specifically, the classifier $\Psi(\boldsymbol{z}_d, \boldsymbol{z}_c; \psi)$ is trained to distinguish the factorized representation $\boldsymbol{z}_d \sim q(\boldsymbol{z})$ from the coupled representation $\boldsymbol{z}_c \sim \bar{q}(\boldsymbol{z}; \mu_c, \sigma_c)$, where μ_c, σ_c are the parameters of the neural copula function discussed in Section 4.2.3.

Algorithm 5 shows the C^{2}VAE processes. It involves a two-phase optimization process. Parameters ϕ, θ are fixed in optimizing Eq. (4.5); the same in optimizing Eq. (4.4) by fixing parameters ψ.

Algorithm 5 The training process of C^{2}VAE

```
Require: Training data $\mathscr{D}$, training batch $B$
Ensure: Parameters of encoder $\phi$, decoder $\theta$, and classifier $\psi$
while not converged do
    for each $B$ in $\mathscr{D}$ do
        Generate the TC loss in terms of the discriminator
        Compute gradients of Eq. (4.4) with respect to $\theta$ and $\phi$
        Update the parameters of encoder $\theta$ and decoder $\phi$
    end for
    for each $B$ in $\mathscr{D}$ do
        Generate coupled representations per Algorithm 4
        Compute the gradients of Eq. (4.5) with respect to $\psi$
        Update parameters $\psi$ of the classifier
    end for
end while
```

4.3 Experiments

4.3.1 Data and Baselines

Datasets We evaluate C^{2}VAE on (1) two grayscale datasets: dSprites [61] as a binary 2D shape dataset with 737,280 samples, and SmallNORB [85] as a toy dataset with 48,600 synthetically rendered images; and (2) two color datasets: 3D Shapes [19] as a 3D shape dataset with 480,000 RGB images, and 3D Cars [121] as a 3D car dataset with 17,568 images generated from 24 rotation angles corresponding to 199 car models.

Baselines For a fair comparison with the total correlation-based VAEs, we compare C^{2}VAE with three VAEs, which involve some decomposition and approximation under a mild assumption and sharing the same deep frameworks. β-VAE [61] is a variant of the basic VAE with a penalty on D_{KL} in the vanilla ELBO by an additional coefficient β to acquire the disentangled representations. β-TCVAE [27] was the first work splitting the TC term to obtain the more factorizable posterior in a Monte Carlo estimator. FactorVAE [74] shows another way to acquire the factorized posterior in a density ratio estimator.

4.3.2 Effect of Learning Disentangled Representations

Disentanglement measures For comprehensive and fair quantitative evaluation, we use the following measures [22] to assess the effect of disentangled representation learning: (1) intervention-based: FactorVAE score (FAC); (2) information-based: Mutual Information Gap (MIG) [92]; and (3) prediction-based: Separated Attribute Predictability (SAP) [80], [74].

Further, to verify the effectiveness of a learned factorized prior, the Unsupervised Score [99] estimates the discrepancy between learned representations and optimal ones. The unsupervised score is measured by the Mutual Information (MI) score verifying the correlations between latent variables, the Total Correlation (TC), and the Normalized Wasserstein Distance (WCN) where their lower values identify stronger correlations between a Gaussian posterior and its marginals.

The settings of the baselines for disentangled representation learning are shown in Table 4.1.

Quantitative Results of Disentanglement. Table 4.2 depicts the quantitative evaluation results of each algorithm. The results of each entry are averaged over five random seeds. We follow the experimental settings in literature to set coefficients as $\beta = 4$ for β-VAE [61], $\beta = 4$ for β-TCVAE [74], $\gamma = 10$ for FactorVAE. This affects the relation between parts in the surrogate loss which plays an important role in balancing reconstruction and representation. In addition, $\gamma = 6.4$ is another optimal hyperparameter in [74] to generate disentangled representations for latent traversals.

On dSprites, C^{2}VAE outperforms the factorized VAE FactorVAE over all metrics except for the total correlation distance. In particular, C^{2}VAE performs well on latent metrics, SAP and FAC, rather than on representation-based metrics like MIG, which are estimated by the Monte Carlo sampling. Similar observations can be seen in the other three datasets. The unsupervised score shows the effect of the learned factorized distribution. C^{2}VAE fits the assumption with the lowest WCN in acquiring the most factorized posterior with the multiplication of marginal distributions.

Qualitative Results of Disentanglement. The disentanglement performance on four datasets over latent traversals can be seen in Figure. 4.3, Figure. 4.4, Figure.

Table 4.1: Four disentanglement datasets with their ground-truth generative factors. 'g' stands for grayscale images, and 'c' stands for color images. In SmallNORB and 3D Shapes, their 64-size version is used for the base model.

Dataset	ColorMode	Ground Truth Factors	ImageSize
dSprites	g	Shape: square, ellipse, heart Scale: 6 values linearly spaced in $[0.5,1]$ Orientation: 40 values in $[0,2\pi]$ Position X: 32 values in $[0,1]$ Position Y: 32 values in $[0,1]$	(64, 64, 1)
SmallNORB	g	Category: 0 to 9 Elevation: 9 values in [0, 8] Azimuth: 18 values in $[0,340]$ Lighting: 6 values in $[0,5]$	(64, 64, 1)
3D Shapes	c	Floor hue: 10 values in [0, 1] Wall hue: 10 values in $[0,1]$ Object hue: 10 values in $[0,1]$ Scale: 8 values in $[0,1]$ Shape: $\{0,1,2,3\}$ Orientation: 15 values in $[-30,30]$	(64, 64 ,3)
3D Cars	c	Car types Color rotation (2 types) Roof height	(64, 64, 3)

4.5, and Figure. 4.6. Latent traversals can assess the disentanglement properties of a trained generative model. For latent traversal in disentanglement, one modifies a specific dimension within a set range, such as -3 to +3 for standardized latent spaces, while holding other dimensions constant. Upon decoding the adjusted latent vectors, generated samples are assessed for variations. Successful disentanglement is evident when alterations in a single latent dimension correspond solely to one distinguishable factor in the data, like an object's rotation, with other attributes remaining unaltered. In summary, compared with FactorVAE, C^{2}VAE achieves the best disentanglement than others with less reconstruction error.

From Figure. 4.3, utilizing a latent traversal on the dSprites dataset with the trained models reveals that FactorVAE can disentangle the x and y features, as observed in rows 1 and 7 on the left side of Figure. 4.3, effectively. However, it entangles other features. Specifically, it exhibits partial disentanglement for scale and shape features, as shown by rows 9 and 10 on the left side of Figure. 4.3. Although the latent space

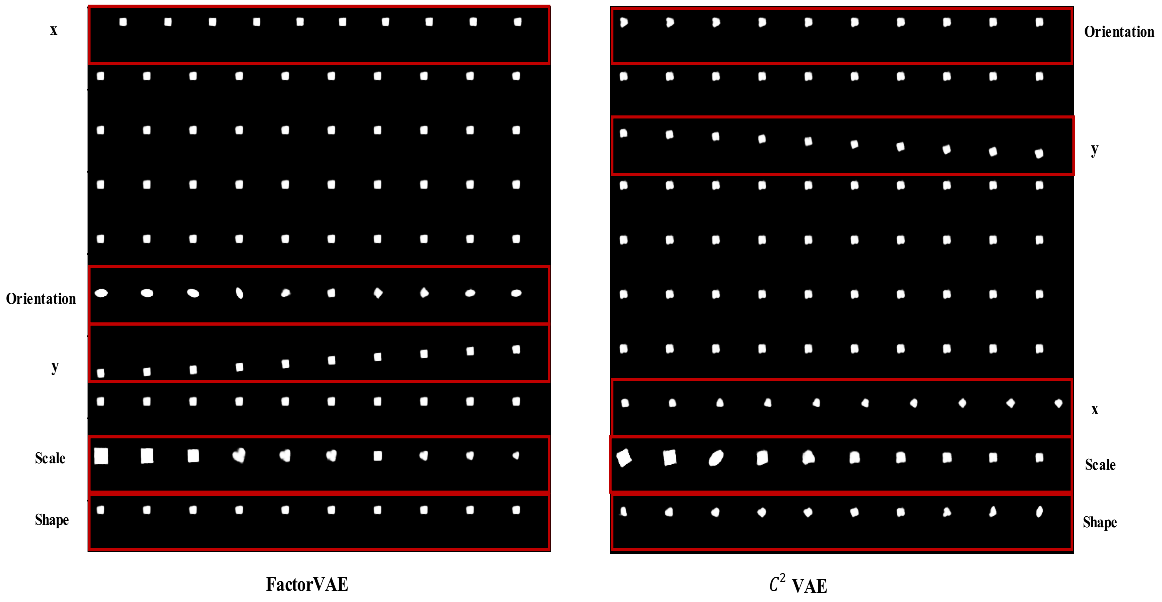


Figure 4.3: Traversal results of FactorVAE and C^{2}VAE on dSprites in terms of five factors: x, y, orientation, scale, and shape.

captures the continuous changes of scale and shape, i.e., transitioning from large to small shapes and from elliptical to circular and then square, these transformations remain intertwined with shape features. FactorVAE fails to disentangle the orientation feature, where continuous sampling in this dimension causes the shape, direction, and orientation features to become entangled. In contrast, C^{2}VAE displays disentanglement on the x and y features, as shown by rows 3 and 7 on the right side in Figure. 4.3, and exhibits partial disentanglement on the other three features, as shown by rows 1, 9 and 10 on the right side in Figure. 4.3, with shape features still somewhat embedded in the compressed latent dimensions.

Figure. 4.4 shows that a latent traversal on the SmallNORB dataset indicates that both models demonstrate some disentanglement capability on the azimuth, lighting and category features. However, compared to FactorVAE, C^{2}VAE captures a broader spectrum of lighting variations. Due to the inherent encoding capacities of the base models, both exhibit limited reconstruction abilities, resulting in blurred outcomes.

Further, the results in Figure. 4.5 show that a latent traversal of the 3D Shapes dataset demonstrates that FactorVAE adeptly disentangles features such as orientation, floor hue, and object hue, as illustrated by rows 1, 8 and 10 on the left side in Figure. 4.5. However, there is considerable overlap in the disentanglement of distinct color domains for wall hue, as shown in row 6 on the left side of Figure. 4.5, leading to ambiguity in

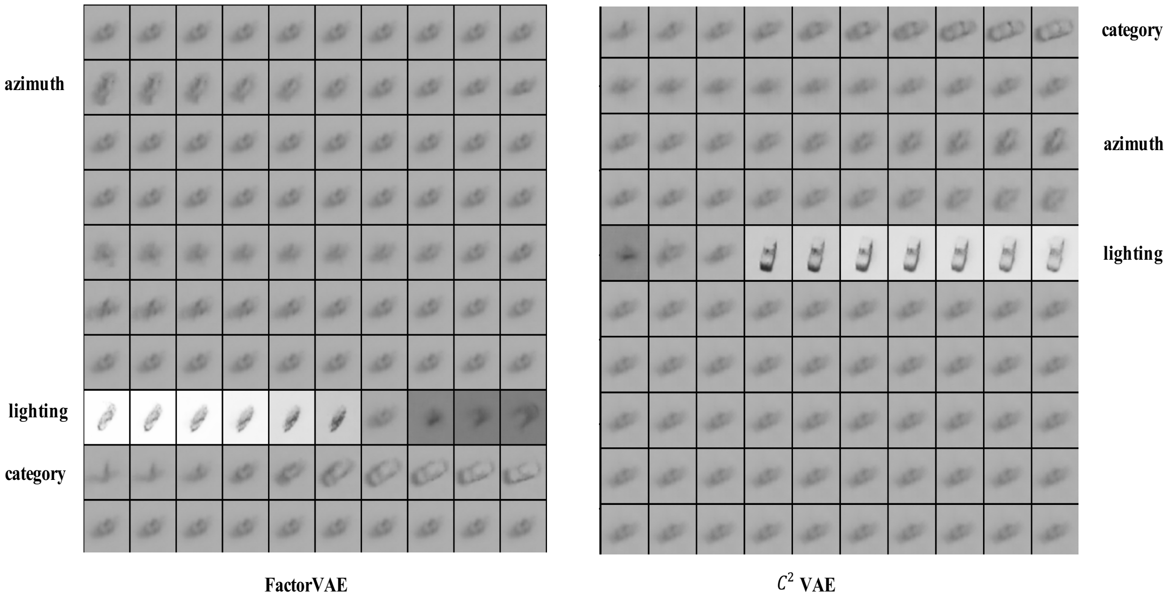


Figure 4.4: Traversal results of FactorVAE and C^{2}VAE on SmallNORB in terms of three factors: azimuth, lighting, and category.

representing wall color. Moreover, the object hue feature appears to be intertwined with both the wall hue and scale. In contrast, C^{2}VAE successfully disentangles five distinct features: wall hue, object hue, scale, floor hue, and orientation, as illustrated in rows 1, 3, 5 and 9 on the right side of Figure. 4.5. It's worth noting, however, that the shape attribute is entangled with the object hue feature, as indicated by row 4 on the right side of Figure. 4.5.

In addition, Figure. 4.6 shows that a latent traversal analysis of the 3D Cars dataset, where FactorVAE adeptly disentangles features related to color (as shown on the second row on the left side in Figure. 4.6) and the second rotation (as shown by the sixth row on the left side in Figure. 4.6). This suggests that variations in car color and orientation correspond independently and continuously to the changes of their latent variables, respectively. However, the disentanglement becomes less pronounced for features such as car type, overall rotation, and roof height. Conversely, C^{2}VAE more distinctly disentangles features, notably the two rotational aspects (as shown by the third and sixth rows on the right side of Figure. 4.6), roof height (illustrated by the first row on the right side in Figure. 4.6), and color (highlighted by the third and seventh rows on the right side in Figure. 4.6). The model also demonstrates a partial disentanglement of the car type feature, as shown by the third and ninth rows on the right side in Figure. 4.6. It is pertinent that continuous sampling in the latent space, as defined by

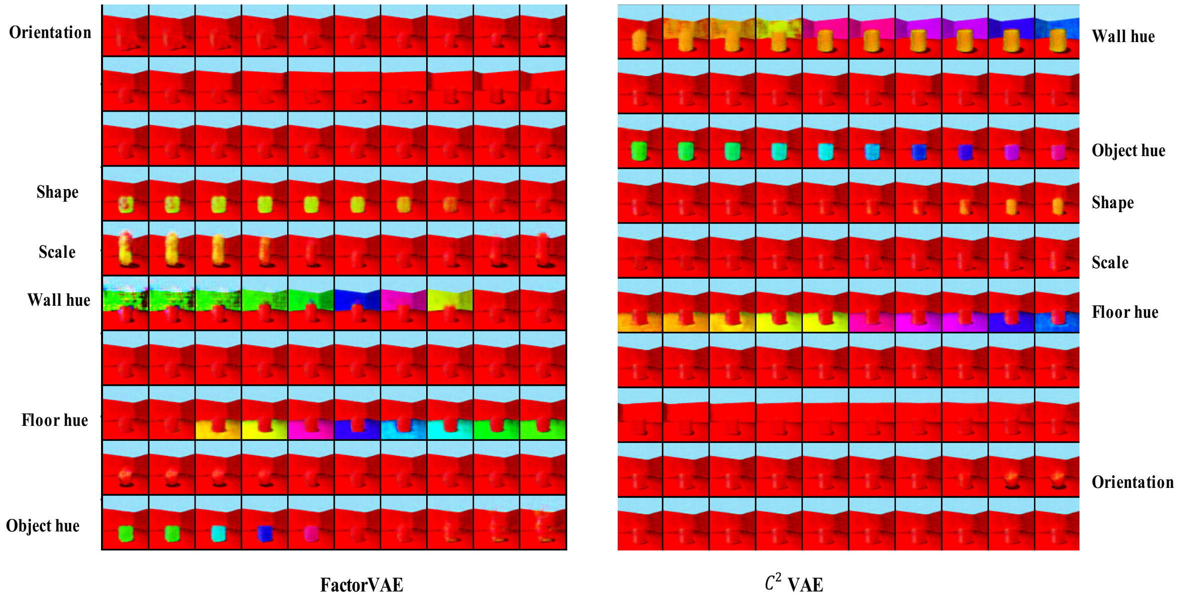


Figure 4.5: Traversal results of FactorVAE and C^{2}VAE on 3D Shapes in terms of six factors: orientation, shape, scale, wall hue, floor hue, object hue.

FactorVAE, occasionally results in reconstruction failure, manifesting this as incomplete imagery data. This phenomenon intimates that the FactorVAE model might suffer from a posterior collapse issue, which implies challenges in achieving stable representation learning and a smooth, continuous representational space.

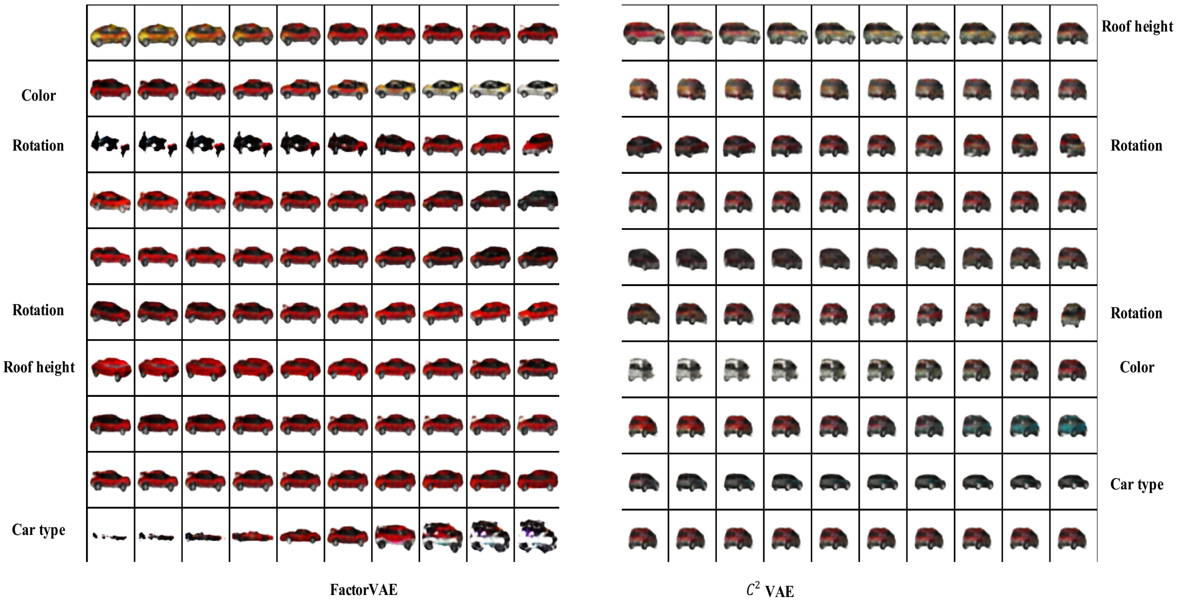


Figure 4.6: Traversal results of five factors of FactorVAE and C^{2}VAE on 3D Cars in terms color rotation1, rotation2, roof height, and car type.

Table 4.2: Performance (mean ± std) on different datasets and by different models w.r.t. different evaluation metrics. We evaluate β-VAE, β-TCVAE, and FactorVAE on dSprites and 3D Shapes.

dSprites	**Unsupervised Scores**			**MIG**	**SAP**	**FAC**
	MI	**TC**	**WCN**			
β-VAE ($\beta = 4$)	0.15 ± 0.06	10.7 ± 0.16	0.12 ± 0.41	0.19 ± 0,01	0.019 ± 0.009	0.78 ± 0.026
β-TCVAE	0.17 ±0.15	11.2±0.06	0.11 ± 0.007	0.17 ± 0.06	0.031 ± 0.006	0.70 ± 0.009
FactorVAE	0.11 ± 0.92	**10.05** ± 0.922	0.11 ± 0.009	0.20 ± 0.010	0.028 ± 0.015	0.81 ± 0.034
C^{2}VAE ($\gamma = 10$)	0.11 ± 0.33	11.8 ± 0.3	0.099 ± 0.026	0.20 ± 0.001	**0.044** ± 0.22	0.84 ± 0.001
C^{2}VAE ($\gamma = 6.4$)	**0.11** ± 0.57	12.4 ± 0.015	**0.079** ± 0.13	**0.21** ± 0.003	**0.035** ± 0.014	**0.85** ± 0.002

SmallNORB	**Unsupervised Scores**			**MIG**	**SAP**	**FAC**
	MI	**TC**	**WCN**			
β-VAE ($\beta = 4$)	0.17 ± 0.022	12.38 ±0.76	0.34 ± 0.14	0.10 ± 0.002	0.04 ± 0.008	0.59 ± 0.20
β-TCVAE	0.14 ± 0.012	12.1 ± 0.19	0.32 ± 0.001	0.13 ± 0.010	0.05 ± 0.003	0.60 ± 0.01
FactorVAE	0.21 ± 0.007	12.23 ±0.560	0.38 ±0.033	0.14 ± 0.019	0.061 ± 0.008	**0.62** ± 0.30
C^{2}VAE ($\gamma = 10$)	0.14 ±0.016	**11.55** ± 0.5	**0.25** ± 0.14	0.15 ± 0.0001	0.066 ±0.007	0.62 ± 0.0004
C^{2}VAE ($\gamma = 6.4$)	**0.14** ±0.017	11.96 ±0.734	0.27 ±0.011	**0.15** ± 0.017	**0.066** ± 0.006	0.61 ± 0.26

3D Shapes	**Unsupervised Scores**			**MIG**	**SAP**	**FAC**
	MI	**TC**	**WCN**			
β-VAE ($\beta = 4$)	0.15 ± 0.21	2.3 ± 0.16	0.12 ± 0.52	0.24 ± 0.005	0.058 ± 0.0005	0.93 ± 0.005
β-TCVAE	0.11 ± 0.007	2.1 ± 0.31	0.007 ± 0.052	0.32 ± 0.004	0.050 ± 0.009	0.97 ± 0.36
FactorVAE	0.11 ± 0.014	**1.5** ± 0.14	0.06 ± 0.042	**0.33** ± 0.004	0.047 ± 0.0004	0.98 ± 0.21
C^{2}VAE ($\gamma = 10$)	**0.08** ± 0.015	4.1 ± 0.48	0.08 ± 0.016	0.17 ± 0.003	0.054 ± 0.0002	0.95 ± 0.003
C^{2}VAE ($\gamma = 6.4$)	0.09 ± 0.006	2.8 ± 0.18	**0.06** ± 0.024	0.23 ± 0.002	**0.075** ± 0.001	**0.99** ± 0.025

3D Cars	**Unsupervised Scores**			**MIG**	**SAP**	**FAC**
	MI	**TC**	**WCN**			
β-VAE ($\beta = 4$)	0.18 ± 0.006	14.7 ± 0.78	0.38 ±0.03	0.04 ± 0.032	0.02 ± 0.098	0.82 ± 0.088
β-TCVAE	0.13 ± 0.012	11.6 ± 0.66	0.28 ± 0.03	**0.07** ± 0.024	0.02 ± 0.014	**0.89** ± 0.064
FactorVAE	0.16 ± 0.008	13.9 ± 0.98	0.37 ± 0.02	0.06 ± 0.029	0.02±0.005	0.86 ± 0.036
C^{2}VAE ($\gamma = 10$)	0.13 ± 0.007	**11.3** ± 0.76	0.14 ± 0.04	0.06 ±0.0001	0.02 ± 0.004	0.87 ± 0.0003
C^{2}VAE ($\gamma = 6.4$)	**0.12** ± 0.007	11.5 ± 0.80	**0.14** ± 0.04	0.05 ± 0.018	**0.02** ±0.002	0.86 ± 0.024

4.3.3 Trade-off between Reconstruction and Representation

By bringing the total correlation-based estimation into VAE optimization, C^2VAE acquires a loose bound in Eq. (4.2.2) than the original ELBO. This contributes to obtaining better disentanglement performance but hinders the model from overfitting data.

By evaluating the trade-off between reconstruction and representation, we draw the training curves of reconstruction loss over iterations. Figure. 4.7 shows a comparison of reconstruction error on dSprites with five random seeds on the two TC-based models. It shows that C^2VAE retains a stable training curve with smaller variance over five trials in acquiring a reasonable representation induced by a stable training stage as shown in [126]. In addition, C^2VAE induces more accurate amortized inference with the contrastive classifier to achieve a smaller reconstruction loss than the compared VAEs.

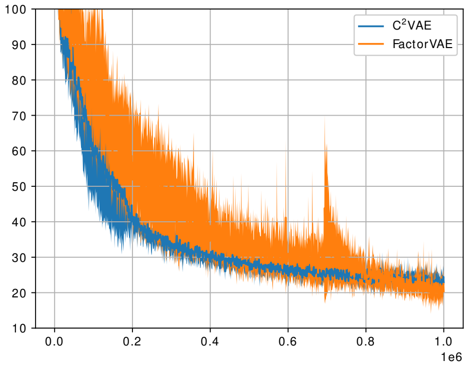


Figure 4.7: Learning curves on dSprites. The horizontal axis represents the number of training iterations, while the vertical axis represents the reconstruction loss.

4.3.4 Ablation Studies

We further investigate the effect of different coupled representations captured by various copula functions in C^2VAE. The following C^2VAE variants are created to capture different dependencies between dimensions.

- C^2VAE-I, where the contrastive posterior is estimated by permuting batch latent variables under the independence test assumption [10].
- C^2VAE-G, where the contrastive representation is sampled by Gaussian copula based on the learned neural posterior distribution.
- C^2VAE-S, where the contrastive representation is sampled by Student copula. Student copula is a copula function that incorporates the student's t-distribution.

It is often used to model variables with heavy-tailed distributions or when extreme values are more likely. It can be denoted as:

$$\begin{aligned}&C(u_1,u_2,\dots,u_n;\rho,\nu)\\ =&T\left(T^{-1}(u_1;\nu),T^{-1}(u_2;\nu),\dots,T^{-1}(u_n;\nu);\rho\right),\end{aligned}$$

where ρ refers to the correlation matrix, ν is the degree of freedom, and T refers to the cumulative distribution function of the t distribution.

- C^{2}VAE-M, where the contrastive representation is sampled by Gaussian mixture copula. The Gaussian mixture copula is a copula function based on the Gaussian mixture model, used for modeling the dependence structure among multivariate random variables. It combines the characteristics of the Gaussian distribution and copula functions, allowing flexible capture of different dependencies among variables. It can be denoted as:

$$\begin{aligned}&C\big(u_1,u_2,\dots,u_n;\theta)\\ =&\sum_{i=1}^{k} w_i\cdot C_i(\Phi^{-1}(u_1;\mu_{1i},\sigma_{1i}),\\ &\Phi^{-1}(u_2;\mu_{2i},\sigma_{2i}),\dots,\Phi^{-1}(u_n;\mu_{ni},\sigma_{ni})\big),\end{aligned}$$

θ refers to the correlation matrix, and w_i is the weight of each copula part.

Table 4.3 shows the results, where we can summarize that the C^{2}VAE with different representations may converge at different stages. The C^{2}VAE with Gaussian copula achieves better disentanglement performance w.r.t. the metric SAP.

Table 4.3: Representation and data fitting performance of the C^{2}VAE variants with different dependency functions for contrastive representation learning. SAP measures the disentanglement learning performance, and KL and Reconstruction Loss for the data fitting effect.

	C^{2}VAE-G	**C^{2}VAE-I**	**C^{2}VAE-S**	**C^{2}VAE-M**
SAP	**0.6**	0.4	0.2	0.2
KL	22	20	22	26
ReconstructionLoss	19	28	19	20

4.3.5 Hyperparameters

We further evaluate the disentanglement performance of C^{2}VAE with its variants in terms of hyperparameter γ. Table 4.4 shows the effect of coefficient γ on disentanglement.

It shows that the performance of disentanglement is sensitive to γ, where C^{2}VAE variants achieve the best performance at around $\gamma = 6$.

Table 4.4: Representation and data fitting performance of the C^{2}VAE variants by varying the hyperparameter γ. SAP evaluates the disentanglement learning performance, and KL and Reconstruction Loss measure the data fitting effect.

	$\gamma = 1$	$\gamma = 2$	$\gamma = 4$	$\gamma = 6$	$\gamma = 8$	$\gamma = 10$
SAP	0.54	0.55	0.55	0.64	0.52	0.70
KL	17	19	22	23	22	20
ReconstructionLoss	35	18	30	27	17	27

4.4 Summary of This Chapter

In this chapter, we propose a Contrastive Copula VAE (C^{2}VAE), a self-supervised variational autoencoder to jointly learn disentangled and coupled hidden factors and then enhance disentangled representation learning by a self-supervised classifier to eliminate coupled representations in a contrastive manner. To this end, a Contrastive Copula VAE (C^{2}VAE) follows the probabilistic principle without relying on prior knowledge about data and involving strong modeling assumptions on the posterior in the neural architecture. C^{2}VAE simultaneously factorizes the posterior (evidence lower bound, ELBO) with total correlation (TC)-driven decomposition to learn factorized disentangled representations and extracts the dependencies between hidden features by a neural Gaussian copula to learn coupled representations. Then, a self-supervised contrastive classifier differentiates the disentangled from the coupled representations, where a contrastive loss regularizes this contrastive classification and the TC loss eliminates entangled factors by strengthening disentangled representations. C^{2}VAE demonstrates a strong effect in enhancing disentangled representation learning. C^{2}VAE further contributes to improved optimization by addressing the TC-based VAE instability and the trade-off between reconstruction and representation.

CHAPTER 5

AUGUMENTED VARIATIONAL AUTOENCODER

Variational Autoencoders (VAEs) have shown promise in anomaly detection by mapping data to latent distributions that capture normal patterns. However, challenges such as latent space mismatching and data imbalance hinder their robustness. This motivates **RQ3**: *How can weak augmentation improve inference robustness in DVGM for anomaly detection?*

To achieve **RO3**, we propose the **Weakly Augmented VAE (WAVAE)**, a model that leverages self-supervised learning and weak augmentation to enhance latent space expressiveness. By maximizing mutual information in a contrastive training framework, WAVAE addresses these challenges, improving anomaly sensitivity and reconstruction accuracy. The following sections outline WAVAE's mechanisms and its ability to mitigate latent space disruptions for robust anomaly detection.

5.1 Weakly Augmented Variational Autoencoder for Time Series Anomaly Detection

Deep probabilistic generative models build the likelihood distribution on the whole dataset, mapping data points to a distribution and providing an uncertainty estimation for the generative process. In modelling the generative process of data in an unsupervised manner, the hidden structure of latent variables can be achieved at the same time. Among these, the VAEs model has been applied in modelling sequence data due to the flexible autoencoder mechanism compared with the unstable discriminators training in GANs and fast generation speed compared with the multi-step generation in diffusion models. Aside from learning the likelihood of time series data, VAE based model can exhibit anomaly detection ability in an unsupervised manner. The underlying assumption is that unknown anomaly patterns typically exhibit statistical characteristics that deviate significantly from the normal distribution.

To acquire the robust likelihood acquired by VAEs, recent methods tend to introduce the meta-prior to learn the spatiotemporal dependence inherent in posteriors. For example, time-varying priors that adapt to dynamic assumptions [34, 72, 116, 137] have demonstrated effectiveness and powerful capabilities in capturing sequence data likelihoods. Additionally, other studies [83, 89] have proposed the creation of task-specific priors based on a factorized assumption explicitly designed to model contextual dependence among data. Various strategies have been employed to achieve this, such as using prototype distribution-based representations optimized by meta-learning and decomposing contextual representations.

However, the implicit or explicit design manner of prior overlooks the learning of the hidden structure of latent space, inducing the mismatching from latent area to normal region, degrading the accuracy in detection anonymously. Besides, the anomalous sequence in real scenarios is limited compared to the normal sequence, and the anomalous point always disputes the normal points to learn the smooth and continuous latent space. The data imbalance and unsupervised manner will aggravate the mismatching from latent pattern to datapoint in $p_\theta(\mathbf{x}|\mathbf{z})$. Specifically, since these anomalies lack the spatiotemporal dependence on normal sequence, they disrupt the formation of a continuous and smooth latent space for normal samples. Consequently, representations sampled from these latent holes fail to accurately reconstruct input samples, causing a discrepancy between the representations and the reconstructed data. This mismatch significantly impairs an anomaly detector's performance and compromises the model's overall robustness. Figure. 5.1 presents a more intuitive understanding of this phenomenon. To address this, we try to increase the Expressive of encoders in a self-supervised manner.

In light of these challenges, we propose to improve data utilization using self-supervised learning (SSL) to enhance representation learning and induce latent space robustness. SSL [183] enables models to extract more informative representations from unlabeled data, leading to sufficient training. To achieve this, we employ data augmentation on unlabeled data through SSL strategies, facilitating the training of models by contrastive or adversarial methods for TSAD. Our contributions include:

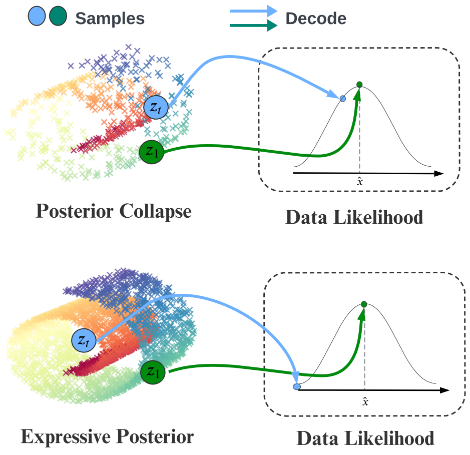


Figure 5.1: Comparison of the latent space induced by anomalies in non-robust VAE-based TSAD models (upper section) with the robust representation learning space fostered by WAVAE (lower section). We illustrate it by the Swissroll rather than Gaussian distribution. The upper part delineates the rise of the latent hole by a non-robust TSAD model and its effect on the model robustness. Anomalous sequences $\boldsymbol{x}_t^r$ (depicted within the blue window in the upper section), when encoded into the representation space, disrupt the structural integrity of the latent space. This disruption results in latent hole primarily because these anomalous sequences $\boldsymbol{x}_t^r$ lack the spatio-temporal coherence inherent in the normal sequence $\boldsymbol{x}_1^r$. Consequently, sampling from these discontinuous regions leads to a mismatch between the representation (indicated by the blue dot $\boldsymbol{z}_t$) and its generation (also shown by the blue dot in the likelihood function), as illustrated in the upper section, a disproportionately high likelihood function mass characterizes the representation in the latent space. In such scenarios, the TSAD model may erroneously classify an anomaly as normal, compromising its robustness. In contrast, the lower section demonstrates how data augmentation via the WAVAE model can engender a more continuous and smoothly distributed data likelihood (as depicted in the central part of the bottom figure). In this context, representation $\boldsymbol{z}_t$) encoded by anomalous sequences $\boldsymbol{x}_t^a$ sampled from regions outside the normal latent space are associated with a lower likelihood function mass, thereby enhancing the robustness and efficacy of anomaly detection in TSAD tasks.

- **A generative self-supervised learning framework for TSAD**: WAVAE presents an enhanced generative framework using self-supervised learning, with a likelihood function for learning and the derivation of a surrogate error for optimization. This novel approach sets the stage for more effective model design in VAE-based TSAD.

- **Deep and shallow learning in augmented models**: WAVAE implements weakly augmented anomaly detection by incorporating data augmentation, enabling undergo thorough training with support from augmented counterparts. Both deep and shallow learning methods are introduced to integrate these two models effectively.

Extensive experiments on five public datasets demonstrate the effectiveness of our approach. WAVAE achieves superior performance w.r.t. ROC-AUC and PR-AUC, surpassing the state-of-the-art models. Additionally, comprehensive ablation studies verify the performance over the design of the VAE model, time series preprocessing, and sensitivity analysis on different modules and hyperparameters in deep optimization.

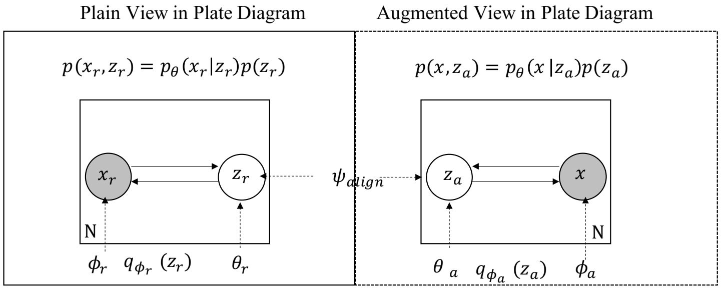

Plain View in Plate Diagram
Augmented View in Plate Diagram
$p(x_r, z_r) = p_\theta(x_r|z_r)p(z_r)$
$p(x, z_a) = p_\theta(x|z_a)p(z_a)$
x_r
z_r
ψ_{align}
z_a
x
N
N
ϕ_r $q_{\phi_r}(z_r)$ θ_r
θ_a $q_{\phi_a}(z_a)$ ϕ_a

Figure 5.2: Graphical model for augmented variational autoencoders. Under the plate notation rules, a white circle denotes a hidden (or latent) variable, while a grey circle signifies an observed variable. The variables contained in the square denote local variables, which are independently repeated N times. Dashed arrow edges imply conditional dependence. Dotted lines represent parameters. Our methodology utilizes two generative models. The inference part of models, i.e., q_{ϕ_r} and q_{ϕ_a}, encodes the raw input, denoted as $\boldsymbol{x}_\mathrm{r}$, and the augmented input, $\boldsymbol{x}_\mathrm{a}$, into their respective low-dimensional representations, $\boldsymbol{z}_\mathrm{r}$ and $\boldsymbol{z}_\mathrm{a}$. Subsequently, the generative part of models p_{θ_r} and p_{θ_a} samples the latent space to reconstruct the input samples, respectively. We employ a ψ parameterized module to synchronize the learning outcomes of both models.

5.2 The WAVAE Model

We first provide the problem definition for generative model-based TSAD. Then, we depict the structure of an augmented generative model with random variables and their dependency structure. The augmented generative anomaly detection model operates within the framework of probabilistic generative modeling, employing self-supervised techniques to augment the latent variables $\boldsymbol{z}$ during training a generative model, thereby enhancing a deep model's fit to the data likelihood. Accordingly, we implement a self-supervised VAE based on input data augmentation, which preprocesses the input data $\boldsymbol{x}$ to generate latent variables $\boldsymbol{z}_{\mathrm{a}}$ with different views. To align the likelihood functions of the raw and augmented models, we introduce two distinct mutual information loss functions, one grounded on depth and the other on statistics. By maximizing the mutual information between them, the models draw samples more fitting the same distribution.

5.2.1 Problem Definition

Time series data is succinctly represented as $\mathscr{X} := \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{n}$, encompassing n time-stamped observations $\boldsymbol{x} \in \mathbb{R}^c$ situated within a c-dimensional representation space, each paired with a discrete observation $\boldsymbol{y}$. The observation $\boldsymbol{y}$ is assigned discrete values across l predefined classes, delineated as $\boldsymbol{y} \in \{0, 1, \ldots, l-1\}$. Here, c denotes the feature dimensionality at each time point, categorizing the dataset as an MTS when $c > 1$ and as a univariate time series for $c = 1$. In the figures and Eq.s followed, to enhance notational conciseness, we abbreviate raw as r and augmentation as a.

In generative models-based TSAD, the focus is on learning a reconstructing model, i.e., $\mathscr{M}_{\mathrm{normal}}$, to model the mass of loglikelihood of the majority of normal data points within the entire dataset $\mathscr{X} = \{\mathscr{X}_{\mathrm{normal}}, \mathscr{X}_{\mathrm{abnormal}}\}$. Anomalies are then identified in an unsupervised, end-to-end fashion by calculating the anomaly score, denoted as $AS(\boldsymbol{x}, \hat{\boldsymbol{x}})$. It quantifies the difference between a given input $\boldsymbol{x}$ and its modeled copy $\hat{\boldsymbol{x}}$ as reconstructed by $\mathscr{M}_{\mathrm{normal}}$. This approach is feasible, assuming that the log-likelihood learned from normal observations will diverge notably when encountering anomalous data, yielding elevated anomaly scores.

5.2.2 Data Augmentation-guided Probabilistic Generative Model

The plate diagram in Figure. 5.2 defines an augmented probabilistic generative model. The upper part of the diagram specifies the learned joint distribution of the original data

$\boldsymbol{x}_\mathrm{r}$ and its latent variable $\boldsymbol{z}_\mathrm{r}$, i.e., $p(\boldsymbol{x}_\mathrm{r},\boldsymbol{z}_\mathrm{r})$, where the latent variable $\boldsymbol{z}_\mathrm{r}$ is generated by an inference network $q_{\phi_\mathrm{r}}(\boldsymbol{z}_\mathrm{r}|\boldsymbol{x}_\mathrm{r})$ parameterized by ϕ_r, and the reconstructed variable $\hat{x}_\mathrm{r}$ is produced by a generative network $p_{\theta_\mathrm{r}}(\boldsymbol{x}_\mathrm{r}|\boldsymbol{z}_\mathrm{r})$ parameterized by θ_r. The model optimizes an approximate surrogate error $\mathscr{L}^\mathrm{r}_\mathrm{ELBO}$, comprising a reconstruction loss that maximizes the likelihood distribution $\mathscr{L}^\mathrm{r}_\mathrm{R}$ and a D_KL loss that minimizes the discrepancy between the prior of latent variables and their variational posterior $\mathscr{L}^\mathrm{r}_\mathrm{I}$, i.e.

$$\mathscr{L}^\mathrm{r}_\mathrm{ELBO} := \underbrace{\mathbb{E}_{q_{\phi_\mathrm{r}}(\boldsymbol{z}_\mathrm{r}|\boldsymbol{x}_\mathrm{r})}\Big[\log p_{\theta_\mathrm{r}}(\boldsymbol{x}_\mathrm{r} \mid \boldsymbol{z}_\mathrm{r})}_{\mathscr{L}^\mathrm{r}_\mathrm{R}}\Big]$$

$$- \beta \underbrace{D_{KL}\big(q_\phi(\boldsymbol{z}_\mathrm{r} \mid \boldsymbol{x}_\mathrm{r})||p(\boldsymbol{z})}_{\mathscr{L}^\mathrm{r}_\mathrm{I}}\big).$$

The lower part of the plate diagram outlines the probabilistic model of the joint distribution of the augmented view data $\boldsymbol{x}_\mathrm{a}$ and latent variables $\boldsymbol{z}_\mathrm{a}$, i.e., $p(\boldsymbol{x}_\mathrm{a},\boldsymbol{z}_\mathrm{a})$ with latent variables $\boldsymbol{z}_\mathrm{a}$ derived from an augmented inference network $q_{\phi_\mathrm{a}}(\boldsymbol{z}_\mathrm{a}|\boldsymbol{x})$, i.e., $\boldsymbol{z}_\mathrm{a} \sim q_{\phi_\mathrm{a}}(\boldsymbol{z}_\mathrm{a}|\boldsymbol{x})$. Similar to the above, the model optimizes an augmented reconstruction loss $\mathscr{L}^\mathrm{a}_\mathrm{R}$ and inference loss $\mathscr{L}^\mathrm{a}_\mathrm{I}$, i.e.,

$$\mathscr{L}^\mathrm{a}_\mathrm{ELBO} := \underbrace{\mathbb{E}_{q_{\phi_\mathrm{a}}(\boldsymbol{z}_\mathrm{a}|\boldsymbol{x}_\mathrm{a})}\Big[\log p_{\theta_\mathrm{a}}(\boldsymbol{x}_\mathrm{a} \mid \boldsymbol{z}_\mathrm{a})}_{\mathscr{L}^\mathrm{a}_\mathrm{R}}\Big]$$

$$- \beta \underbrace{D_\mathrm{KL}\big(q_\phi(\boldsymbol{z}_\mathrm{a} \mid \boldsymbol{x}_\mathrm{a})||p(\boldsymbol{z})}_{\mathscr{L}^\mathrm{a}_\mathrm{I}}\big).$$

On one hand, both models strive to fit their respective data distribution likelihoods. On the other, we leverage the advantage of data augmentation by maximizing the mutual information $I(\boldsymbol{z}_\mathrm{r},\boldsymbol{z}_\mathrm{a})$ between two latent models, optimizing an MI loss parameterized by ψ (in deep learning approximation) to train the models for maximal data likelihood jointly.

Given the variety of latent variable augmentations, we augment the raw data to augment the model. In summary, we propose an augmented probabilistic generative model to learn a joint likelihood function $p(\boldsymbol{x}_\mathrm{r},\boldsymbol{z}_\mathrm{r},\boldsymbol{x}_\mathrm{a},\boldsymbol{z}_\mathrm{a})$ for anomaly detection while simultaneously optimizing an inference network parameterized by $\phi_\mathrm{r},\phi_\mathrm{a}$, a generative network parameterized by $\theta_\mathrm{r},\theta_\mathrm{a}$, and an alignment network parameterized by ψ. ψ can be parameterized by neurons in deep learning approximation and pseudo-parameters in shallow learning. The generative process is as follows:

$$p\,(\boldsymbol{x}_\mathrm{r},\boldsymbol{x}_\mathrm{a}) = \int p\,(\boldsymbol{x}_\mathrm{r},\boldsymbol{x}_\mathrm{a},\boldsymbol{z}_\mathrm{r},\boldsymbol{z}_\mathrm{a})\mathrm{d}\boldsymbol{z}_\mathrm{r}\mathrm{d}\boldsymbol{z}_\mathrm{a}, \tag{5.1}$$

where $\boldsymbol{x}_{\mathrm{r}}$ represents the raw input data points, $\boldsymbol{x}_{\mathrm{a}}$ refers to the augmented samples based on the input, $\boldsymbol{z}_{\mathrm{r}}$ refers to the raw latent variable, and $\boldsymbol{z}_{\mathrm{a}}$ refers to the augmented latent variable.

The joint distribution is often too high-dimensional and sophisticated to solve directly. To address this, a tractable variational distribution $q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})$ is employed as an approximation within the framework of variational inference (VI). Due to the computational convenience it offers, we typically take the logarithm of the distribution. Consequently, as depicted in Eq. 5.1, the likelihood of data that encompasses latent variables can be decomposed as follows:

$$\begin{aligned}
&p\left(\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}}\right)\\
=&\int \frac{p\left(\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}},\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}\right)q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}{q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}\mathrm{d}\boldsymbol{z}_{\mathrm{r}}\mathrm{d}\boldsymbol{z}_{\mathrm{a}},
\end{aligned} \tag{5.2}$$

and we can obtain the log versions as follows:

$$\begin{aligned}
&\log p\left(\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}}\right)\\
=&\log\int \frac{p\left(\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}},\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}\right)q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}{q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}\mathrm{d}\boldsymbol{z}_{\mathrm{r}}\mathrm{d}\boldsymbol{z}_{\mathrm{a}}.
\end{aligned} \tag{5.3}$$

Given the log function is convex, we can obtain a lower bound by Jensen's inequality:

$$\begin{aligned}
&\log p\left(\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}}\right)\\
=&\log\mathbb{E}_{q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}|\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}})}\left[\frac{p\left(\boldsymbol{x}_{\mathrm{r}},x_{\mathrm{a}},\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}\right)}{q\left(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}\mid\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}}\right)}\right]\\
\geq&\mathbb{E}_{q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}|\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}})}\log\left[\frac{p\left(\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}},\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}\right)}{q\left(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}\mid\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}}\right)}\right]\\
=&\mathbb{E}_{q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}|\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}})}\log\left[\frac{p(\boldsymbol{x}_{\mathrm{r}}\mid\boldsymbol{z}_{\mathrm{r}})p\left(\boldsymbol{x}_{\mathrm{a}}\mid\boldsymbol{z}_{\mathrm{a}}\right)p\left(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}\right)}{q(\boldsymbol{z}_{\mathrm{r}}\mid\boldsymbol{x}_{\mathrm{r}})q\left(\boldsymbol{z}_{\mathrm{a}}\mid x_{\mathrm{a}}\right)}\right]\\
=&\underbrace{\mathbb{E}_{q(\boldsymbol{z}_{\mathrm{r}}|\boldsymbol{x}_{\mathrm{r}})}\log[p(\boldsymbol{x}_{\mathrm{r}}\mid\boldsymbol{z}_{\mathrm{r}})]+\mathbb{E}_{q(\boldsymbol{z}_{\mathrm{a}}|\boldsymbol{x}_{\mathrm{a}})}\log\left[p\left(\boldsymbol{x}_{\mathrm{a}}\mid\boldsymbol{z}_{\mathrm{a}}\right)\right]}_{1}\\
&+\underbrace{\mathbb{E}_{q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}|\boldsymbol{x}_{\mathrm{r}},\boldsymbol{x}_{\mathrm{a}})}\log\left[\frac{p\left(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}\right)}{q(\boldsymbol{z}_{\mathrm{r}}\mid\boldsymbol{x}_{\mathrm{r}})q\left(\boldsymbol{z}_{\mathrm{a}}\mid\boldsymbol{x}_{\mathrm{a}}\right)}\right]}_{2}.
\end{aligned} \tag{5.4}$$

As we can see, the 1 part can be decomposed into two reconstruction losses, i.e., $1 = \mathscr{L}_{\mathrm{R}}^{\mathrm{r}} + \mathscr{L}_{\mathrm{R}}^{\mathrm{a}}$, and the 2 part in Eq. 5.4 can be decomposed as:

$$
\begin{aligned}
&\mathbb{E}_{q(\boldsymbol{z}_\mathrm{r},\boldsymbol{z}_\mathrm{a}|\boldsymbol{x}_\mathrm{r},\boldsymbol{x}_\mathrm{a})}\log\left[\frac{p(\boldsymbol{z}_\mathrm{r},\boldsymbol{z}_\mathrm{a})}{q(\boldsymbol{z}_\mathrm{r}\mid\boldsymbol{x}_\mathrm{r})q(\boldsymbol{z}_\mathrm{a}\mid\boldsymbol{x}_\mathrm{a})}\right]\\
=&\mathbb{E}_{q(\boldsymbol{z}_\mathrm{r},\boldsymbol{z}_\mathrm{a}|\boldsymbol{x}_\mathrm{r},\boldsymbol{x}_\mathrm{a})}\log\left[\frac{p(\boldsymbol{z}_\mathrm{r},\boldsymbol{z}_\mathrm{a})}{p(\boldsymbol{z}_\mathrm{r})p(\boldsymbol{z}_\mathrm{a})}\right]\\
&+\mathbb{E}_{q(\boldsymbol{z}_\mathrm{r},\boldsymbol{z}_\mathrm{a}|\boldsymbol{x}_\mathrm{r},\boldsymbol{x}_\mathrm{a})}\log\left[\frac{p(\boldsymbol{z}_\mathrm{r})p(\boldsymbol{z}_\mathrm{a})}{q(\boldsymbol{z}_\mathrm{r}\mid\boldsymbol{x}_\mathrm{r})q(\boldsymbol{z}_\mathrm{a}\mid\boldsymbol{x}_\mathrm{a})}\right]\\
=&\underbrace{\mathbb{E}_{q(\boldsymbol{z}_\mathrm{r},\boldsymbol{z}_\mathrm{a}|\boldsymbol{x}_\mathrm{r},\boldsymbol{x}_\mathrm{a})}\log\left[\frac{p(\boldsymbol{z}_\mathrm{r},\boldsymbol{z}_\mathrm{a})}{p(\boldsymbol{z}_\mathrm{r})p(\boldsymbol{z}_\mathrm{a})}\right]}_{A}\\
&-\underbrace{D_{KL}[q(\boldsymbol{z}_\mathrm{r}\mid\boldsymbol{x}_\mathrm{r})\|p(\boldsymbol{z}_\mathrm{r})]}_{i}-\underbrace{D_{KL}[q(\boldsymbol{z}_\mathrm{a}\mid\boldsymbol{x}_\mathrm{a})\|p(\boldsymbol{z}_\mathrm{a})]}_{ii},
\end{aligned}
\tag{5.5}
$$

where the 2 part can be the combination of two inference losses and the mutual information between latent variables, i.e., $2 = \mathscr{L}_\mathrm{I}^\mathrm{r} + \mathscr{L}_\mathrm{I}^\mathrm{a} + I(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a})$, where we denote $i = \mathscr{L}_\mathrm{I}^\mathrm{r}$, $ii = \mathscr{L}_\mathrm{I}^\mathrm{a}$, and $A = I(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a})$.

Minimizing the term 1 leads to an increased log-likelihood for both $p(\boldsymbol{x}_\mathrm{r}|\boldsymbol{z}_\mathrm{r})$ and $p(\boldsymbol{x}_\mathrm{a}|\boldsymbol{z}_\mathrm{a})$, which is applicable to both the raw and augmented data perspectives. Reducing the inference loss, represented as $\mathscr{L}_\mathrm{I}^\mathrm{r}$, $\mathscr{L}_\mathrm{I}^\mathrm{a}$ within the section labeled as 2, contributes to a more coherent latent space that facilitates the reconstruction process. Moreover, enhancing the mutual information, denoted as $I(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a})$, serves to bridge the disparity between the raw and augmented models. This process ensures a cohesive framework for incorporating data augmentation into the generative model. In conclusion, the proposed objective for learning is to approximate the joint distribution $p(\boldsymbol{x}_\mathrm{r}, \boldsymbol{x}_\mathrm{a})$ in an augmentation-informed generative modeling context, denoted as:

$$
\begin{aligned}
\mathscr{L}_\mathrm{AVAE} =&1 + A - i - ii\\
=&\mathscr{L}_\mathrm{ELBO}^\mathrm{r} + \mathscr{L}_\mathrm{ELBO}^\mathrm{a} + I(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a}),
\end{aligned}
\tag{5.6}
$$

where $\mathscr{L}_\mathrm{AVAE}$ represents the augmentation based VAE loss.

5.2.3 Deep and Shallow Learning in Mutual Information Approximation

5.2.3.1 MI approximation in shallow learning

We employ a $\mathscr{L}_\mathrm{infoNCE}$ (Noise Contrastive Estimation [112]) loss to approximate the lower bound of MI. Since this method uses a non-parametric variational distribution for VI, it can be considered as a form of shallow learning. When the variational distribution $q(\boldsymbol{z}_\mathrm{r}|\boldsymbol{z}_\mathrm{a})$ is employed to approximate the intractable posterior distribution $p(\boldsymbol{z}_\mathrm{r}|\boldsymbol{z}_\mathrm{a})$, as in Eq. (5.7a), we can derive a lower bound as shown in Eq. (5.7b). Specifically, by using an energy-based variational function $q(\boldsymbol{z}_{\mathrm{r}}|\boldsymbol{z}_{\mathrm{a}}) = \frac{p(\boldsymbol{z}_{\mathrm{r}})}{a(\boldsymbol{z}_{\mathrm{a}})} e^{f(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}$, where $f(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})$ is a critic value function, and $a(\boldsymbol{z}_{\mathrm{a}}) = \mathbb{E}_{p(x)}\left[e^{f(x,y)}\right]$ for approximation, we use the convexity of the log function to apply Jensen's inequality to $\mathbb{E}_{p(\boldsymbol{z}_{\mathrm{a}})}[\log a(\boldsymbol{z}_{\mathrm{a}})]$ to further derive a lower bound, as in Eq. (5.7c). By utilizing the inequality: $\log(\boldsymbol{z}) \leq \frac{\boldsymbol{z}}{\tau} + \log(\tau) - 1$, we can further approximate another lower bound, as in Eq. (5.7d). Using K samples for an unbiased estimate, we obtain Eq. (5.7e), we can approximate it to the infoNCE loss Monte Carlo estimation, that is, $\mathscr{L}_{\mathrm{infoNCE}}$ in Eq. (5.7f):

$$
\begin{align*}
& I(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}}) \\
\tag{5.7a} =& \mathbb{E}_{p(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}\left[\log \frac{q(\boldsymbol{z}_{\mathrm{r}} \mid \boldsymbol{z}_{\mathrm{a}}) p(\boldsymbol{z}_{\mathrm{r}}|\boldsymbol{z}_{\mathrm{a}})}{p(\boldsymbol{z}_{\mathrm{r}}) q(\boldsymbol{z}_{\mathrm{r}} \mid \boldsymbol{z}_{\mathrm{a}})}\right] \\
=& \mathbb{E}_{p(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}\left[\log \frac{q(\boldsymbol{z}_{\mathrm{r}} \mid \boldsymbol{z}_{\mathrm{a}})}{p(\boldsymbol{z}_{\mathrm{r}})}\right] \\
& + \mathbb{E}_{p(\boldsymbol{z}_{\mathrm{a}})}[D_{\mathrm{KL}}(p(\boldsymbol{z}_{\mathrm{r}} \mid \boldsymbol{z}_{\mathrm{a}}) \| q(\boldsymbol{z}_{\mathrm{r}} \mid \boldsymbol{z}_{\mathrm{a}}))] \\
\tag{5.7b} \geq& \mathbb{E}_{p(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}[\log q(\boldsymbol{z}_{\mathrm{r}} \mid \boldsymbol{z}_{\mathrm{a}})] \\
\geq& \mathbb{E}_{p(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}[f(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})] \\
\tag{5.7c} & - \mathbb{E}_{p(\boldsymbol{z}_{\mathrm{a}})}\left[\frac{\mathbb{E}_{p(\boldsymbol{z}_{\mathrm{r}})}\left[e^{f(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}\right]}{a(\boldsymbol{z}_{\mathrm{a}})} + \log(a(\boldsymbol{z}_{\mathrm{a}})) - 1\right] \\
\geq& 1 - \mathbb{E}_{p\left(\boldsymbol{z}_{(\mathrm{r},1:K)}\right)p(\boldsymbol{z}_{\mathrm{a}})}\left[\frac{e^{f\left(\boldsymbol{z}_{(\mathrm{r},1)},\boldsymbol{z}_{\mathrm{a}}\right)}}{a\left(\boldsymbol{z}_{\mathrm{a}};\boldsymbol{z}_{(\mathrm{r},1:K)}\right)}\right] \\
\tag{5.7d} & + \mathbb{E}_{p\left(\boldsymbol{z}_{(\mathrm{r},1:K)}\right)p\left(\boldsymbol{z}_{\mathrm{a}}|\boldsymbol{z}_{(\mathrm{r},1)}\right)}\left[\log \frac{e^{f\left(\boldsymbol{z}_{(\mathrm{r},1)},\boldsymbol{z}_{\mathrm{a}}\right)}}{a\left(\boldsymbol{z}_{\mathrm{a}},\boldsymbol{z}_{(\mathrm{r},1:K)}\right)}\right] \\
\tag{5.7e} \geq& \mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K} \log \frac{e^{f\left(\boldsymbol{z}_{(\mathrm{r},i)},\boldsymbol{z}_{(\mathrm{a},i)}\right)}}{\frac{1}{K}\sum_{j=1}^{K} e^{f\left(\boldsymbol{z}_{(\mathrm{r},i)},\boldsymbol{z}_{(\mathrm{a},j)}\right)}}\right] \\
\tag{5.7f} \geq& \mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K} \log \frac{p\left(\boldsymbol{z}_{(\mathrm{a},i)} \mid \boldsymbol{z}_{(\mathrm{r},i)}\right)}{\frac{1}{K}\sum_{j=1}^{K} p\left(\boldsymbol{z}_{(\mathrm{a},i)} \mid \boldsymbol{z}_{(\mathrm{r},j)}\right)}\right] \triangleq \mathscr{L}_{\mathrm{infoNCE}}.
\end{align*}
$$

We optimize an infoNCE loss scaled by the temperature coefficient τ:

$$
\begin{align*}
& \mathscr{L}_{\mathrm{InfoNCE}} \\
\tag{5.8} =& -\log \frac{\exp\left(\boldsymbol{z}_u^{\mathrm{r},\top} \boldsymbol{z}_u^{\mathrm{a}}/\tau\right)}{\sum_v \exp\left(\boldsymbol{z}_u^{\mathrm{r},\top} \boldsymbol{z}_v^{\mathrm{a}}/\tau\right) + \sum_{v \neq u} \exp\left(\boldsymbol{z}_u^{\mathrm{r},\top} \boldsymbol{z}_v/\tau\right)},
\end{align*}
$$

where the negative pairs are none, indicating that the negative keys for a sample are the positive keys for others.

5.2.3.2 MI approximation in deep learning

We can decompose the MI into two ratios in Eq. (5.9a) and approximate it per a density ratio trick, guided by [33]. In that case, the density ratio is approached by a parameterized neural network, and we can approximate the MI implicitly in a deep learning scheme. Specifically, instead of modeling two distributions directly, i.e., $q(\boldsymbol{z}_{\mathrm{r}}, \boldsymbol{z}_{\mathrm{a}})$ and $q(\boldsymbol{z}_{\mathrm{r}})$, we can learn the ratio $r = \frac{q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})}{q(\boldsymbol{z}_{\mathrm{r}})}$ in an adversarial manner, i.e., training a discriminator to classify whether the label comes from the target distribution $\mathscr{P}$ or not, as shown in Eq. (5.9b), where y is a preset pseudo label. Since we use the discriminator method to estimate the MI, the upper bound is denoted as Eq. (5.9c).

$$\mathbb{E}_{q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})} \frac{q\left(\boldsymbol{z}_{\mathrm{r}} | \boldsymbol{z}_{\mathrm{a}}\right)}{q\left(\boldsymbol{z}_{\mathrm{r}}\right)}$$

$$= \mathbb{E}_{q(\boldsymbol{z}_{\mathrm{r}},\boldsymbol{z}_{\mathrm{a}})} \frac{q\left(\boldsymbol{z}_{\mathrm{r}} | \boldsymbol{z}_{\mathrm{a}}\right)}{q(\boldsymbol{z}_{\mathrm{r}})} \tag{5.9a}$$

$$\leq \log \frac{\mathscr{P}(y=1 \mid \boldsymbol{z}_{\mathrm{r}})}{\mathscr{P}(y=0 \mid \boldsymbol{z}_{\mathrm{r}})} + \log \frac{\mathscr{P}(y=1 \mid \boldsymbol{z}_{\mathrm{a}})}{\mathscr{P}(y=0 \mid \boldsymbol{z}_{\mathrm{a}})} \tag{5.9b}$$

$$\leq \log \frac{\Psi(\boldsymbol{z}_{\mathrm{r}})}{1-\Psi(\boldsymbol{z}_{\mathrm{r}})} + \log \frac{\Psi_{\mathrm{a}}(\boldsymbol{z}_{\mathrm{a}})}{1-\Psi_{\mathrm{a}}(\boldsymbol{z}_{\mathrm{a}})} \triangleq \mathscr{L}_{\text{adversial}} \tag{5.9c}$$

5.2.4 End-to-End Anomaly Detection Training

This section introduces an end-to-end TSAD model based on a weakly augmented generative model.

5.2.4.1 Weakly Augmentation

In augmentation-based generative models, the likelihood fitting is enhanced by reusing training data. In VAEs, this leads to improved generative models $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ parameterized by θ via enriched data representations in the inference network $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ parameterized by ϕ. The augmented latent variable, $\boldsymbol{z}_{\mathrm{a}}$, is derived as $\boldsymbol{z}_{\mathrm{a}} \sim q(\boldsymbol{z}_{\mathrm{a}} | \boldsymbol{x})$. During data preprocessing, we can augment latent representations directly by manipulating the input data augmentation, represented as $\boldsymbol{z}_{\mathrm{a}} \sim q(\boldsymbol{z}_{\mathrm{a}} | \boldsymbol{x}_{\mathrm{a}})$. Here, the augmented input $\boldsymbol{x}_{\mathrm{a}}$ is obtained from the raw input $\boldsymbol{x}_{\mathrm{r}}$ using the augmentation operation $\mathscr{O}$, formulated as $\boldsymbol{x}_{\mathrm{a}} = \mathscr{O}(\boldsymbol{x}_{\mathrm{r}})$.

Data augmentation methods for time series data typically require an input array of size `(batch, time_steps, channel)`, with manipulation possible in the batch domain (such as jittering with noise, scaling, and normalization) or in the time domain (including window slicing and warping). Additionally, augmentations can be applied in the frequency domain. These techniques enrich the original dataset through various methods, effectively enhancing data diversity. This diversification is crucial for models to comprehend better and predict time series patterns.

Nevertheless, our findings suggest that applying weak augmentation to the original input data may yield more favourable outcomes for likelihood fitting in anomaly detection. Specifically, weak augmentation involves subtle modification of the data, primarily through different normalization techniques. These include:

- **Standardization**:

$$\boldsymbol{x}_{\mathrm{a}} = \mathcal{O}^{\mathrm{stand}}(\boldsymbol{x}_{\mathrm{r}}) = \frac{\boldsymbol{x}_{\mathrm{r}} - \mu}{\sigma}, \tag{5.10}$$

μ is the mean and σ as the standard deviation of the data.

- **Min-Max Normalization**:

$$\boldsymbol{x}_{\mathrm{a}} = \mathcal{O}^{\mathrm{mm}}(\boldsymbol{x}_{\mathrm{r}}) = \frac{\boldsymbol{x}_{\mathrm{r}} - \min(\boldsymbol{x}_{\mathrm{r}})}{\max(\boldsymbol{x}_{\mathrm{r}}) - \min(\boldsymbol{x}_{\mathrm{r}})}, \tag{5.11}$$

This scales the data to a specified range, such as 0 to 1.

Such moderate adjustments typically preserve the fundamental characteristics and trends of the time series. They enable the model to discern the core attributes of the data more effectively, thereby enhancing prediction accuracy. Furthermore, weak augmentation maintains the data's authenticity, mitigating the risk of over-distorting the original data structure. This is vital for preserving both the reliability and interpretability of the model.

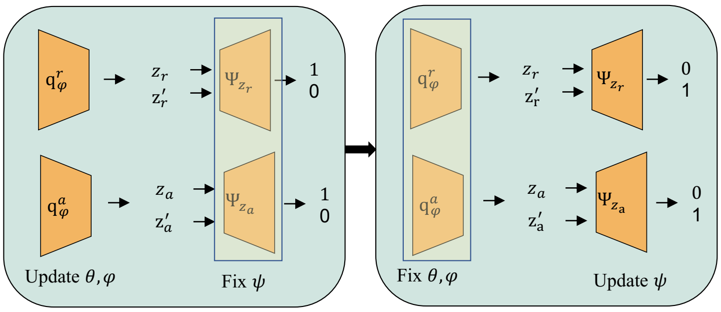


Figure 5.3: Illustration of adversarial learning in mutation information approximation. In the first stage, the discriminator is frozen to update the parameters of encoders and decoders. In the second stage, we freeze the generator's parameters and train the discriminator while simultaneously inverting the pseudo-labels of positive and negative samples to train the discriminator.

5.2.4.2 Training

For raw input $\boldsymbol{x}_\mathrm{r}$, we first obtain its augmented variable $\boldsymbol{x}_\mathrm{a}$ during the data preprocessing phase. Then, we train the inference networks for both the raw and augmented perspectives, namely $q_{\phi_\mathrm{r}}(\boldsymbol{z}_\mathrm{r} \mid \boldsymbol{x}_\mathrm{r})$ and $q_{\phi_\mathrm{a}}(\boldsymbol{z}_\mathrm{a} \mid \boldsymbol{x}_\mathrm{a})$, as well as the generative networks $p_{\theta_\mathrm{r}}(\boldsymbol{x}_\mathrm{r} \mid \boldsymbol{z}_\mathrm{r})$ and $p_{\theta_\mathrm{a}}(\boldsymbol{x}_\mathrm{a} \mid \boldsymbol{z}_\mathrm{a})$. At the same time, we optimize the inference and reconstruction losses based on Eq. (5.2.2) and Eq. (5.2.2).

During training, we employ a strategy of sharing parameters for joint likelihood consolidation of the raw data distribution $p(\boldsymbol{x}_\mathrm{r}, \boldsymbol{z}_\mathrm{r})$ and augmented data distribution $p(\boldsymbol{x}_\mathrm{a}, \boldsymbol{z}_\mathrm{a})$ to align the reconstruction effect. This allows the inference and reconstruction networks to share the same structure and parameters during training. Such an approach reduces the number of model parameters and increases the generalization of the model, enabling it to learn normal patterns of different data distributions in reconstructing normal data.

In conducting VI of the joint distribution's posterior distribution, we maximize the MI of the two latent variables to encourage the original generator and the augmented generator to produce similar data distributions. In our actual optimization objectives, i.e., the surrogated loss $\mathscr{L}_{\mathrm{WAVAE}}$, we implement two methods to control the divergence between the two likelihood distributions: the $\mathscr{L}_{\mathrm{WAVAE}}^{\mathrm{infoNCE}}$ loss based on contrastive learning:

$$
\begin{aligned}
\mathscr{L}_{\mathrm{WAVAE}}^{\mathrm{infoNCE}} &= 1 + A - i - ii \\
&= \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{r}} + \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{a}} + I(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a}) \\
&= \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{r}} + \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{a}} + \mathscr{L}_{\mathrm{infoNCE}}(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a}) \\
&:= \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{r}} + \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{a}} + \alpha \mathscr{L}_{\mathrm{infoNCE}}(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a}),
\end{aligned} \tag{5.12}
$$

and the $\mathscr{L}_{\mathrm{WAVAE}}^{\mathrm{adversial}}$ loss based on adversarial learning:

$$
\begin{aligned}
\mathscr{L}_{\mathrm{WAVAE}}^{\mathrm{adversial}} &= 1 + A - i - ii \\
&= \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{r}} + \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{a}} + I(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a}) \\
&= \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{r}} + \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{a}} + \mathscr{L}_{\mathrm{adversial}}(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a}) \\
&:= \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{r}} + \mathscr{L}_{\mathrm{ELBO}}^{\mathrm{a}} + \gamma \mathscr{L}_{\mathrm{adversial}}(\boldsymbol{z}_\mathrm{r}, \boldsymbol{z}_\mathrm{a}).
\end{aligned} \tag{5.13}
$$

The discriminator's performance is optimized in an adversarial manner, with the specific optimization process illustrated in Figure. 5.3. The first part focuses on maximizing the encoder-decoder capabilities, and the second part involves swapping pseudo-labels to maximize discriminator loss. In the training stage, we use α and γ to constrain the infoNCE loss and adversarial loss, respectively, to achieve the balance between inference and repesentation.

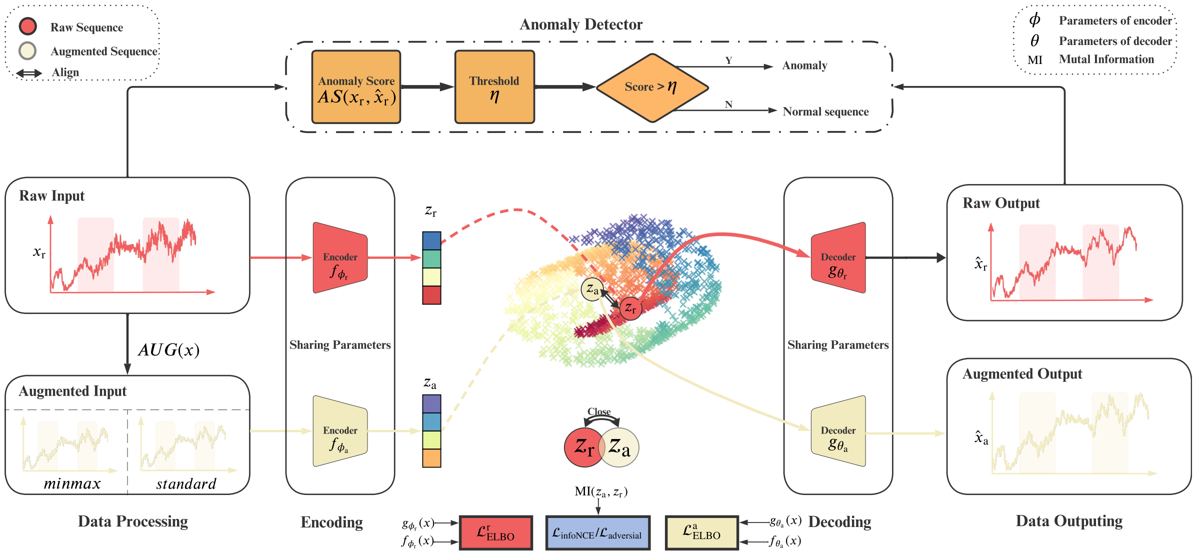


Figure 5.4: The overall framework of WAVAE, training begins with the raw data $\boldsymbol{x}_{\mathrm{r}}$ undergoing an augmentation algorithm AUG, resulting in augmented data $\boldsymbol{x}_{\mathrm{a}}$. Concurrently, we train a shared-parameter VAE separately for both sets of data. However, evaluation, i.e., anomaly detector, is conducted solely on the original model between raw input $\boldsymbol{x}_{\mathrm{r}}$ and its reconstruction $\hat{\boldsymbol{x}}_{\mathrm{r}}$, essentially designing an end-to-end anomaly detector.

5.2.4.3 Anomaly Scores

The training process and the determination of anomalies are illustrated in Figure. 5.4. The reconstruction-based anomaly detection utilizes the deviation between the original data and the reconstructed data as an anomaly score, denoted as $AS(\boldsymbol{x}, \hat{\boldsymbol{x}})$. We determine whether the input data is anomalous by comparing the anomaly score with a preset threshold η. The specific process is shown in Algorithm 6.

5.3 Experiments

5.3.1 Benchmarks

To validate the effectiveness of our approach, we select 16 reconstruction-based models for anomaly detection in time series data as benchmarks, which included 6 generative models (GANs and AEs based):

1. Transformer autoencoder (TAE) [104]: A transformer autoencoder encodes and decodes time-series data to capture temporal dependencies.

Algorithm 6 The training process of WAVAE

```
Require: Dataset $\mathscr{D} = \{B^i_{tr}, B^i_e\}^m_{i=1}$, Training batch $B_{\text{tr}} = \{(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(j)})\}^b_{j=1} \in \mathbb{R}^{b\times s\times f}$, Evaluation batch $B_{\text{e}} \in \mathbb{R}^{b\times s\times c}$.    ▷ $b, s, c$ refer to the size of batch, sequence length, and features respectively
Ensure: Parameters of encoder $f_\phi$, decoder $g_\theta$, and discriminator $\psi$, anomaly threshold $\eta$.
for each $B^i$ in the training batch $B_{\text{tr}}$ do
    $B^i_{\text{a}} \leftarrow \mathscr{O}(B^i)$    ▷ Operation $\mathscr{O}$ is defined in Eq. (5.10) and Eq. (5.11)
    $B^i_{\text{a}} \subset B_{\text{a}}$
end for
while not converged do
    for each $B^i, B^i_{\text{a}}$ in $\mathscr{D}$ do
        Compute gradients of Eq. (5.12) or Eq. (5.13) w.r.t. $\theta$ and $\phi$
        Update the parameters of $f$ and $g$
    end for
end while
for each $B^i_e$ in the evaluation batch $B_{\text{e}}$ do
    $\hat{B^j_e} \leftarrow g_\theta\big(f_\phi(B^j_e)\big)$    ▷ Reconstruct the sequence
    for each $\boldsymbol{x}^i_{\text{r}}, \boldsymbol{x}^i_{\text{a}}$ in $B_e$ do
        $Score \leftarrow AS(\boldsymbol{x}^j_{\text{r}}, \boldsymbol{x}^j_{\text{a}})$    ▷ Calculate the anomaly score based on similarity
        if $Score < \eta$ then
            $\boldsymbol{x}^i_{\text{r}}$ is an anomaly
        else
            $\boldsymbol{x}^i_{\text{r}}$ is not an anomaly
        end if
    end for
end for
```

2. The multi-scale CNN-RNN based autoencoder (MSCREA): A CNN, RNN-based autoencoder leverages CNN for representation extraction across multiple scales and RNN to capture temporal dependencies, tailored for enhancing anomaly detection in time-series data.

3. BeatGAN (BGAN) [190]: A GAN-based model for ECG anomaly detection, learning normal heartbeats to identify irregular patterns in time-series data.

4. RNN-based autoencoder (RAE) [102]: A gated recurrent unit (GRU) based autoencoder designed to encode and decode time-series data for anomaly detection by learning sequential patterns and temporal relationships.

5. CNN-based autoencoder [181]: A CNN-based autoencoder architecture tailored for

time-series analysis, utilizing convolutional layers to identify spatial patterns in data, essential for detecting anomalies in sequential datasets.

6. RandNet (RN) [25]: An ensemble of randomly structured autoencoders with adaptive sampling recognize efficiently and robustly detect outliers in data.

In addition, 10 anomaly detection methods for time-series data involve probabilistic generative models, namely Variational autoencoders, including:

1. The Gaussian mixture model variational autoencoder (GMMVAE) [93]: The VAE with GMM priors combines the probabilistic framework of Gaussian mixtures with the generative capabilities of VAEs to model complex distributions in time-series data, facilitating robust anomaly detection through learned latent representations.

2. Variational autoencoder (VAE) [164]: modeling the likelihood of generative data.

3. Recurrent neural network based VAE (RNNVAE) [116]: Merging an RNN with the variational approach to autoencoding, capturing temporal dependencies in sequential data for improved anomaly detection through stochastic latent spaces.

4. The variational RNN autoencoder (VRAE) [137]: combining the sequence modeling strengths of RNNs with the probabilistic latent space of variational autoencoders, aiming to improve anomaly detection in time-series by learning complex temporal structures.

5. α^{T}-VQRAE, β^{T}-VQRAE, and γ^{T}-VQRAE [72]: Extensions of VRAE, with RNN substituted by a quasi-recurrent network and α, β, γ-log-likelihood loss to help the VAE model achieve robust representation.

6. α-biVQRAE, β-biVQRAE, and γ-biVQRAE [72]: Variants of VQRAE, with RNN extended to bilevel to achieve time dependence on time-series data, while the α, β, γ-loglikelihood loss helps the VAE-based model achieve robust representation.

5.3.2 Experimental Setup

Datasets: To validate the effectiveness of our proposed methodology, we undertake a series of experiments on a quartet of MTS datasets below. These datasets, encompassing several hundred temporal sequences, are sourced from real-world industrial systems or are synthetically generated, comprehensively evaluating the modeling performance.

The Genesis demonstrator dataset for machine learning (GD)[1] comprises five distinct sequences, encapsulating continuous or discrete signals recorded from portable pick-and-place robots at millisecond intervals. We harness the sequence replete with anomalies to target anomaly detection, specifically the `Genesis_AnomalyLabels.csv`, which consists of 16,220 records. Within this framework, records marked with class 0 are designated as normal, whereas the remaining classifications, classes 1 and 2, are delineated as anomalies.

The high storage system data for energy optimization (HSS)[2] dataset is composed of four sequences documenting the readings from induction sensors situated on conveyor belts.Within each sequence, records tagged with class 0 are categorized as normal, whereas those labelled with class 1 are identified as anomalous.

The electrocardiogram dataset (ECG)[3] comprises a solitary time-series sequence collected from PhysioNet signals attributed to a patient with severe congestive heart failure. For the sake of consistent comparison, we follow the guidelines in [32], utilizing `ECG5000_TRAIN.tsv` from the training dataset of anomaly detection. This approach involves classifying three classes (supraventricular ectopic beats, PVC, and unclassifiable events) as anomalies, while the two remaining classes (R-on-T premature ventricular contraction, normal) are maintained as the normal data.

The trajectory dataset (TD)[4] encapsulates a unique time-series sequence, with each data point being two-dimensional. These points represent the detection algorithm's accuracy in delineating the skeletal structure of a hand, coupled with assessments from three human evaluators on the algorithm's predictive accuracy. We undertake an unsupervised anomaly detection task in this setting using the `HandOutlines_TRAIN.tsv` file extracted from the training set, comprising 1,000 instances. Within this dataset, instances classified as normal bear the label of class 1, and those recognized as anomalies carry the label of class 0.

Implementation details: Our experimental setup was standardized to ensure a level playing field and control for potential performance biases introduced by Pytorch Versions. The versions selected for all implementations are Python 3.7.16, PyTorch 1.1.0, NumPy 1.19.2, CUDA toolkit 10.0.130, and cuDNN 7.6.5. This approach guarantees that the proposed and comparative algorithms are evaluated under equivalent computational environments. Our hardware setup included NVIDIA Quadro RTX 6000 GPUs

[1] `https://www.kaggle.com/datasets/inIT-OWL/genesis-demonstrator-data-for-machine-learning`

[2] `https://www.kaggle.com/datasets/inIT-OWL/high-storage-system-data-for-energy-optimization`

[3] `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`

[4] `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`

with driver version 525.105.17 and CUDA version 12.0. Additionally, we incorporate a randomness control module, employing seed values to govern the stochasticity across computational units such as GPU, Python, and PyTorch.

5.3.3 Robust Detection Performance by Expressive Posteriors

The anomaly detection performance on five public datasets can be found in Tables 5.1 and 5.2. With `PRAUC` and `ROCAUC` metrics, our method outperforms the baselines across all datasets, regardless of whether they are AE and GAN-based generative models or VAE-based ones. Note that our comparative data is originated from [72]. Additionally, we observe that the methods based on adversarial mechanisms generally underperform than those using contrastive loss.

Table 5.1: Overall accuracy, PR-AUC. For each dataset, the three best-performing methods are denoted using distinct markings: **bold** for the top method, superscript asterisk* for the second-best, and $\underline{\text{underline}}$ for the third-best.

Models/Datasets	**GD**	**HSS**	**ECG**	**TD**
TAE	0.088	0.195	0.138	0.175
MSCREA	0.075	0.161	0.105	0.148
BGAN	0.109	0.214	0.103	0.151
RAE	0.128	0.242	0.118	0.163
CAE	0.116	0.207	0.107	0.177
RN	0.112	0.146	0.105	0.168
GMMVAE	0.142	0.216	0.163	0.364
VAE	0.097	0.203	0.131	0.188
RNNVAE	0.086	0.204	0.079	0.118
VRAE	0.131	0.219	0.144	0.165
α^{T}**-VQRAE**	0.235	$\underline{0.225}$	0.177	0.428
β^{T}**-VQRAE**	0.242	0.223	0.177	0.427
γ^{T}**-VQRAE**	0.245	0.222	0.184	0.423
α**-biVQRAE**	0.249	0.227*	0.141	0.429
β**-biVQRAE**	$\underline{0.256}$	0.224	0.189*	$\underline{0.430}$
γ**-biVQRAE**	0.258*	0.222	$\underline{0.186}$	0.432
WAVQRAE-Adverisal	**0.304**	**0.286**	**0.190**	**0.440**
WAVQRAE-Contrast	**0.307**	**0.358**	**0.200**	**0.504**

Table 5.2: Overall accuracy, ROC-AUC.

Models/Datasets	**GD**	**HSS**	**ECG**	**TD**
TAE	0.652	0.563*	0.542	0.531
MSCREA	0.582	0.509	0.509	0.519
BGAN	0.673	0.549	0.547	0.622
RAE	0.608	0.537	0.552	0.593
CAE	0.641	0.560	0.574	0.583
RN	0.731	0.526	0.524	0.533
GMMVAE	0.763	0.534	0.533	0.531
VAE	0.664	0.525	0.531	0.643
RNNVAE	0.595	0.516	0.536	0.574
VRAE	0.658	0.521	0.551	0.662*
α^{T}**-VQRAE**	0.970	0.529	0.592	0.539
β^{T}**-VQRAE**	0.968	0.520	0.583	0.535
γ^{T}**-VQRAE**	0.969	0.524	0.598	0.547
α**-biVQRAE**	0.975	0.538	0.597	0.542
β**-biVQRAE**	0.976	0.527	0.603*	0.546
γ**-biVQRAE**	0.978*	0.526	0.601	0.549
WAVQRAE-Adverisal	**0.991**	**0.563**	**0.612**	**0.579**
WAVQRAE-Contrast	**0.996**	**0.575**	**0.630**	**0.646**

5.3.4 Sensitivity Analysis

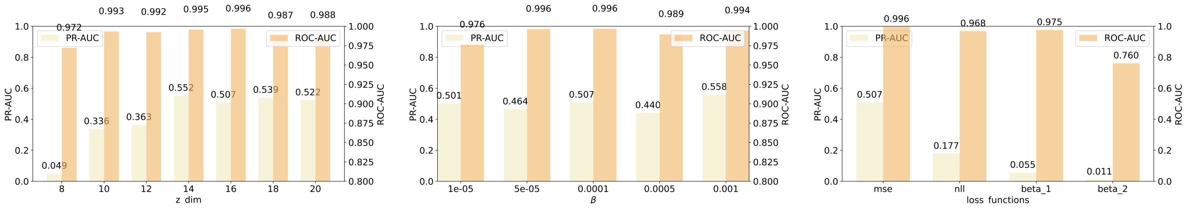
An extensive series of ablation experiments rigorously assess the sensitivity of hyperparameters in our model. This comprehensive evaluation encompasses many hyperparameter sets throughout the entire end-to-end training process.

- We investigate variations in VAE-related hyperparameters such as the β in Eq. (5.2.2) and Eq. (5.2.2) to balance the inference and reconstruction in VAE training, the dimension of $\boldsymbol{z}$, and the reconstruction loss $\mathscr{L}_{\mathrm{R}}$.

- We also scrutinize SSL-related hyperparameters like the number of discriminator layers, the weight of infoVAE loss, and the augmentation method.

- In addition, we delve into hyperparameters pertinent to time-series processing, including the sequence length and hidden variables in embeddings.

- Lastly, we explore adjustments in deep learning hyperparameters, including batch size, learning rates, and the number of epochs.

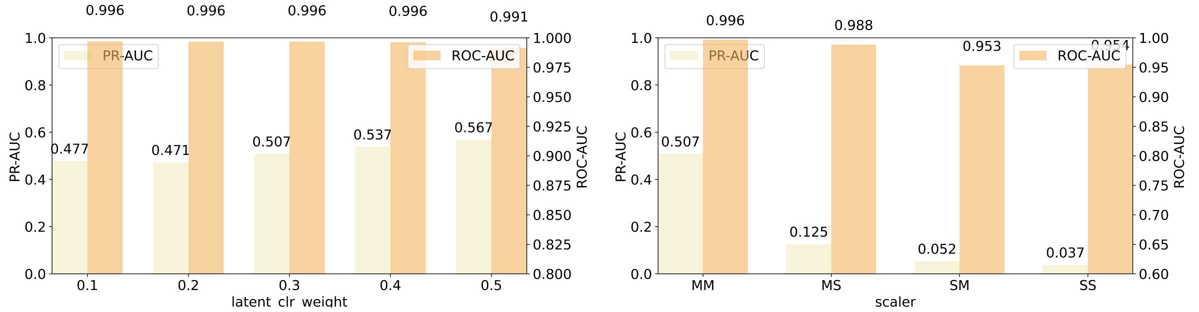


(a) Dimensions of z − space

(b) KL Divergence Weight β

(c) Reconstruction Loss Function

Figure 5.5: Sensitivity analysis of VAE-related hyperparameters indicates significant findings: (a) reveals that the dimension of $\boldsymbol{z}$ profoundly influences outcomes, with optimal performance when the dimensionality ranges between 14 and 20. (b) shows that β exerts a minimal effect on optimization, peaking in efficacy at 0.001. (c) demonstrates the superior performance of the MSE loss function.



(a) Latent CLR weight

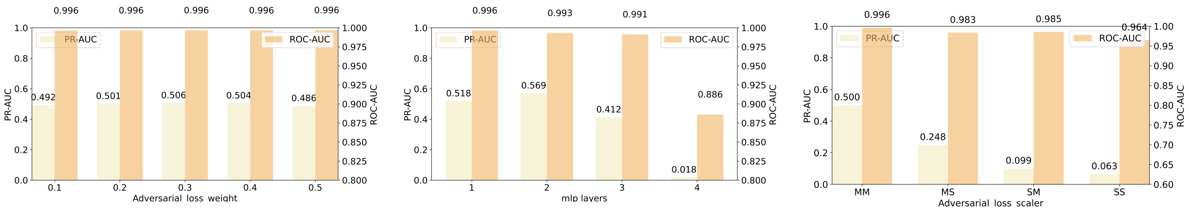
(b) NCE Loss Scaler

Figure 5.6: Sensitivity analysis of infoNCE loss related hyperparameters. From (a), it is observed that the weight of the infoNCE Loss has a minimal impact on the overall effectiveness, and a weight around 0.5 can achieve the best performance. (b) illustrates varying augmentation approaches, indicating that using the min-max normalization (`MinMax`) on both original and augmented data is the most effective.

In each set of experiments, we systematically vary a selected hyperparameter within its feasible range while maintaining the default settings for all other hyperparameters to isolate and understand the individual impact of each hyperparameter adjustment on the modeling performance.

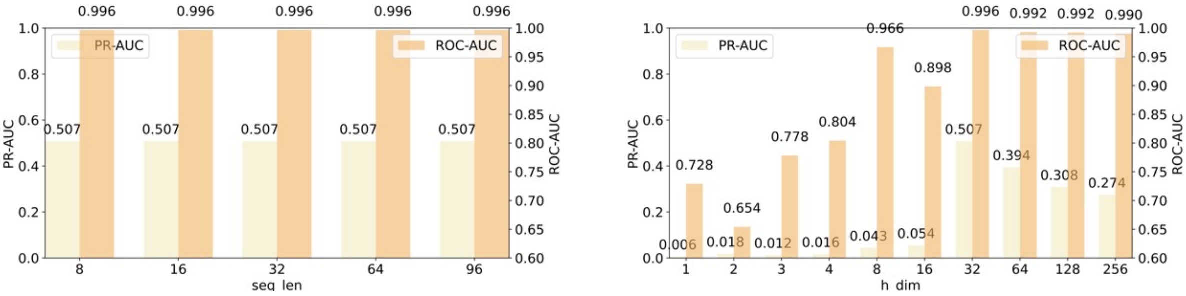
5.3.4.1 Effect of VAEs

Ideally, VAE is adept at modelling data distributions, encapsulating the potential to fit the likelihood of diverse data modalities through its sophisticated encoder-decoder architecture rooted in deep neural networks. Concurrently, it postulates a manifold-based, low-dimensional, continuous and smooth space. However, in real-world applications, the efficacy of a VAE-based data likelihood estimation is subject to substantial variability, influenced by the selection of encoder-decoder architectures, the diversity of



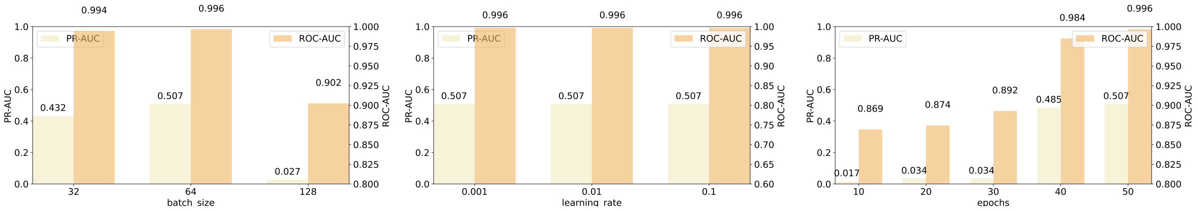
(a) Adversarial Loss Weight (b) MLP Layers (c) Adversarial Loss Scaler

Figure 5.7: Sensitivity analysis of adversarial loss related hyperparameters. From (a), it is observed that the weight of the infoNCE Loss has a minimal impact on the overall effectiveness. Conversely, (b) indicates that the number of layers in the discriminator significantly affects the results, with the best performance observed between 2 and 3 layers. (c) illustrates varying augmentation approaches, indicating that using the min-max normalization (`MinMax`) on both original and augmented data is the most effective.



(a) Sequence Length (b) Dimensions of h − space

Figure 5.8: Sensitivity analysis of sequence-related hyperparameters. (a) indicates that the model's anomaly detection performance is not affected by the length of the series. (b) shows that the encoding network achieves the best performance when the hidden state size is 32.



(a) Batch Sizes (b) Learning Rates (c) Epochs

Figure 5.9: Sensitivity analysis of deep learning related hyperparameters. (a) A batch size of 64 yields optimal results. (b) The learning rate has minimal impact on the model. (c) The best performance is observed at 50 epochs.

data modalities, and the specificity of the task at hand. To isolate and assess the effects of these factors on anomaly detection performance, we embark on a systematic sensitivity analysis of hyperparameters spanning three pivotal domains: weight of KL divergence β, dimensions of latent variables $\boldsymbol{d}$, and reconstruction loss function. Through this methodical examination, we aim to elucidate the impact of these variables on the VAE's reconstruction proficiency, thereby enhancing the model's suitability for anomaly detection endeavours.

Dimensions of Latent variables: The dimensionality of latent variables determines the amount of information the encoder compresses to maximize the log likelihood under the information bottleneck theory. Simultaneously, it influences the dependency and causality of low-dimensional space representations under the manifold assumption. The fundamental assumptions of generative models-based time-series anomaly detection posit that anomaly data will deviate from the likelihood of normal data. We conduct experiments with varying dimensions of latent variables $\boldsymbol{z}$ to develop a robust likelihood function, specifically exploring $\{8,10,12,14,16,18,20\}$. Figure. 5.5 (a) shows the outcomes and an in-depth analysis of these experiments, indicating the optimal dimension for $\boldsymbol{z}$.

KL Divergence Weight: The KL weight controls the balance between representation learning and reconstruction in the VAE model and the information during the compression process, affecting the model's robustness during training. We use the hyperparameter β to adjust the VAE's compression capability. We select five distinct values for the KL term to assess their impact, specifically $\{1e-5, 5e-5, 1e-4, 5e-4, 1e-3\}$. Detailed results and analysis of this exploration are presented in Figure. 5.5 (b).

Reconstruction Loss Function:

In Eq. (5.2.2) and Eq. (5.2.2), we fit different likelihood distributions by optimizing the specific reconstruction loss. For discrete data, we optimize the binary cross entropy (BCE) loss, i.e., $\mathscr{L}_{\mathrm{R}}^{\mathrm{BCE}}$, to fit the log-likelihood of a multivariate Bernoulli distribution, denoted as:

$$\begin{aligned}
&\mathscr{L}_{\mathrm{R}}^{\mathrm{BCE}} \\
&= E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})] \\
&= E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{d=1}^{D} \boldsymbol{x}_d \log \lambda_{\theta,d}(\boldsymbol{z}) + (1-\boldsymbol{x}_d)\log\left(1-\lambda_{\theta,d}(\boldsymbol{z})\right)\right],
\end{aligned} \tag{5.14}$$

Where $\boldsymbol{x} \in \{0,1\}^D$ and $\lambda \in \{0,1\}^D$ are the parameters of univariate Bernoulli distributions. For continuous data, we optimize the mean square error (MSE) $\mathscr{L}_{\mathrm{R}}^{\mathrm{MSE}}$ to fit the log-likelihood of a multivariate Gaussian distribution, denoted:

$$\begin{aligned}\mathscr{L}_{\mathrm{R}}^{\mathrm{MSE}} &= E_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x} \mid \boldsymbol{z})] \\ &= \frac{1}{D}\sum_{d=1}^{D} ||\boldsymbol{x}_d - \hat{\boldsymbol{x}}_d||^2.\end{aligned} \tag{5.15}$$

We also test two robust variants [50] based on the Bernoulli likelihood distribution:

$$\begin{aligned}&\mathscr{L}_{\mathrm{R}}^{\mathrm{robust1}} \\ &= \frac{\alpha_1 + 1}{\alpha_1}\left(\prod_{d=1}^{D}\left(\boldsymbol{x}_d \hat{\boldsymbol{x}}_d^{\alpha_1} + (1-\boldsymbol{x}_d)(1-\hat{\boldsymbol{x}}_d)^{\alpha_1}\right) - 1\right),\end{aligned} \tag{5.16}$$

and Gaussian likelihood distribution:

$$\begin{aligned}&\mathscr{L}_{\mathrm{R}}^{\mathrm{robust2}} \\ &= \frac{\alpha_2 + 1}{\alpha_2}\left(\frac{1}{\left(2\pi\sigma^2\right)^{\alpha_2 D/2}}\exp\left(-\frac{\alpha_2}{2\sigma^2}\sum_{d=1}^{D} \|\hat{\boldsymbol{x}}_d - \boldsymbol{x}_d\|^2\right) - 1\right),\end{aligned} \tag{5.17}$$

where α_1, α_1 are the hyperparameters and σ is the variance. The analysis and comparison of four types of loss functions are illustrated in Figure. 5.5 (c).

5.3.4.2 Effect of SSL Loss

The SSL Loss in Eq. (5.12) and Eq. (5.13) will be biased by the approximation methods and augmentation types.

InfoNCE loss: Our study investigates the infoNCE loss in contrastive learning for mutual information maximization with weight hyperparameters $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, detailed in Figure. 5.6 (a), different scaler, shown in Figure. 5.6 (b).

Adversarial loss: Our study investigates the adversarial learning discriminator loss for mutual information maximization with weight hyperparameters $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, detailed in Figure. 5.7 (a), varying layers [1,2,3,4], analyzed in Figure. 5.7 (b), different scaler, shown in Figure. 5.7 (c).

Augmentation Methods:

For self-supervised methods applied to time-series, augmentation can mine the intrinsic characteristics of the data, addressing the issue of insufficient data for deep models. To validate the effectiveness of our approach, we experiment with various strong augmentations to enhance the time dependencies and frequency domain representations of time-series data. In parallel, we also explore several weak augmentations, specifically normalization techniques applied to time-series data. Our findings indicate that the domains transformed by strong augmentations are ill-suited for generating robust likelihood functions, leading to suboptimal results in anomaly detection. In that case, We conduct sensitivity analysis experiments by testing two combinations of weak augmentations. These combinations include both raw and augmented data using `MinMax` (Figure. 5.6 (b), Figure. 5.7 (c) `MM`), raw data with `MinMax` and augmented data with `Standard` (Figure. 5.6 (b), Figure. 5.7 (c) `MS`), raw data with `Standard` and augmented data with `MinMax` (Figure. 5.6 (b), Figure. 5.7 (c) `SM`), and both raw and augmented data using standardization (Figure. 5.6 (b), Figure. 5.7 (c) `SS`). Specific experimental results and analysis are presented in Figure. 5.6 (b), Figure. 5.7 (c).

5.3.4.3 Effect of Time Series Processing

The inherent characteristics of time series, such as the window size in a batch and the memory step length in the encoding model, can impact modeling performance.

Sequence Length: In time-series data analysis, the window length is critical as it sets the data truncation extent, essential for detecting anomalies with periodicity or spatio-temporal continuity. Furthermore, the length of the time series plays a significant role in identifying contextual anomalies. For our sensitivity analysis, we choose the time series lengths of $\{8, 16, 32, 64, 96\}$. Detailed experimental results and analyses are illustrated in Figure. 5.8 (a).

Hidden Variables: We evaluate the impact of different hidden space sizes in the embedding, experimenting with the dimensionality of $\{1, 2, 3, 4, 8, 16, 32, 64, 128, 256\}$. Detailed analysis and results are presented in Figure. 5.8 (b).

5.3.4.4 Effect of Deep Learning

In deep models, batch size, learning rates, and epochs cooperate to guide the model convergence to the optimal.

Batch Sizes: By modulating the batch size, we gain insights into the stability of gradient updates and their consequent impact on model convergence. To this end, we select batch sizes $\{32, 64, 128\}$ to ascertain their influence on the modelling performance empirically. Detailed experimental results and analyses are illustrated in Figure. 5.9 (a).

Learning Rates: The step size in gradient descent induced by the learning rate is taken during optimization and can significantly influence the modelling ability to find minima. We test the learning rates of $\{0.001, 0.01, 0.1\}$ and systematically study their effects and optimize the modelling performance. Figure. 5.9 (b) illustrates detailed experimental results and analyses.

Number of Epochs: In the context of unsupervised anomaly detection, rather than focusing on model generalization, we prioritize the impact of training duration on performance. We fix the randomness and maintain consistent hyperparameters, testing the same model's anomaly detection capabilities at epochs $\{10,20,30,40,50\}$. Figure. 5.9 (c) illustrates detailed experimental results and analyses.

5.4 Summary of This Chapter

In this chapter, we propose a Weakly Augmented VAE (WAVAE), which incorporates self-supervised learning (SSL) into VAE to augment input and better estimate the anomaly-sensitive likelihood for more robust reconstruction. Specifically, WAVAE mitigates the disruption of anomalies in the low-dimensional representation space, thus resulting in augmented latent representations for TSAD. By augmenting latent representations via enhanced training, WAVAE increases the robustness of estimating the likelihood of normal data and improves the sensitivity to anomalies. It synchronizes the training of both augmented and raw models and aligns their convergence during data likelihood optimization. This is achieved by maximizing mutual information in the Evidence Lower Bound with contrastive learning for shallow learning and a discriminator-based adversarial strategy for deep learning. WAVAE significantly advances VAE by integrating SSL into likelihood enhancement on five public synthetic and real datasets, validating the efficacy of WAVAE for TSAD, compared to state-of-the-art models and through comprehensive ablation studies.

CHAPTER 6

Parameter Representation Bayesian Flow Model

Bayesian flow networks (BFNs) have emerged as a promising generative framework for handling mixed-type data, yet they often fail to capture low-dimensional latent semantics necessary for robust representation learning. This limitation motivates **RQ4**: *How can DVGM enhance inference in complex parameter spaces for better generation?*

To address **RO4**, we introduce **ParamReL**, a framework that shifts representation learning from observation spaces to parameter spaces. By leveraging progressive encoding and mutual information maximization, ParamReL extracts latent semantics directly from intermediate parameters, enabling the model to handle diverse data formats and achieve clearer, disentangled representations. The following sections outline ParamReL's innovations and its effectiveness in capturing meaningful semantics across discrete and continuous data.

6.1 ParamReL: Learning Parameter Space Representation via Progressively Encoding Bayesian Flow Networks

This work explores a new important question: *How to learn latent semantics in parameter spaces rather than in observation spaces of mixed-type data comprising continuous, discrete, and even discretized observations*? We propose a novel unified *parameter space representation learning* framework that utilizes the parameter spaces rather than the observation spaces for mixed-type data.

Representation learning [15] aims to discover low-dimensional latent semantics from high-dimensional observations, widely applied in areas including computer vision [41, 87, 186], and data analytics [113, 142]. While the main focus has been on continuous-valued data [27, 74, 105], it is more challenging to uncover semantics in discrete [11, 28] and even discretized [120, 146] data. However, existing efforts often encounter issues like inconsistent discoveries and redundant modeling [78, 191]. Recently, Bayesian flow networks (BFNs) [54, 135, 171] emerged as a promising deep generative model. BFNs use multiple steps similar to diffusion models [62, 131] to refine parameters of an output distribution for reconstructing observations. Accordingly, BFNs offer a unified strategy to handle mixed-type data while enabling fast sampling. However, they struggle to capture low-dimensional latent semantics, raising the above open question.

Correspondingly, we propose a novel unified *Param*eter space *Re*presentation *L*earning framework, ParamReL, which leverages the multi-step generative learning of BFNs for representation learning on mixed-type data. ParamReL tackles this by performing representation learning in the parameter space to extract high-level latent semantics. The key insight lies in progressively self-encoding the intermediate parameters of BFNs, generating low-dimensional latent semantics step by step. Specifically, ParamReL adopts an architecture similar to BFNs but with two significant innovations: (1) a *self-encoder* encodes intermediate parameters into lower-dimensional latent semantics, capturing gradual semantic changes throughout the multi-step generation process; and (2) a *conditional decoder*, which conditions on latent semantics and intermediate parameters, and forms the parameters of an *output distribution* for simulating observations. Additionally, ParamReL involves *a reverse-sampling procedure* customized for tasks like image reconstruction and interpolation. Variational inference method is used in learning ParamReL, where mutual information is used to promote disentangled latent semantic learning, resulting in distinct and meaningful representations.

We evaluate ParamReL in learning meaningful high-level latent semantics from both discrete and continuous-valued observations on benchmark data. The sampling and reverse-sampling mechanisms of ParamReL successfully perform tasks such as latent interpolation, disentanglement, time-varying conditional reconstruction, and conditional generation. Notably, the self-encoder reveals progressive semantics throughout flow steps, enabling ParamReL to generate semantics with improved clarity, while maintaining high quality of sample generation.

6.2 The ParamReL Model

Here, we explain the framework of ParamReL and its main design mechanisms.

ParamReL leverages the parameter space for representation learning by extracting low-dimensional latent semantics from high-dimensional mixed-type data. Different from BFNs in approximating data distribution $p(\mathbf{x}_0)$, ParamReL learns the joint distribution over observation $\mathbf{x}_0$ and a series of latent semantics $\{\mathbf{z}_t\}_{t=1}^{T}$, with $|\mathbf{z}_t| \ll |\mathbf{x}_0|, \forall t \in \{1,\dots,T\}$. That is, ParamReL seeks to reconstruct $\mathbf{x}_0$ while obtaining meaningful low-dimensional latent semantics $\{\mathbf{z}_t\}_{t=1}^{T}$.

Building on BFNs, ParamReL consists of four main components:

(1) *A self-encoder*, conditioning on the intermediate (posterior) parameters $\boldsymbol{\theta}_t$ to generate progressive latent semantics $\mathbf{z}_t$, described in Section 6.2.1.

(2) *A conditional decoder*, using a neural network on latent semantics $\mathbf{z}_t$ and intermediate parameters $\boldsymbol{\theta}_t$ to form the output distribution for subsequent steps, detailed in Section 6.2.2.

(3) *A sampling and reverse-sampling process*, facilitating tasks such as image reconstruction and interpolation, outlined in Section 6.2.3.

(4) *A training and testing procedure*, as discussed in Section 6.2.4, optimizing latent semantics $\mathbf{z}_t$ and ensuring effective model generalization.

Together, ParamReL forms a robust framework to capture and utilize latent semantics and to improve the performance of tasks including unconditional image generation and reconstruction.

6.2.1 Parameter Encoding through A Self-encoder

The *self-encoder*, denoted as $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t)$, progressively encodes intermediate parameters $\boldsymbol{\theta}_t$ into low-dimensional latent semantics $\mathbf{z}_t$, which facilitates representation learning from high-dimensional, mixed-type data at each step t. [14] has shown that upsampling layers from a U-Net in pre-trained diffusion models [123] may capture meaningful semantic information. Inspiring from this discovery and in training ParamReL, we adopt approaches similar to [101] to parameterize $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t)$ for more details). Through $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t)$, the intermediate parameter $\boldsymbol{\theta}_t$ effectively encodes itself into $\mathbf{z}_t$, together they form $\psi(\boldsymbol{\theta}_t, \mathbf{z}_t)$ for the output distribution.

Ideally, the latent semantics $\mathbf{z}_t$ should provide low-dimensional semantics distinct from the intermediate parameters $\boldsymbol{\theta}_t$ in BFNs but without compromising the data reconstruction process. To learn high-quality latent semantics, a smooth, learnable latent space is necessary, which is ensured by integrating the prior distribution $p(\mathbf{z}_t)$ into a robust probabilistic framework, allowing efficient sampling of $\mathbf{x}_0$. For simplicity and efficiency, we assume $p(\mathbf{z}_t)$ follows a Gaussian distribution.

$q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t)$ differs from traditional auto-encoders $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_0)$ in two key aspects:

- $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t)$ is conditioned on the intermediate parameter $\boldsymbol{\theta}_t$, rather than being conditioned on $\mathbf{x}_0$. This summarizes information from all previous steps to enable generating latent semantic $\mathbf{z}_t$ through all the T steps.

- The self-encoder generates a step-wise semantic $\mathbf{z}_t$, which is tailored to the dynamic behavior of variables over time t. This series of latent semantics $\{\mathbf{z}_t\}_{t=1}^{T}$ are expected to exhibit progressive semantic behaviors (such as gradual changes in age, smile, or skin color) throughout the generation process.

When observations $\mathbf{x}_0$ are unavailable, e.g. sample generation tasks, it is also worth noting that directly using regular auto-encoders like $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_0)$ to generate latent semantics is infeasible. They may require an additional module to generate latent semantics [118], while training such modules would introduce computational overhead. However, in their case, not using auto-encoders $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_0)$ would lead to inefficient resource use.

6.2.2 Conditional Decoder

The conditional decoder refers to the output distribution $p_O(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t, \mathbf{z}_t))$ which conditions on latent semantics $\mathbf{z}_t$ and intermediate parameter $\boldsymbol{\theta}_t$ to simulate $\mathbf{x}_t$. The condition $\psi(\boldsymbol{\theta}_t, \mathbf{z}_t)$ explicitly incorporates $\mathbf{z}_t$ as part of its conditioning mechanism. Following the settings in diffusion models [62, 131], we use the U-Net architecture with the Cross-Attention in each layer specified as

$$\text{Cross-Attention}(\boldsymbol{\theta}_t, \mathbf{z}_t) = (\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V}, \text{ where } \mathbf{Q} = \mathbf{W}^Q\boldsymbol{\theta}_t, \mathbf{K} = \mathbf{W}^K\mathbf{z}_t, \mathbf{V} = \mathbf{W}^V\mathbf{z}_t$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$ are the query, key and value weight matrix, respectively.

Since $\mathbf{z}_t$ works together with the corresponding intermediate parameter $\boldsymbol{\theta}_t$, it is expected that $\mathbf{z}_t$ aligns well with the progressively structured parameter $\boldsymbol{\theta}_t$. Lower-level

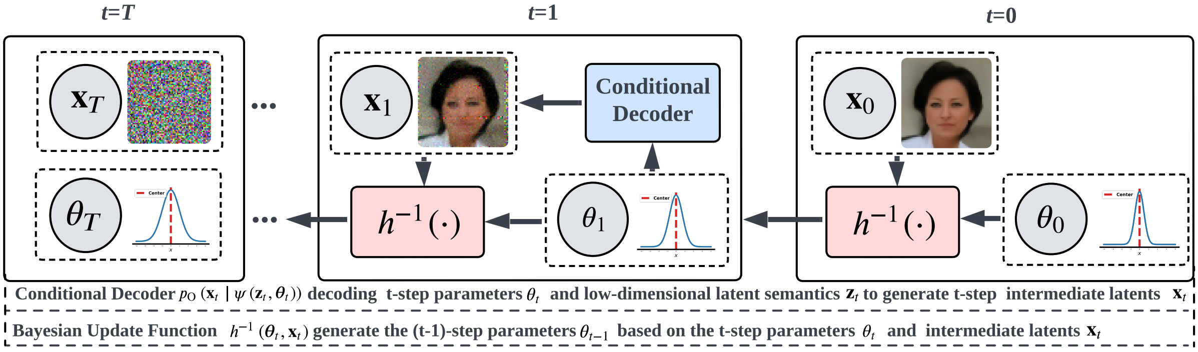

t=T
t=1
t=0
$\mathbf{x}_T$
$\mathbf{x}_1$
$\mathbf{x}_0$
Conditional Decoder
θ_T
$h^{-1}(\cdot)$
θ_1
$h^{-1}(\cdot)$
θ_0
Center
Conditional Decoder $p_O(\mathbf{x}_t \mid \psi(\mathbf{z}_t, \boldsymbol{\theta}_t))$ decoding t-step parameters θ_t and low-dimensional latent semantics $\mathbf{z}_t$ to generate t-step intermediate latents $\mathbf{x}_t$
Bayesian Update Function $h^{-1}(\boldsymbol{\theta}_t, \mathbf{x}_t)$ generate the (t-1)-step parameters θ_{t-1} based on the t-step parameters θ_t and intermediate latents $\mathbf{x}_t$

Figure 6.1: Reverse-sampling process in BFNs.

intermediate latent $\mathbf{x}_t$ (such as hair texture) is progressively incorporated. The proposed self-encoder works consistently with the conditional decoder here as both work on $\boldsymbol{\theta}_t$.

6.2.3 Sampling and Reverse-sampling Processes

After training ParamReL, the sampling and reverse-sampling processes play a crucial role in generating and reconstructing data, which is essential for tasks such as image generation and interpolation. Generating samples begins with an initial guess of the intermediate parameters $\boldsymbol{\theta}_{T+1}$. From $\boldsymbol{\theta}_{T+1}$, this sampling process sequentially generates $\mathbf{x}_T, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_0$. Specifically, given the parameter $\boldsymbol{\theta}_t$ at each step t, we have:

$$\mathbf{z}_t \sim q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t),\ \mathbf{x}_t \sim p_{\mathrm{O}}(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t, \mathbf{z}_t)),\ \boldsymbol{\theta}_{t-1} = h(\boldsymbol{\theta}_t, \mathbf{x}_t). \tag{6.1}$$

We use the trained encoder $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t)$ to replace the prior $p(\mathbf{z}_t)$ of $\mathbf{z}_t$ for improving the sampling quality. After $\boldsymbol{\theta}_0$ is obtained, a sample can be generated as $\mathbf{z}_0 \sim q_{\boldsymbol{\phi}}(\mathbf{z}_0|\boldsymbol{\theta}_0, 0), \mathbf{x}_0 \sim p_{\mathrm{O}}(\mathbf{x}_0|\psi(\boldsymbol{\theta}_0, \mathbf{z}_0))$.

However, the reverse-sampling process, which transits the observation $\mathbf{x}_0$ through the intermediate latents $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T-1}$ until $\mathbf{x}_T$, is not as straightforward as the sampling procedure. Without a clearly defined reverse-sampling process, it would be challenging to perform tasks such as image reconstruction and interpolation. In fact, by taking the inverse of the Bayesian update function $h(\cdot)$ as $\boldsymbol{\theta}_t = h^{-1}(\boldsymbol{\theta}_{t-1}, \mathbf{x}_{t-1})$, the intermediate latent $\mathbf{x}_{t-1}$ can transit to $\mathbf{x}_t$ as:

$$\boldsymbol{\theta}_t = h^{-1}(\boldsymbol{\theta}_{t-1}, \mathbf{x}_{t-1}),\ \mathbf{z}_t \sim q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t),\ \mathbf{x}_t \sim p_{\mathrm{O}}(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t, \mathbf{z}_t)). \tag{6.2}$$

Given the straightforward definition of Bayesian update function $h(\cdot)$, its inverse operation is generally easy to derive. Furthermore, this developed reverse-sampling process can be naturally extended to BFNs. Transiting $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ at time t can be performed as

$\boldsymbol{\theta}_t = h^{-1}(\boldsymbol{\theta}_{t-1}, \mathbf{x}_{t-1})$, with $\mathbf{x}_t$ sampled as $\mathbf{x}_t \sim p_{\mathrm{O}}(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t))$. With this approach, BFNs can effectively perform tasks like image reconstruction and interpolation, which were difficult or even impossible by previous BFNs models. Figure 6.1 shows the reverse-sampling process of BFNs.

6.2.4 Training and Test with ParamReL

Here, we outline the process of training and testing ParamReL by focusing on optimizing ParamReL to learn meaningful latent semantics while ensuring effective reconstruction of observations. The training process involves variational inference to approximate the joint distribution of latent variables, and a mutual information term is integrated into improving the quality of learned latent semantics by strengthening the relationship between intermediate parameters and latent semantics.

Variational Inference for Intractable Joint Distribution In ParamReL, the joint distribution over $\mathbf{x}_0$, intermediate latents $\{\mathbf{x}_t\}_{t=1}^T$ and latent semantics $\{\mathbf{z}_t\}_{t=1}^T$ can be defined as $p(\mathbf{x}_0, \{\mathbf{x}_t\}_{t=1}^T, \{\mathbf{z}_t\}_{t=1}^T|-) = p_{\mathrm{O}}(\mathbf{x}_0|\psi(\boldsymbol{\theta}_0, \mathbf{z}_0))\cdot\prod_{t=1}^T \left[p(\mathbf{z}_t)\mathbb{E}_{p_{\mathrm{O}}(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t,\mathbf{z}_t))}[p_{\mathrm{S}}(\mathbf{x}_{t-1}|\mathbf{x}_t)]\right]$, where the output distribution $p_{\mathrm{O}}(\mathbf{x}_0|\psi(\boldsymbol{\theta}_0, \mathbf{z}_0))$ at step 0 is used to model observation $\mathbf{x}_0$, and $\mathbb{E}_{p_{\mathrm{O}}(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t,\mathbf{z}_t))}[p_{\mathrm{S}}(\mathbf{x}_{t-1}|\mathbf{x}_t)]$ follows the definition of BFNs to model intermediate latent $\mathbf{x}_{t-1}$, and $p_{\mathrm{S}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is a noisy distribution of $\mathbf{x}_t$.

With $q_{\boldsymbol{\phi}}(\mathbf{z}_t\,|\,\boldsymbol{\theta}_t, t)$ defined as the encoder for $\mathbf{z}_t$ and $p_{\mathrm{S}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ defined as the variational distribution for $\mathbf{x}_{t-1}$, the evidence lower bound (ELBO) on the marginal log-likelihood of observation $\mathbf{x}_0$ is:

$$\begin{aligned}(6.3)\quad \log p(\mathbf{x}_0) \geq -\sum_{t=1}^{T} \mathbb{E}_{p_{\mathrm{F}}(\boldsymbol{\theta}_t|-)}\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t,t)}\Big\{&\mathrm{KL}\left[p_{\mathrm{S}}(\mathbf{x}_{t-1}\,|\,\mathbf{x}_0)\;\|\;\mathbb{E}_{p_{\mathrm{O}}(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t,\mathbf{z}_t))}[p_{\mathrm{S}}(\mathbf{x}_{t-1}|\mathbf{x}_t)]\right] \\ &-\mathrm{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_t\,|\,\boldsymbol{\theta}_t, t)\;\|\;p(\mathbf{z}_t)\right]\Big\} + \mathbb{E}_{p_F(\boldsymbol{\theta}_0|-)q_{\boldsymbol{\phi}}(\mathbf{z}_0|\boldsymbol{\theta}_0,0)}\left[\ln p_{\mathrm{O}}(\mathbf{x}_0|\psi(\boldsymbol{\theta}_0,\mathbf{z}_0))\right] := \mathrm{ELBO}.\end{aligned}$$

Maximizing ELBO is equivalent to performing amortized inference [76] through encoders $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t)$ and learning likelihood function through decoders [188]. When the encodable posterior $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t)$ is used to infer high-level semantics $\mathbf{z}_t$, those intermediate latents $\{\mathbf{x}_t\}_{t=1}^T$ contain low-level information in generating the observations. In ParamReL, the parameters of the output distribution are learned through iteratively proceeding the Bayesian updating functions and a learned noise model $\psi(\boldsymbol{\theta}, \mathbf{z})$ parameterized by neural networks ψ.

Mutual Information Regularization Ideally, during the training phase, we want to acquire the latent semantic $\mathbf{z}_t$ by the self-encoder $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\boldsymbol{\theta}_t, t)$ and achieve high-quality reconstruction $\widehat{\mathbf{x}_0}$ by the decoder (i.e., the output distribution $p_{\mathrm{O}}(\mathbf{x}_0|\psi(\boldsymbol{\theta}_0, \mathbf{z}_0))$). However, there exists a trade-off between inference and learning [127, 158] coherent in optimizing the ELBO in Eq. (6.3). In most cases, optimizing ELBO favours fitting likelihood rather than inference [188]. Based on the rate-distortion theory [7, 12], the rate, represented by the KL divergence term constrained by the encoders, compresses sufficient information to minimize the distortion, or reconstruction error, while simultaneously limiting the informativeness to promote a smooth latent space.

To remedy the insufficient representation learning during the inference stage, we want to increase the dependence between intermediate parameters $\boldsymbol{\theta}_t$ and latent semantics $\mathbf{z}_t$ by maximizing their mutual information $I(\boldsymbol{\theta}_t, \mathbf{z}_t)$. We can rewrite the tractable learning object in ParamReL by adding the mutual information maximization term as $\text{ELBO}_+ = \text{ELBO} + \frac{\gamma}{T}\sum_t I_q(\boldsymbol{\theta}_t; \mathbf{z}_t)$, where γ is the trade-off parameter. Considering that we cannot optimize this object directly, we can rewrite it by factorizing the rate term into mutual information and total correlation (TC).

6.3 Experiments

We present two ParamReL variants operating in different parameter spaces: ParamReLd for discrete input distributions (Section 6.3.2), and ParamReLc for continuous input distributions (Section 6.3.3), respectively. We evaluate the representation learning capabilities of ParamReL in three reconstruction-based tasks: latent interpolation, disentanglement, and time-varying conditional reconstruction. Additionally, we evaluate the model for unconditional generation, where samples are generated *only from the decoder* using a given prior.

6.3.1 Evaluation Setup

We conduct a two-fold comparison to evaluate the performance of ParamReL variants. Firstly, we compare our parameter-based models (ParamReLc and ParamReLd) with established sample-based representation learning baselines, including AE and VAE-based models such as β-VAE [61], infoVAE [188], and diffusion-based models such as DiffAE [118] and InfoDiffusion [150]. These models represent key advancements in the field: β-VAE introduce disentanglement into VAE, infoVAE incorporates MMD for balancing generation and representation, while DiffAE and InfoDiffusion explore the integration of AEs and VAEs into diffusion models to learn encodable latents and disentangled representations, respectively. Secondly, we compare the performance of ParamReLc and

ParamReLd across various input distributions for continuous and discrete data, respectively. The discrete datasets include binarized versions of MNIST (bMNIST) [38], FashionMNIST (bFashionMNIST) [160], while the continuous datasets include CelebA [97], CIFAR10 [79], and Shapes3D [19][1]. This comparison allows for a detailed examination of how different parameter space assumptions impact the representation learning of discrete and continuous data.

6.3.2 Semantic Representation of Discrete Data by ParamReLd

Here, we measure the quality of the learned latent semantics $\mathbf{z}_0$ through the downstream classification tasks. Since $\mathbf{z}_0$ locates at step 0, they should be *general* and *transferable* [48]. Various datasets by deep classifiers are assessed to ensure their universality. Specifically, following the approach in [161], we train a classifier on labeled test sets for each ParamReL model. We allocate 80% of the dataset for training a classifier and reserve the remaining 20% for test purposes. The performance on the test set is evaluated based on AUROC. This process is conducted in a 5-fold cross-validation manner, with the results reported as mean ± one standard deviation. The results are shown in Figure 6.2 (a). Higher AUROC suggests that the learned latent semantics $\mathbf{z}_0$ contain more information about data.

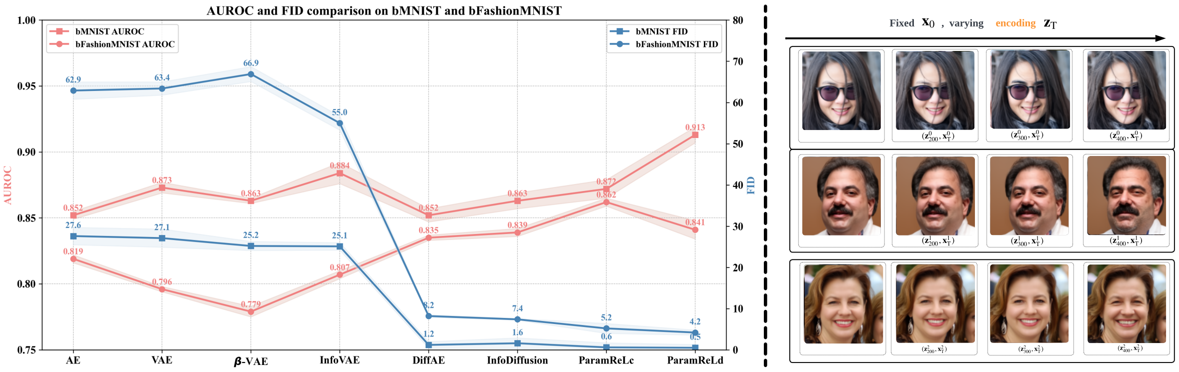
6.3.3 Semantic Representation of Continuous Data by ParamReLc

On continuous data, we evaluate ParamReLc for conditional generation, conditional reconstruction, latent interpolation, and disentanglement.

High-level Representation Learning for Conditional Generation Plot demonstrates that high-level semantic information is captured by the learned latent semantics $\{\mathbf{z}_t\}_{t=1}^T$ for image generation. This is illustrated by a set of latent-sample pairs $<\{\mathbf{z}_t^i\}_{t=1}^T, \mathbf{x}_T^{i,j}>$, where $\{\mathbf{z}_t^i\}_{t=1}^T$ are obtained by reverse-sampling from the i-th input image through the trained ParamReL, and $\mathbf{x}_T^{i,j}$ is the j-th sample from $\mathcal{N}(\mathbf{0},\mathbf{I})$ corresponding to the i-th input image. Concurrently, the low-level information, such as local attributes in images (e.g., `Narrow_Eyes`, `Mouth_Slightly_Open`, `Blond_Hair`), are determined by $\mathbf{x}_T^{i,j}$.

[1] For the discrete version, continuous data (k-bit images) can be discretized into 2^k bins by dividing the data range $[-1,1]$ into k intervals, each of length $2/k$.

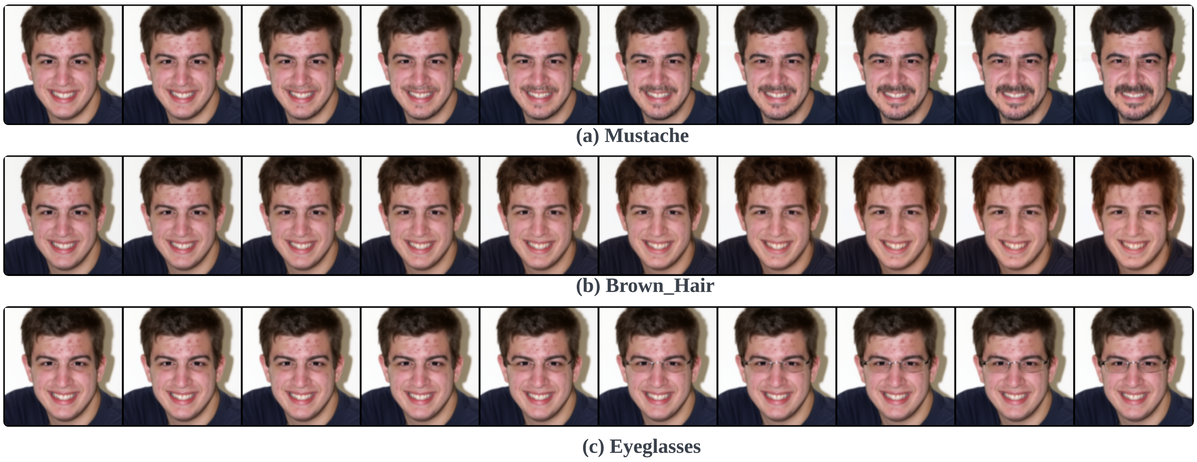
Time-varying Representation Learning for Conditional Reconstruction We design a *new* time-varying reconstruction task to evaluate the effectiveness of the progressive latent semantics learned by the self-encoder. A latent-sample pair $<\{\mathbf{z}_t^{\text{fixed}}\}_{t=1}^{T}, \mathbf{x}_T^{\text{fixed}}>$ is first obtained by apply the trained ParamReL's reverse-sampling process on an image. Then, we use the latent semantics at step t^* to replace other steps' ones and "reconstruct" the image as $\mathbf{x}_t \sim p_{\text{O}}(\mathbf{x}_t|\psi(\boldsymbol{\theta}_t, \mathbf{z}_{t^*}^{\text{fixed}})), \boldsymbol{\theta}_{t-1} = h(\boldsymbol{\theta}_t, \mathbf{x}_t), \forall t = T, \ldots, 1$. In that case, the attributes vary due to the semantics evolution encoded by time-specific latent.



(a): Comparison on discrete data by classification accuracy and generation performance. **(b): Time-varying representation learning of ParamReL**

Figure 6.2: Quantitative representation learning comparison over generative models on discrete data (a). ParamReL demonstrates competitive performance in capturing semantic information for classification, achieving approximately 0.84 AUROC for bFashionMNIST and 0.91 for bMNIST. Additionally, it shows robust generative capabilities, with FID values ranging from 0.5 to 0.6 for bMNIST and around 5 for bFashionMNIST. Among the ParamReL-based models, ParamReLd with a categorical distribution is particularly effective in modelling discrete data distributions, yielding lower FID values of 0.5 for bMNIST and 4.2 for bFashionMNIST. As shown in (b), the learned semantics exhibit progressive, time-varying changes. By varying time encodes at 200, 300, 400 time steps, more attributes will be influenced in the reconstruction stage: the `Wavy_hair`, `Brown_hair`, `Arched_Eyebrows` attributes in the first line, the `Double_Chin`, `Mustache`, `Goatee` attributes in the second line and the `Young`, `High_Cheekbones`, `Arched_Eyebrows` attributes in the third line. Notations: [AUROC, FID]; [(•, bMNIST), (■, bFashionMNIST)]; [(−, ParamReLd),(− · −, ParamReLc)].

Smooth Representation Learning for Latent Interpolation Latent space interpolation [53, 61] is commonly used to validate the smoothness, continuity, and semantic coherence of the learned latent semantics in generative models. Typically, two samples are embedded into the latent space, and interpolating between the latent variables generates interpolated representations. The reconstructed outputs produced by the sampling process reveal the semantic richness of the latent space.



(a) Mustache

(b) Brown_Hair

(c) Eyeglasses

Figure 6.3: Disentanglement of ParamReL on FFHQ-128. The interpretable traversal directions are displayed by traversing the encodings ranging from $[-3,3]$.

ParamReL achieves near-exact reconstruction, in contrast to the downgraded performance of VAE variants such as (a) vanilla VAE, and (b) β-VAE. Compared with diffusion models (c) DiffAE and (d) InfoDiffusion, ParamReL characterizes a smoother and more consistent latent space with high-quality samples.

Disentanglement We perform latent traversals on the FFHQ and CelebA datasets to evaluate the disentanglement properties of our trained ParamReL. In this process, we modify one dimension of the learned latent semantics $\{\mathbf{z}_t\}_{t=1}^{T}$ each step, and replace it with M evenly distributed numbers within a standardized range (e.g., -3 to $+3$), while keeping the other dimensions fixed. After decoding these adjusted latent semantics, we evaluate the generated samples for changes in specific attributes. Successful disentanglement is verified when manipulating one single dimension alters only one distinguishable attribute, such as age, while leaving all other attributes unchanged. ParamReL effectively isolates and controls individual data attributes in both FFHQ and CelebA. For example, on FFHQ, manipulating latent dimensions controls attributes like `Mustache`, `Brown Hair`, and `Eyeglasses`, while other attributes remain constant. Similarly, on CelebA, attributes such as `Smiling`, `Pale Skin`, and `Big Nose` are independently manipulated without affecting others.

To provide a thorough and unbiased quantitative assessment of disentanglement, we utilize two metrics: 1) Disentanglement, Completeness, and Informativeness (DCI) [43], which is a prediction-based indicator; and 2) Total AUROC Difference (TAD) [179], an intervention-based criterion. Additionally, we report the generation quality and conclude that ParamReL achieves near-exact reconstruction on CelebA. Both the qualitative latent traversal results and the quantitative disentanglement metrics show that ParamReL effectively learns disentangled representations, with visual traversals closely aligning with the attributes that the latent semantics are intended to capture.

6.4 Summary of This Chapter

In this chapter, we propose a novel unified *parameter space representation learning* framework, ParamReL, which extracts progressive latent semantics in parameter spaces of mixed-type data. In ParamReL, a *self-encoder* learns latent semantics from intermediate parameters rather than observations. A significant challenge in representation learning is to capture latent semantics in data mixing continuous, discrete, and even discretized observations (called mixed-type data), encountering issues like inconsistent discoveries and redundant modeling. Recently, Bayesian flow networks (BFNs) offer a unified strategy to represent such mixed-type data in the parameter space but cannot learn low-dimensional latent semantics since BFNs assume the size of parameters being the same as that of observations. This raises a new important question: *how to learn latent semantics in parameter spaces rather than in observation spaces of mixed-type data*? The learned semantics are then integrated into BFNs to efficiently learn unified representations of mixed-type data. Additionally, a *reverse-sampling procedure* can empower BFNs for tasks including input reconstruction and interpolation. Extensive experiments verify the effectiveness of ParamReL in learning parameter space representations for latent interpolation, disentanglement, time-varying conditional reconstruction, and conditional generation.

Chapter 7

Progressively Self-encoding Diffusion Model

Diffusion models (DMs) have demonstrated exceptional performance in generating high-quality samples, yet challenges remain in uncovering low-dimensional, aligned semantic representations throughout the generative process. This leads to **RQ5**: *How can progressive inference facilitate low-dimensional generation in diffusion models?*

To address **RO5**, we propose **ProgDiffusion**, a novel diffusion model that incorporates a self-encoding mechanism to generate dynamic, timestep-specific semantic representations. By conditioning on intermediate latents and upsampling features, ProgDiffusion aligns progressive semantic representations with latent changes over time, enabling efficient unconditional generation. The following sections detail ProgDiffusion's mechanisms and its contributions to learning aligned semantics and improving sample generation quality.

7.1 ProgDiffusion: Progressively Self-encoding Diffusion Models

Diffusion models (DMs) [39, 62, 131] have obtained tremendous successes in generating high-quality images. In general, DMs first define a noising scheme to sequentially add noises upon original observations $\mathbf{x}_0$ to obtain a sequence of noisy intermediate latents $\mathbf{x}_1,\ldots,\mathbf{x}_T$, and then learn in a reverse way to predict these $\mathbf{x}_T,\ldots,\mathbf{x}_1$ to reconstruct $\mathbf{x}_0$. While DMs' outstanding image generation capability has endorsed their capabilities in learning meaningful semantic information, these semantics are unable to be directly produced in standard DMs. The intermediate latents $\mathbf{x}_1,\ldots,\mathbf{x}_T$ have the same shape as that of the observations $\mathbf{x}_0$ as $|\mathbf{x}_0| = |\mathbf{x}_1| = \ldots = |\mathbf{x}_T|$, making it difficult to claim meaningful low-dimensional semantics are uncovered.

Existing methods in learning semantic representations in DMs may be roughly categorized into two groups: (1) Diffusion Auto-Encoder (DAE) models [30, 56, 65, 114, 118, 150, 153] encode observations $\mathbf{x}_0$ into low-dimensional semantic representations $\mathbf{z}$ and then use $\mathbf{z}$ as a condition to generate denoised intermediate latents in the reverse process. However, since observations are required in generating semantics, unconditional generation would be difficult since observations are unavailable in such tasks [1]; (2) Diffusion hyperfeature methods

[59, 88, 101, 139, 144, 166, 182, 189] which investigate the upsampling features, denoted as $\{\mathbf{u}_{t+1}\}_t$, in pre-trained DMs' U-Net architecture into various downstream tasks since these layers' dimensions are lower than the observations. However, These approaches do not propose making fundamental changes to model architectures and training methodologies. It is thus unclear the actual architectural components and techniques in learning useful semantic representations.

Unaligned and undetermined semantic representation is another issue. When the semantic representation $\mathbf{z}$ is *static* in DAE methods, $\mathbf{z}$ may not align well with the progressive behaviors of intermediate latents $\mathbf{x}_1,\ldots,\mathbf{x}_T$. For, using the same $\mathbf{z}$ for generating $\mathbf{x}_T$ and $\mathbf{x}_1$ might be questionable since $\mathbf{x}_T,\mathbf{x}_1$ should contain different levels of semantics and the former is close to white noises while the latter is more similar to the observations. In Diffusion hyperfeatures, identifying an ideal denoising timestep as well as layer number for highest predictive performance is usually non-trivial and it might need great efforts to address it.

When inefficient sampling affects the sampling generation, inconsistent training target and unaligned semantics might deteriorate learning correct representation structures. This paper proposes a Progressive self-encoded Diffusion model (ProgDiffusion) to address them. Similar to DAE, ProgDiffusion uses DDIM [131]'s forward diffusion process, and conditions on semantic representation and the previous denoising timestep's intermediate latent $\mathbf{x}_t$ to sample the current intermediate latent $\mathbf{x}_{t-1}$ in the reverse process. In contrast, instead of using a commonly used *static* encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_0)$, ProgDiffusion comprises a self-encoder $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{u}_{t+1},t)$, which conditions on the intermediate

[1] DiffAE [118] has separately trained an additional latent Denoising Diffusion Implicit Model (DDIM) to generate semantics $\mathbf{z}$. However, it requires additional training resources and adds additional uncertainties to the sample generation qualities

latent $\mathbf{x}_t$, step $(t+1)$'s upsampling features $\mathbf{u}_{t+1}$, and the denoising timestep t to generate a stepwise semantic representation $\mathbf{z}_t$. $\mathbf{z}_t$ is then used as conditions to generate $\mathbf{x}_{t-1}$ through the t-th denoising step as $\mathbf{x}_{t-1} \sim p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_t)$.

The self-encoder $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1})$ rules out the observation $\mathbf{x}_0$ required in generating semantic representations. Efficient unconditional sampling is thus achieved by interleaving the generation of progressive semantic representations through $\mathbf{z}_t \sim q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1})$ and intermediate latents through $\mathbf{x}_{t-1} \sim p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_t)$, until the final sample $\mathbf{x}_0$ is obtained. In particular, the former enables a series of progressive semantic representations to be learned, whereas the later conditions on low-level details and high-level semantics to generate the intermediate latents. As $\mathbf{x}_t$ progressively approaches the observation over the denoising step t, $\mathbf{z}_t$ also obtains clearer semantics with denoising timestep t decreases.

We use all the potential information, including $\mathbf{x}_t$ and $\mathbf{u}_{t+1}$, in forming the conditions to generate semantics as $\mathbf{z}_t \sim q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1}, t)$. The learned $\mathbf{z}_t$ may avoid the nontrivial selection of upsampling layer, and we might also obtain meaningful architectural insights in learning DMs' semantics. In particular, as $\mathbf{u}_{t+1}$ also contains certain semantics, transferring these semantics $\mathbf{u}_{t+1}$ with intermediate latent $\mathbf{x}_t$ should be easier than learning the semantics from $\mathbf{x}_t$ only.

For this, we introduce a mutual information term $I(\mathbf{x}_t, \mathbf{z}_t)$ between the intermediate latent $\mathbf{x}_t$ and semantic representation $\mathbf{z}_t$ to the objective function. Maximizing this term ensures that $\mathbf{z}_t$ stores sufficient information from $\mathbf{x}_t$. Please be noted that $I(\mathbf{x}_t, \mathbf{z}_t)$ is different from $I(\mathbf{x}_0, \mathbf{z})$ as in [150, 188], as we store information from $\mathbf{x}_t$ rather than observation $\mathbf{x}_0$. In this way, ProgDiffusion fits the intermediate latents and learns appropriate generation and amortized inference at the same time. Figure 7.5 illustrates and compares ProgDiffusion with DiffAE.

The experiments involve progressive semantic visualization tasks to understand the effect of aligned semantics. We also test ProgDiffusion on Other tasks, such as image interpolation, unconditional generation, and disentanglement, further verifying the effectiveness of our design.

The main contributions of this work include: (1) ProgDiffusion enables effective unconditional generation using a refined encoder; (2) ProgDiffusion conditions on all the intermediate information, including latent $\mathbf{x}_t$ as well as U-Net's upsampling features $\mathbf{u}_{t+1}$, to generate semantic representation $\mathbf{z}_t$; (3) ProgDiffusion learns progressive semantic representations which align well with intermediate latents; (4) ProgDiffusion integrates a stepwise mutual information term $I(\mathbf{z}_t, \mathbf{x}_t)$ to fit intermediate latents and learn effective models at the same time; (5) A new task is designed to visualize the progressive structured semantic representation learning.

Table 7.1: Comparative assessment of ProgDiffusion and diverse generative models, focusing on high-quality generation and specific representation learning capabilities.

	Generation	**Representation**			
Model	**High Quality**	**Low Dim.**	**Continuous**	**Smooth**	**Time Specific**
AE	×	✓	×	×	×
VAE	×	✓	×	×	×
GAN	×	×	×	×	×
DDPM	✓	×	✓	×	×
DDIM	✓	×	✓	✓	×
LDM	✓	✓	✓	×	×
DiffAE	✓	✓	×	✓	×
PDAE	✓	✓	×	✓	×
InfoDiff	✓	✓	✓	✓	×
DisDiff	✓	✓	×	✓	×
DiTi	✓	✓	×	✓	×
HDAE	✓	✓	×	✓	×
ProgDiffusion	✓	✓	✓	✓	✓

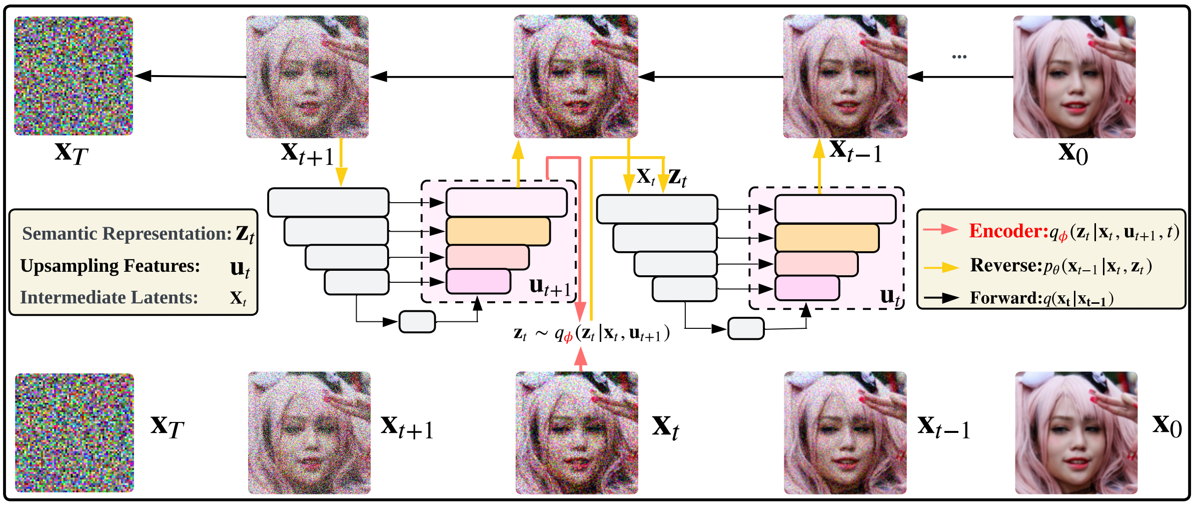

$\mathbf{x}_T$
$\mathbf{x}_{t+1}$
$\mathbf{x}_{t-1}$
...
$\mathbf{x}_0$
$\mathbf{x}_t$ $\mathbf{z}_t$
Semantic Representation: $\mathbf{z}_t$
Upsampling Features: $\mathbf{u}_t$
Intermediate Latents: $\mathbf{x}_t$
$\mathbf{u}_{t+1}$
$\mathbf{u}_t$
$\mathbf{z}_t \sim q_\phi(\mathbf{z}_t|\mathbf{x}_t,\mathbf{u}_{t+1})$
Encoder: $q_\phi(\mathbf{z}_t|\mathbf{x}_t,\mathbf{u}_{t+1},t)$
Reverse: $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{z}_t)$
Forward: $q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_T$
$\mathbf{x}_{t+1}$
$\mathbf{x}_t$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$

Figure 7.1: A detailed pipeline of training ProgDiffusion, consisting of a self-encoder and a diffusion based sample decoder. During the training phase, the noise sample $\mathbf{x}_t$ at step t is employed to predict the next step sample $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t, \mathbf{z}_t)$ by a noise prediction net parameterized by $\boldsymbol{\theta}$, which is conditioned on the induced semantic representations $\mathbf{z}_t$ sampled from $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1}, t)$.

7.2 The ProgDiffusion Model

ProgDiffusion is a DAE-like model that adopts the same diffusion process as DDIM [131], and uses an encoder to learn semantic representations and then conditions on it to generate the next intermediate latent in the reverse process.

Accordingly, we highlight a few parts of the training object in the reverse process of ProgDiffusion: (1) the self-encoder, which encodes time-specific information progressively by a hierarchical time dependent encoder framework; (2) the time specific ELBO for training ProgDiffusion; (3) the time specific mutual information regularizer. Figure. 7.1 displays the general framework of ProgDiffusion.

7.2.1 The self-encoder

ProgDiffusion generates time-specific semantic representations $\{\mathbf{z}_t\}_t$ through a self-encoder as:

$$\mathbf{z}_t \sim q_{\phi}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1}, t), \forall t = 1, \ldots, T \tag{7.1}$$

where we name $q_{\phi}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1}, t)$ since $\mathbf{x}_t$ first encodes itself into $\mathbf{z}_t$ and then works tother with $\mathbf{z}_t$ to form the denoising step.

$q_{\phi}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1}, t)$ and the commonly used encoder $q_{\phi}(\mathbf{z}|\mathbf{x}_0)$ (e.g., amortized inference [76]) have the following two major differences: (1), there are T semantic representations $\{\mathbf{z}_t\}_t$ that can track the gradual changes of semantics along with the reverse steps, whereas the semantic is *static* in the amortized inference; (2), each $\mathbf{z}_t$ is conditioned on the t-th intermediate latent $\mathbf{x}_t$, the upsampling features $\mathbf{u}_{t+1}$, and the denoising step t rather than the observation $\mathbf{x}_0$ as in existing DAE models. Therefore, the semantics $\mathbf{z}_T$ and $\mathbf{z}_1$ should be quite different since $\mathbf{x}_T$ and $\mathbf{x}_1$ are close to noises and the observations, respectively.

U-Net's upsampling features $\mathbf{u}_{t+1}$ In addition to $\mathbf{x}_t$ and the denoising timestep t, the proposed self-encoder also conditions on the upsampling features $\mathbf{u}_{t+1}$ to generate semantic representation $\mathbf{z}_t$. Recent works [59, 88, 101, 139, 144, 166, 182, 189] show that these upsampling features might contain important spatial semantic information. While those approaches focused on investigating these layers from a pre-trained diffusion models for downstream tasks, ProgDiffusion may propose making fundamental changes to model architectures and training methodologies, exploring the actual architectural components and techniques in learning useful semantic representations. Also, it might be easier in learning semantics from upsampling features and intermediate latents than from the intermediate latents only.

Since the upsampling features contain different sizes, we use an aggregation network which takes the form as $\sum_{l=1}^{L} \omega_l \cdot h(\mathbf{u}_{t+1,l})$, where L is the number of upsampling features in U-Net, $\mathbf{u}_{t+1,l}$ is the l-th upsampling features, ω_l is the corresponding mixing weight, and $h(\cdot)$ is the upsampling operation to ensure all the $\{h(\mathbf{u}_{t+1,l})\}_l$ are in the same size.

Differences between $\mathbf{z}_t$ and U-Net's bottleneck layer ProgDiffusion's time specific semantic representation $\mathbf{z}_t$ is different from the U-Net's bottleneck layer $\mathbf{b}_t$ in the following aspects: (1) although the bottleneck layer $\mathbf{b}_t$ has the lowest resolution, usually its following upsampling features contain more semantic information [159]. $\mathbf{z}_t$ has conditioned on all the upsampling features and $\mathbf{x}_t$ and may contain more semantic inforamtion; (2) $\mathbf{z}_t$ has larger influence. $\mathbf{b}_t$ is used to generate the first upsampling layer only, whereas $\mathbf{z}_t$ is a global semantics representation captured by layer-by-layer attentions among the U-Net layers.

Time specific semantic representations We expect time specific semantic representations would be more adaptive to ProgDiffusion than a static representation. E.g., The amount of semantic information contained by $\mathbf{x}_T$ and $\mathbf{x}_1$ might be proportional to the amount of observation information. Since $\{\mathbf{z}_t\}_t$ are in the same shape, it would also be quite straightforward to learn summarized semantics from a pre-trained ProgDiffusion model.

7.2.2 The Reverse Process

ProgDiffusion uses the semantic representation $\mathbf{z}_t$ and intermediate latent $\mathbf{x}_t$ as conditions in generating the next intermediate latent $\mathbf{x}_{t-1}$ and formulates the reverse step as $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_t)$. Thus, the joint probability over the intermediate latents $\mathbf{x}_{1:T}$, observations $\mathbf{x}_0$ and the semantic representations $\mathbf{z}_{1:T}$ is:

$$p(\mathbf{x}_0, \mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} [p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_t) p(\mathbf{z}_t)] \tag{7.2}$$

in which $\mathbf{z}_t$'s prior is usually defined as $p(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t|\mathbf{0}, \mathbf{I})$. A trained self-encoder $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1}, t)$ may replace the prior $p(\mathbf{z}_t)$ in the sample generation task.

7.2.3 The ELBO for Training

The training objective function can be formalized w.r.t. the evidence lower bound (ELBO) as ELBO:

$$
\begin{aligned}
(7.3)\quad \text{ELBO} = &-\mathbb{E}_{q(\mathbf{x}_t)q(\mathbf{z}_1)} \log p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{x}_1,\mathbf{z}_1) \\
&+ \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_t)} \text{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{u}_{t+1},t) \parallel p(\mathbf{z}_t)\right] + \text{KL}\left[q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)\right] \\
&+ \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t)q(\mathbf{z}_t)} \text{KL}\left[q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) \parallel p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{z}_t)\right]
\end{aligned}
$$

The major ELBO difference between DiffAE and ProgDiffusion lies that ProgDiffusion's ELBO uses all the T KL-divergences between the self-encoders and the prior distribution of each $\mathbf{z}_t$.

7.2.4 Mutual Information Regularizer

During the training phase, ProgDiffusion aims to acquire the semantic representation $\mathbf{z}_t$ through the self-encoder $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t,\mathbf{u}_{t+1},t)$ and achieve high-quality reconstruction $\widehat{\mathbf{x}}_0$ via the decoder. However, there is an inherent trade-off between correct inference and observation reconstruction, as highlighted in [127, 158], which becomes evident when optimizing $\text{ELBO}_{\text{ProgDiffusion}}$ in Eq. (7.3). Typically, ProgDiffusion prioritizes fitting the likelihood over inference [188], resulting in sub-optimal latent space representations.

To remedy the insufficient representation learning during the inference stage, we increase the dependence between the intermediate latent $\mathbf{x}_t$ and its corresponding semantic $\mathbf{z}_t$ by maximizing the mutual information (MI) $I(\mathbf{x}_t,\mathbf{z}_t)$. Hence, we rewrite the learning object in ProgDiffusion by adding the mutual information term as:

$$
\text{ELBO}_{+} = \text{ELBO} + \frac{\gamma}{T}\sum_{t} I_q(\mathbf{x}_t;\mathbf{z}_t), \tag{7.4}
$$

where $I_q(\mathbf{x}_t;\mathbf{z}_t)$ is the mutual information between $\mathbf{x}_t,\mathbf{z}_t$ under distribution $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x},t)$. As we cannot optimize this objective directly, we rewrite it by factorizing the rate term into MI and total correlation (TC) $\text{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_t) \parallel p(\mathbf{z}_t)\right]$:

$$
\begin{aligned}
(7.5)\quad \text{ELBO}_{+} = &\sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t)q(\mathbf{z}_t)} \text{KL}\left[q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) \parallel p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{z}_t)\right] \\
&+ \frac{1-\gamma}{T}\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_t)} \text{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t,t) \parallel p(\mathbf{z}_t)\right] + \frac{\gamma+\lambda-1}{T} \text{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_t) \parallel p(\mathbf{z}_t)\right] \\
&+ \text{KL}\left[q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)\right] - \mathbb{E}_{q(\mathbf{x}_t)q(\mathbf{z}_1)} \log p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{x}_1,\mathbf{z}_1)
\end{aligned}
$$

Mutual Information Learning. Unlike the rest of the terms in Eq. (7.5) that can be optimized directly using reparameterization tricks, the TC term cannot be directly optimized due to the intractable marginal distribution $q_\phi(\mathbf{z}_t)$. Here, we follow the approach in [188] to replace the TC term with any strict divergence D, $D\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right) = 0$ iff $q_\phi(\mathbf{z}) = p(\mathbf{z})$. We implement the maximum-mean discrepancy (MMD) [188] from the divergence family. MMD is a statistical measure to quantify the difference between two probability distributions, which compares their mean embeddings in a high-dimensional feature space. By defining the kernel function $\kappa(\cdot,\cdot)$, D_{MMD} is denoted as:

$$D_{\mathrm{MMD}}\left(q(\mathbf{z})\|p(\mathbf{z})\right) = \mathbb{E}_{p(\mathbf{z}),p(\mathbf{z}')}\left[\kappa\left(\mathbf{z},\mathbf{z}'\right)\right] - 2\mathbb{E}_{q(\mathbf{z}),p(\mathbf{z}')}\left[\kappa\left(\mathbf{z},\mathbf{z}'\right)\right] + \mathbb{E}_{q(\mathbf{z}),q(\mathbf{z}')}\left[\kappa\left(\mathbf{z},\mathbf{z}'\right)\right] \tag{7.6}$$

7.3 Implementing ProgDiffusion

In optimizing the mutual information regularized time specific ELBO, w.r.t. Eq. (7.5), we modify the DiffAE framework and propose ProgDiffusion network illustrated in Figure 7.1.

7.3.1 Implemention of ProgDiffusion

Based on DiffAE's implementation, implementing ProgDiffusion is straightforward. In particular, we may simply substitute the time-independent encoder $q_\phi(\mathbf{z}|\mathbf{x}_0)$ by a self-encoder $q_\phi(\mathbf{z}_t|\mathbf{x}_t,\mathbf{u}_{t+1},t)$, from

```
# model/unet_autoenc.py
class BeatGANsAutoencModel(BeatGANsUNetModel):
    def forward(self, x, t, x_start):
        tmp = self.encode(x_start)
```

to

```
# model/unet_autoenc.py
class BeatGANsAutoencModel(BeatGANsUNetModel):
    def forward(self, x, t, x_start):
        # x = x_t = q_sample(x_start, t)
        ut1 = self.extract(x,t)# extract Upsampling Features in former
        ↪ time steps
        tmp = self.encode(x,,ut1,t)
```

7.3.2 Unconditional Generation

We use the trained self-encoder $q_\phi(\mathbf{z}_t|\mathbf{x}_t,\mathbf{u}_{t+1},t)$, replacing $\mathbf{z}_t$'s prior distribution $p(\mathbf{z}_t)$, to accomplish the unconditional generation task in ProgDiffusion. Accordingly, a sample can be generated as in Algorithm 7. By progressively encoding the intermediate latent $\mathbf{x}_t$ into a time-specific semantic representation $\mathbf{z}_t$, the resultant semantic information is introduced in generating samples.

Algorithm 7 Unconditional generation process of ProgDiffusion

```
Input: number of steps $T$, noise level $\sigma_1$, parameters $\phi$ and $\theta$
Output: $\mathbf{x}_0$
Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T;\mathbf{0},\mathbf{I})$, set $\mathbf{u}_{T+1} = \emptyset$
for $t = T$ to 1 do
    Sample $\mathbf{z}_t \sim q_\phi(\mathbf{z}_t|\mathbf{x}_t,\mathbf{u}_{t+1},t)$
    Sample $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{z}_t)$
end for
Return $\mathbf{x}_0$
```

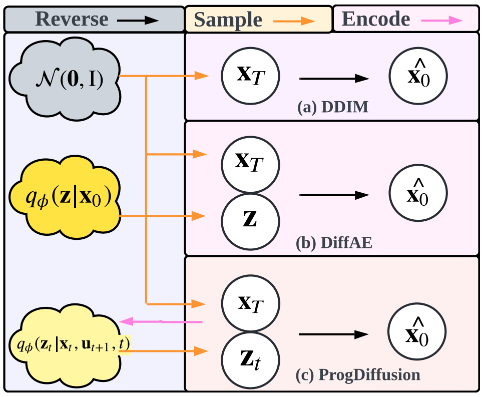
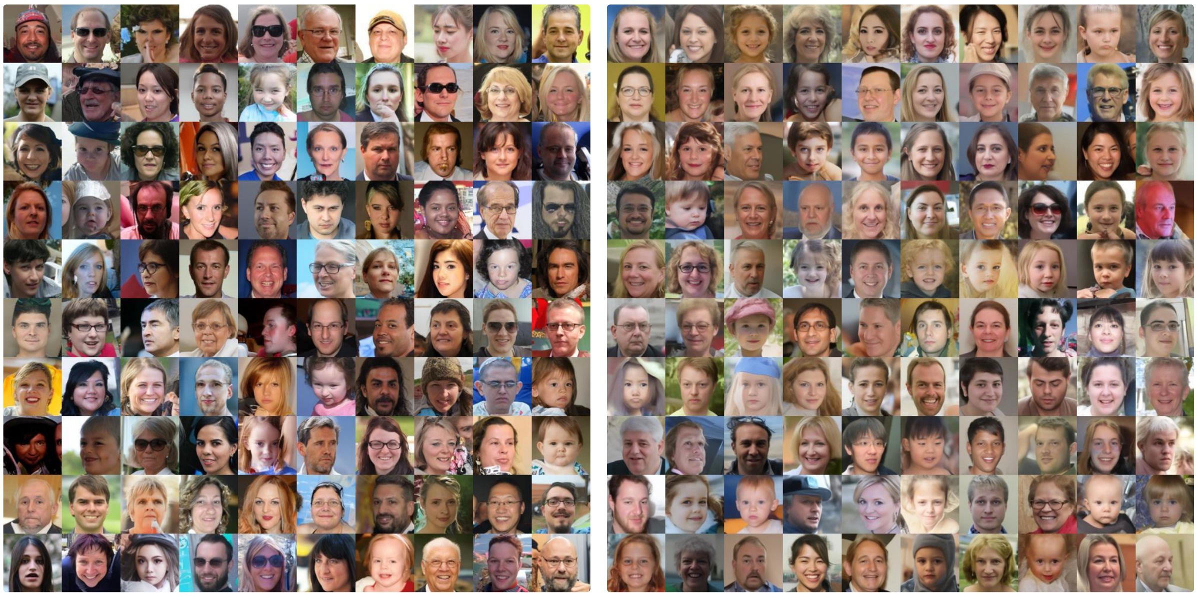


Figure 7.2: Three distinct generation paradigms are compared w.r.t. their representative models. Only (a) DDIM and (c) ProgDiffusion enable unconditional generation from noise samples $\mathbf{x} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. Conversely, (b) DiffAE only can do conditional sampling, depending on the input sample $\mathbf{x}_0$. The generation by ProgDiffusion is based solely on a simple distribution and closely resembles DDIM, leveraging the strengths of probabilistic models to approximate the data likelihood distribution.

7.4 Experiments

We validate ProgDiffusion on several real-world high-resolution datasets, including FFHQ, CelebA-HQ, LSUN-Horse, and LSUN-Bedroom. To comprehensively evaluate



(a) Conditional Generation by DiffAE

(b) Unconditional Generation by ProgDiffusion

Figure 7.3: Comparing the images generated from DiffAE and ProgDiffusion after training on FFHQ: (a) The conditionally generated samples by DiffAE tend to retain redundant background information. In contrast, (b) the unconditionally generated samples by ProgDiffusion focus more closely on the dataset likelihood, capturing more detailed information in each sample.

the generation and representation abilities of ProgDiffusion and their significance, we design several vision tasks to answer the following research questions:

- **RQ1:** What advantages can ProgDiffusion offer in generation and how does ProgDiffusion perform compared with state-of-the-art generative models?

- **RQ2:** How about the effectiveness of the learned time-specific encoding?

- **RQ3:** Is the intermediate latents $\mathbf{x}_t$ smooth and continuous?

- **RQ4:** Is the semantic representation, encoded in ProgDiffusion, represented by $\mathbf{z}_t \sim q_\phi(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1}, t)$ and learned by the encoder, both smooth and continuous?

- **RQ5:** How does ProgDiffusion perform on downstream tasks?

.

7.4.1 RQ1: Improving Unconditional Generation

The key strength of ProgDiffusion lies in its ability to generate unconditionally contrasting to other diffusion-autoencoder-pipeline models. ProgDiffusion has the capability of generating output samples that are not conditioned on any specific input data but rather solely depend on noises sampled from a specified prior (typically a normal distribution). Figure 7.2 elucidates the difference and sustained advantages inherent in the ProgDiffusion model.

We employ the clean-FID [117] to quantify the unconditional generation quality of ProgDiffusion over 10,000 samples. ProgDiffusion conducts the unconditional generation on four real-world datasets based on Algorithm 7. The unconditional generation for DDIM follows [131] merely on sampled noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. It should be noted that DiffAE cannot perform unconditional generation (those generated samples are meaningless with a high FID score at around 300). Table 7.2 shows that ProgDiffusion achieves a lower FID score than DDIM in various time self-encoders conditioned on $\mathbf{x}_t$ only and $\mathbf{x}_t, t$. Figure 7.3 shows that the unconditionally sampled images by ProgDiffusion consistently ensure shape generation quality while maintaining the richness and variety of real-world data sets.

7.4.2 RQ2: Visualizing Time Specific Semantics

To evaluate the effectiveness of our time self-encoder, we design a time-semantic encoding task to intuitively demonstrate the encoded semantic information spanning over time. First, we acquire the fixed subcode $\mathbf{x}_T^{1:4}$, i.e., T-step noises conditioned on the raw input $\mathbf{x}$ in the diffusion process. Second, we diversify the time-specific semantic representation $\mathbf{z}_t$ by our self-encoders by changing the input pairs $\langle \mathbf{x}_t, t\rangle^{1:4}$. For each input pair, we obtain the reconstruction $\hat{\mathbf{x}}^{1:4}$ by the corresponding fixed subcodes, and time-specific encodes $\langle \mathbf{x}_T, \mathbf{z}_t\rangle^{1:4}$ in the reverse process. As shown in Figure 7.4, the reconstructed images change over t times. This demonstrates that our model can capture the time-dependent low-dimensional semantics.

7.4.3 RQ3: Time-dependent Semantics Guided Interpolation

We employ the latent interpolation tasks to validate the smoothness of subcodes learned in ProgDiffusion. Latent interpolation [53, 61] can validate the smoothness, continuity, and semantic coherence property in the representation space learned by generative models. Typically, two samples are embedded into their corresponding latent spaces, and

Table 7.2: FID scores (↓) for unconditional generation. ProgDiffusion is competitive with DDIM baselines. ProgDiffusion has a self-encoder $q(\mathbf{z}_t|\mathbf{x}_t)$, where '+ timeEnc' refers to the ProgDiffusion with sample-time based self-encoder, i.e., $q(\mathbf{z}_t|\mathbf{x}_t, \mathbf{u}_{t+1}, t)$. The bold font indicates the best.

Dataset	**Model**	**FID ↓**			
		T=10	**T=20**	**T=50**	**T=100**
FFHQ-128	DDIM	29.56	21.45	15.08	12.03
	ProgDiffusion	22.77	18.14	15.52	12.88
	+ timeEnc	**21.28**	**17.30**	**13.06**	**11.01**
Horse-128	DDIM	22.12	12.92	**7.92**	**5.97**
	ProgDiffusion	14.70	12.88	9.71	7.12
	+ timeEnc	**13.92**	**10.27**	8.11	6.85
Bedroom-128	DDIM	13.70	9.23	**7.14**	5.94
	ProgDiffusion	12.63	10.59	8.55	7.10
	+ timeEnc	**11.07**	**8.97**	7.62	**5.91**
CelebA-64	DDIM	16.38	12.70	8.52	**5.83**
	ProgDiffusion	15.32	12.17	8.87	7.96
	+ timeEnc	**13.69**	**11.03**	**7.26**	6.29

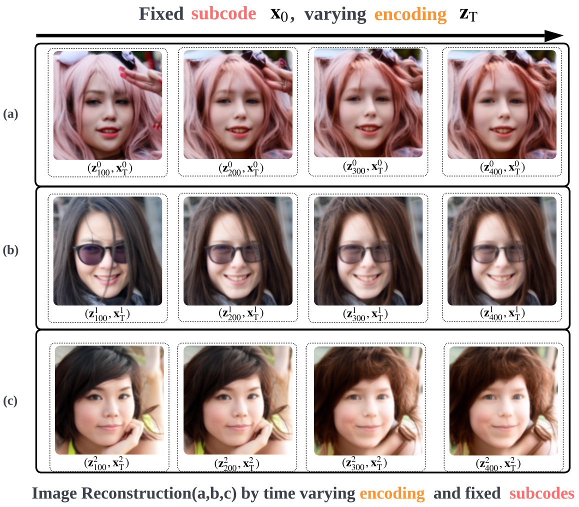

Fixed subcode $\mathbf{x}_0$, varying encoding $\mathbf{z}_T$
(a)
$(\mathbf{z}^0_{100}, \mathbf{x}^0_T)$
$(\mathbf{z}^0_{200}, \mathbf{x}^0_T)$
$(\mathbf{z}^0_{300}, \mathbf{x}^0_T)$
$(\mathbf{z}^0_{400}, \mathbf{x}^0_T)$
(b)
$(\mathbf{z}^1_{100}, \mathbf{x}^1_T)$
$(\mathbf{z}^1_{200}, \mathbf{x}^1_T)$
$(\mathbf{z}^1_{300}, \mathbf{x}^1_T)$
$(\mathbf{z}^1_{400}, \mathbf{x}^1_T)$
(c)
$(\mathbf{z}^2_{100}, \mathbf{x}^2_T)$
$(\mathbf{z}^2_{200}, \mathbf{x}^2_T)$
$(\mathbf{z}^2_{300}, \mathbf{x}^2_T)$
$(\mathbf{z}^2_{400}, \mathbf{x}^2_T)$
Image Reconstruction(a,b,c) by time varying encoding and fixed subcodes

Figure 7.4: Time-specific Semantic Encoding Task: identifying the time-specific changes in the progressive encoding of ProgDiffusion. By varying time encoding for 100, 200, 300, 400 time steps, more attributes will be influenced in the reconstruction stage: the `Young`, `Bangs`, and `Brown_Hair` attributes in (a), the `Blond_Hair`, `Pale_skin` attributes in (b), and the `Wavy_Hair`, `High_Cheekbones` and `Blond_Hair` attributes in (c).

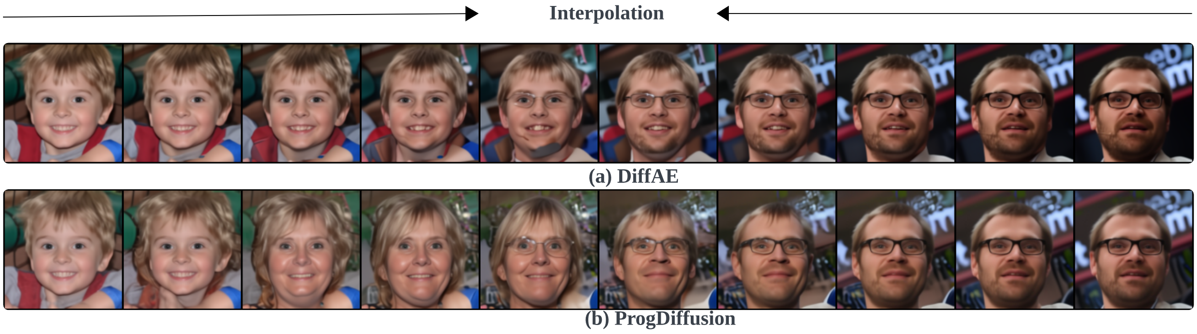


Figure 7.5: The Comparison of DiffAE and ProgDiffusion in latent interpolation tasks. From the left to right side, we can see that both diffusion based models support continuous representation learning with features varying gradually from the first image to the second. However, with time-dependent semantics, ProgDiffusion smoothly guides the latent variation, as depicted in (b). Meanwhile, the fifth image in (a) indicates that DiffAE can lead to the feature collapse during latent variation.

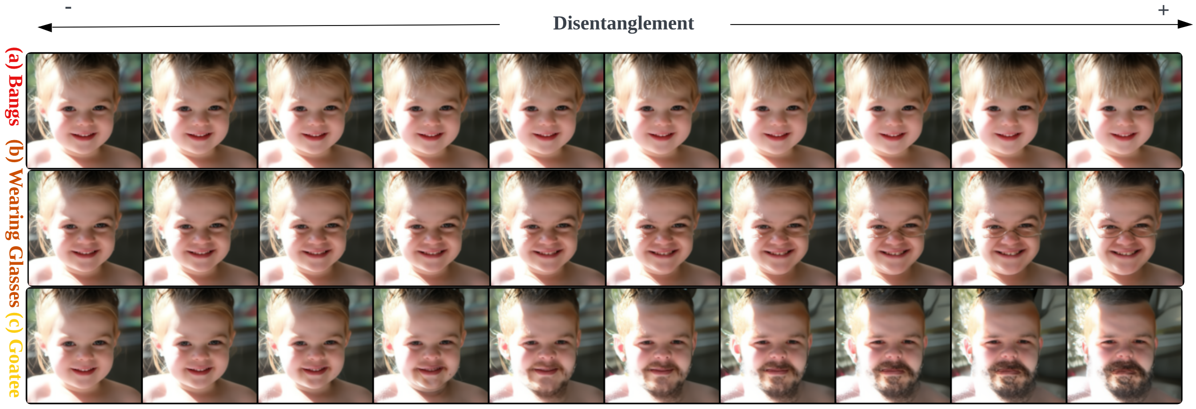


Figure 7.6: Semantic disentanglement observed in ProgDiffusion through latent traversals on FFHQ. The interpretable traversal directions are illustrated by traversing the semantic representation within the range of $[-3,3]$. Independent variations in attributes such as `Bangs` in (a), `Wearing Glasses` in (b), and `Goatee` in (c) are obvious, while other attributes remain unchanged.

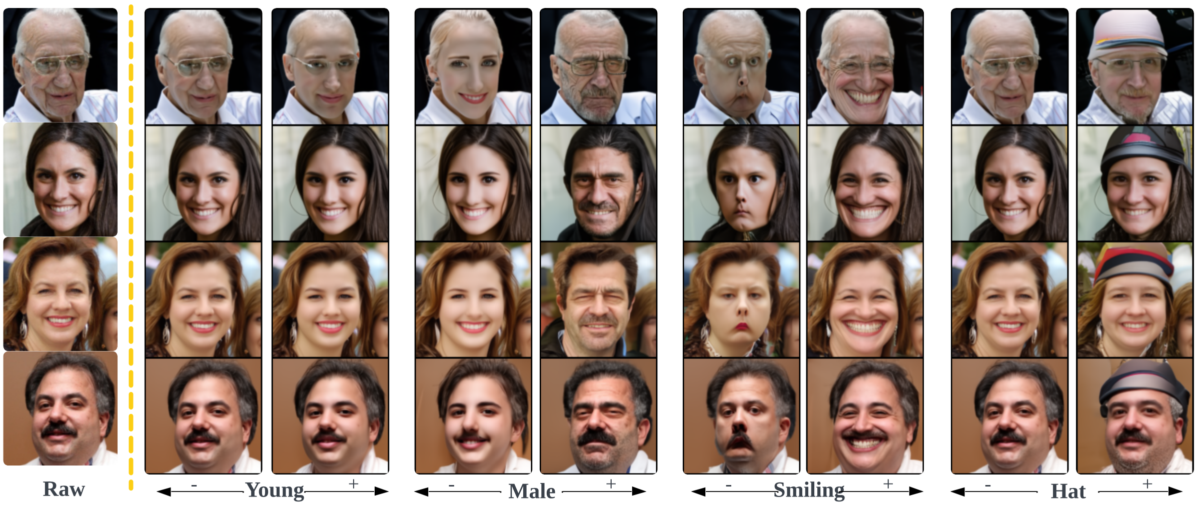


Figure 7.7: Attribute manipulation on FFHQ utilizing a linear classifier.

interpolating these latent variables yields interpolated representations. The semantic richness of the learned space is revealed through the reconstruction performed by generative models.

7.4.4 RQ4: Time-dependent Semantic Encoding for Disentanglement

The validity of the semantic encoding **z** improved by the self-encoders can be examined through disentanglement tasks. Disentanglement refers to the process whereby generative models find the information within a dataset into meaningful segments, each residing within a set of encoding semantics **z**. In ProgDiffusion, we introduce the TC term to diminish the dependencies between the dimensions of **z**, enabling each dimension to encapsulate distinct feature information. The effect of this disentanglement can be visually observed in Figure 7.6. Additionally, we employ Total AUROC Difference (TAD) and learned attributes (ATTRS) [179] to quantify the disentangled information. We also utilize the FID score to evaluate the generative capability of our method while achieving disentangled representation over 10,000 samples. Table 7.3 presents a performance comparison, illustrating that ProgDiffusion consistently outperforms all baseline models across various FID and TAD metrics.

Table 7.3: Performance comparison of different methods on TAD, ATTRS and FID metrics on CelebA. The bold font indicates the best.

Models	**TAD** (↑)	**ATTRS** (↑)	**FID** (↓)
AE	0.042 ± 0.004	1.0 ± 0.0	90.4 ± 1.8
DiffAE [118]	0.155 ± 0.010	2.0 ± 0.0	22.7 ± 2.1
VAE [76]	0.000 ± 0.000	0.0 ± 0.0	94.3 ± 2.8
β-VAE [20]	0.088 ± 0.051	1.6 ± 0.8	99.8 ± 2.4
InfoVAE [188]	0.000 ± 0.000	0.0 ± 0.0	77.8 ± 1.6
InfoDiffusion [150]	0.299 ± 0.006	3.0 ± 0.0	22.3 ± 1.2
DisDiff [175]	0.305 ± 0.010	–	$\mathbf{18.3 \pm 2.1}$
ProgDiffusion	$\mathbf{0.470 \pm 0.008}$	$\mathbf{4.0 \pm 0.0}$	22.8 ± 1.3

7.4.5 RQ5: Time-dependent Semantics for Attribute Manipulation

The semantics encoded by ProgDiffusion are meaningful and low-dimensional, making them suitable for downstream tasks. To validate this, we perform attribute manipulation tasks. Specifically, we train a linear classifier on the latent codes derived from images with negative and positive manifestations of a target attribute, determining the linear trajectory for attribute alterations. Consequently, the classifier can provide gradients for ProgDiffusion to generate targeted modifications, as discussed in [39, 63]. Following [118], we conduct experiments for ProgDiffusion trained on the FFHQ dataset and training the classifier on the 40 attributes in CelebA-HQ. Figure 7.7 illustrates the manipulation outcomes over four distinct attributes of the FFHQ dataset.

7.5 Summary of This Chapter

In this chapter, we propose ProgDiffusion, a novel approach for learning low-dimensional, progressive latents tailored for expanding DMs. Different from the previous DAE-like models, ProgDiffusion integrates a self-encoder conditioned on time-specific input and time steps, enabling it to perform unconditional generation and time-specific semantic encoding tasks. To embed the lantent completely, the self-encoder was implemented by a hierarchical time dependent encoder network to learn multi-scale semantics inherent in U-Net. The quantitative and qualitative experimental results on the 4 high-resolution datasets have demonstrated that ProgDiffusion advanced in unconditional generation quality and various downstream representation learning tasks.

CHAPTER 8

Conclusion and Future Work

In this chapter, we provide a comprehensive conclusion for the entire thesis and outline several promising directions for future research.

8.1 Conclusion

Deep Variational Generative Models (DVGM) have demonstrated significant potential in numerous applications, thanks to their strong representational power and ability to handle uncertainties in complex data distributions. These models have become essential tools in fields such as anomaly detection, data density estimation, image modeling, and representation learning. By combining the strengths of deep learning with variational inference, DVGM provides flexible and expressive models suited for a wide range of data-driven tasks. This versatility has enabled advancements in machine learning, data analysis, and computer vision, where DVGM effectively models high-dimensional data and contributes to areas like semi-supervised learning and complex data generation. Despite their advantages, DVGMs face challenges related to inference and generation due to the complexities of variational inference, such as balancing inference robustness with generation quality and managing coupled representations. Addressing these limitations is key to fully realizing the potential of DVGMs.

- **How can evolutionary mechanisms balance inference and generation in DVGM?** In Chapter 3, we introduce eVAE, the first framework to incorporate evolutionary learning with variational autoencoders. eVAE dynamically optimizes the balance between reconstruction accuracy and inference robustness, addressing a longstanding challenge in VAEs. By using variational genetic operators, eVAE mitigates issues like premature convergence and random search. Our framework combines stochastic gradient descent with genetic algorithms through an inner-outer-joint training mechanism. Guided by information bottleneck theory, eVAE introduces an iteration-specific lower bound to balance compression and decompression over time.

- **How can DVGM calibrate inference to separate disentangled and coupled representations?** In Chapter 4, we propose C^{2}VAE, trained with contrastive disentangled learning to separate and remove coupled features and their representations. This enables C^{2}VAE to learn factorizable representations for disentanglement, effectively eliminating strongly coupled features through copula-based dependency learning.

- **How can weak augmentation improve inference robustness in DVGM for anomaly detection?** In Chapter 5, we present WAVAE, a weakly augmented VAE for time series anomaly detection. The model achieves a more robust latent space representation through joint training on augmented data. We also introduce two self-supervised strategies, adversarial and contrastive learning, to enhance data fitting performance.

- **How can DVGM enhance inference in complex parameter spaces for improved generation?** In Chapter 6, we propose a novel unified parameter space representation learning framework to handle continuous, discrete, and discretized data. Unlike traditional encoders that map observations into static latent semantics, ParamReL uses a self-encoder to derive progressively structured latent semantics from intermediate parameters at each generation step. This framework facilitates effective representation learning across data types, as validated by experiments on tasks like latent interpolation, disentanglement, and conditional generation. The results demonstrate its ability to extract meaningful high-level semantics, yielding unified representations and a clearer semantic understanding of the data.

- **How can progressive inference facilitate low-dimensional generation in diffusion models?** In Chapter 7, we introduce ProgDiffusion, a Progressive Self-

Encoded Diffusion model that achieves efficient unconditional generation and progressively structured semantic representations. This model leverages a self-encoder mechanism using U-Net's upsampling features, intermediate latents, and denoising timesteps to generate time-specific semantic representations. Diverging from conventional methods that rely on observation conditioning, our encoder operates independently of input data, enabling unconditional generation.

8.2 Future Work

The methods presented in this thesis address critical questions in DVGM, yet several areas remain open for further exploration, presenting exciting opportunities for future research.

- **Exploring alternative evolutionary mechanisms for adaptive inference and generation balance:** While eVAE integrates evolutionary learning with variational autoencoders, future research could explore other evolutionary mechanisms to enhance DVGM adaptability. Investigating alternative genetic operators or hybridizing evolutionary algorithms with advanced deep learning techniques may further refine the balance between reconstruction accuracy and inference robustness in complex data settings.

- **Developing advanced disentanglement techniques for coupled representations:** Although C^{2}VAE showcases effective contrastive disentangled learning, disentangling highly coupled features remains challenging in high-dimensional spaces. Future work could focus on more sophisticated dependency modeling methods, such as multi-level copula-based learning or hierarchical contrastive techniques, to improve disentanglement and enable richer, factorized representations across diverse data modalities.

- **Incorporating novel augmentation strategies for anomaly detection:** WAVAE has shown that weak augmentation can improve inference robustness in anomaly detection. Extending this approach with advanced augmentation methods, such as adversarial augmentations or domain-specific transformations, could enhance the robustness and generalizability of anomaly detection models in various applications.

- **Enhancing representation learning in complex parameter spaces for DVGM:** The ParamReL framework is a promising step toward unified parameter space representation learning, but optimizing across heterogeneous data types remains challenging. Future research could explore adaptive parameter space learning methods that dynamically adjust the representation strategy based on data characteristics, improving generation quality and interpretability in complex and dynamic parameter spaces.

- **Refining progressive inference for low-dimensional generation in diffusion models:** While ProgDiffusion demonstrates the potential of progressive inference for low-dimensional generation, further refinement is needed for complex generative tasks. Exploring multi-stage progressive encoding and decoding schemes or leveraging hierarchical latent spaces could yield structured and semantically rich representations, enabling DVGM to excel in applications requiring fine-grained and conditional generation.

- **Expanding DVGM applications to cross-modal and multi-modal generation:** Although DVGM has proven effective in tasks like anomaly detection and image modeling, expanding its applications to cross-modal and multi-modal settings (e.g., text-to-image or audio-visual generation) represents a promising research direction. Cross-modal variational inference and generative modeling could enable DVGM to integrate multiple data modalities, broadening its utility in real-world applications requiring multimodal integration.

Future Perspectives in AI Research

Looking ahead to the next 3–5 years, several emerging trends in artificial intelligence research are poised to reshape the landscape of generative modeling and variational inference. One prominent direction is the growing integration of *multi-modal learning*, where models are designed to jointly process and generate data across different modalities, such as text, image, audio, and video. This necessitates more expressive latent variable models capable of aligning heterogeneous representations while preserving modality-specific structures.

Another key trend is the advancement of *foundation models* and *large-scale pretraining*, which encourage the development of DVGM-based architectures that can leverage transfer learning, continual learning, and in-context learning. These models are expected to exhibit better generalization and adaptability to diverse downstream tasks, especially under data-scarce or domain-shift scenarios.

Additionally, *interpretability* and *controllability* are gaining increased attention, particularly in high-stakes domains such as healthcare, science, and decision-making systems. Future research may focus on developing structured latent spaces that facilitate reasoning, explanation, and human-aligned interaction.

Finally, the emergence of *energy-efficient* and *resource-aware* learning methods also motivates the design of compact yet powerful generative models, where hierarchical and variational techniques can help reduce redundancy and improve computational efficiency. In this context, DVGM will likely play a central role in bridging probabilistic modeling with real-world AI deployment.

BIBLIOGRAPHY

[1] A. ABID AND J. ZOU, *Contrastive variational autoencoder enhances salient features*, arXiv preprint arXiv:1902.04601, (2019).

[2] M. ABROSHAN, K. H. YIP, C. TEKIN, AND M. VAN DER SCHAAR, *Conservative policy construction using variational autoencoders for logged data with missing values*, IEEE Transactions on Neural Networks and Learning Systems, (2022).

[3] Q. AI, P. WANG, L. HE, L. WEN, L. PAN, AND Z. XU, *Generative oversampling for imbalanced data via majority-guided vae*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2023, pp. 3315–3330.

[4] H. AKRAMI, S. AYDORE, R. M. LEAHY, AND A. A. JOSHI, *Robust variational autoencoder for tabular data with beta divergence*, arXiv preprint arXiv:2006.08204, (2020).

[5] M. S. ALBERGO, N. M. BOFFI, AND E. VANDEN-EIJNDEN, *Stochastic interpolants: A unifying framework for flows and diffusions*, arXiv preprint arXiv:2303.08797, (2023).

[6] J. M. L. ALCARAZ AND N. STRODTHOFF, *Diffusion-based time series imputation and forecasting with structured state space models*, arXiv preprint arXiv:2208.09399, (2022).

[7] A. ALEMI, B. POOLE, I. FISCHER, J. DILLON, R. A. SAUROUS, AND K. MURPHY, *Fixing a Broken ELBO*, ICML, (2018).

[8] A. A. ALEMI, I. FISCHER, J. V. DILLON, AND K. MURPHY, *Deep variational information bottleneck*, arXiv preprint arXiv:1612.00410, (2016).

[9] J. ANEJA, A. SCHWING, J. KAUTZ, AND A. VAHDAT, *Ncp-vae: Variational autoencoders with noise contrastive priors*, (2020).

[10] M. A. Arcones and E. Gine, *On the bootstrap of u and v statistics*, The Annals of Statistics, (1992), pp. 655–674.

[11] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg, *Structured denoising diffusion models in discrete state-spaces*, Advances in Neural Information Processing Systems, (2021).

[12] J. Bae, M. R. Zhang, M. Ruan, E. Wang, S. Hasegawa, J. Ba, and R. Grosse, *Multi-rate vae: Train once, get the full rate-distortion curve*, arXiv preprint arXiv:2212.03905, (2022).

[13] K. Bai, P. Cheng, W. Hao, R. Henao, and L. Carin, *Estimating total correlation with mutual information estimators*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2023, pp. 2147–2164.

[14] D. Baranchuk, I. Rubachev, A. Voynov, V. Khrulkov, and A. Babenko, *Label-Efficient Semantic Segmentation with Diffusion Models*, arXiv preprint arXiv:2112.03126, (2021).

[15] Y. Bengio, A. Courville, and P. Vincent, *Representation Learning: A Review and New Perspectives*, TPAMI, 35 (2013), pp. 1798–1828.

[16] R. v. d. Berg, L. Hasenclever, J. M. Tomczak, and M. Welling, *Sylvester normalizing flows for variational inference*, arXiv preprint arXiv:1803.05649, (2018).

[17] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, *Generating sentences from a continuous space*, arXiv preprint arXiv:1511.06349, (2015).

[18] Y. Burda, R. Grosse, and R. Salakhutdinov, *Importance weighted autoencoders*, arXiv preprint arXiv:1509.00519, (2015).

[19] C. Burgess and H. Kim, *3d shapes dataset*.
https://github.com/deepmind/3dshapes-dataset/, 2018.

[20] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, *nderstanding disentangling in β-VAE*, arXiv preprint arXiv:1804.03599, (2018).

[21] L. CAO, P. S. YU, AND Z. ZHAO, *Shallow and deep non-iid learning on complex data*, in KDD '22, 2022, pp. 4774–4775.

[22] M.-A. CARBONNEAU, J. ZAIDI, J. BOILARD, AND G. GAGNON, *Measuring disentanglement: A review of metrics*, IEEE Transactions on Neural Networks and Learning Systems, (2022).

[23] C. I. CHALLU, P. JIANG, Y. N. WU, AND L. CALLOT, *Deep generative model with hierarchical latent factors for time series anomaly detection*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 1643–1654.

[24] K. CHAUHAN, P. SHENOY, M. GUPTA, D. SRIDHARAN, ET AL., *Robust outlier detection by de-biasing vae likelihoods*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9881–9890.

[25] J. CHEN, S. SATHE, C. AGGARWAL, AND D. TURAGA, *Outlier detection with autoencoder ensembles*, in Proceedings of the 2017 SIAM international conference on data mining, SIAM, 2017, pp. 90–98.

[26] R. T. CHEN, X. LI, R. GROSSE, AND D. DUVENAUD, *Isolating sources of disentanglement in VAEs*, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, vol. 2615, p. 2625.

[27] R. T. CHEN, X. LI, R. B. GROSSE, AND D. K. DUVENAUD, *Isolating Sources of Disentanglement in Variational Autoencoders*, NeurIPS, (2018).

[28] T. CHEN, R. ZHANG, AND G. HINTON, *Analog bits: Generating discrete data using diffusion models with self-conditioning*, arXiv preprint arXiv:2208.04202, (2022).

[29] W. CHEN, L. TIAN, B. CHEN, L. DAI, Z. DUAN, AND M. ZHOU, *Deep variational graph convolutional recurrent network for multivariate time series anomaly detection*, in International Conference on Machine Learning, PMLR, 2022, pp. 3621–3633.

[30] X. CHEN, Z. LIU, S. XIE, AND K. HE, *Deconstructing denoising diffusion models for self-supervised learning*, arXiv preprint arXiv:2401.14404, (2024).

[31] X. Chen, C. Wang, X. Lan, N. Zheng, and W. Zeng, *Neighborhood geometric structure-preserving variational autoencoder for smooth and bounded data sources*, IEEE Transactions on Neural Networks and Learning Systems, 33 (2021), pp. 3598–3611.

[32] Y. Chen, Y. Hao, T. Rakthanmanon, J. Zakaria, B. Hu, and E. n. Keogh, *A general framework for never-ending learning from time series streams*, Data mining and knowledge discovery, 29 (2015), pp. 1622–1664.

[33] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, *Club: A contrastive log-ratio upper bound of mutual information*, in International conference on machine learning, PMLR, 2020, pp. 1779–1788.

[34] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, *A recurrent latent variable model for sequential data*, Advances in neural information processing systems, 28 (2015).

[35] W. Dai, K. Ng, K. Severson, W. Huang, F. Anderson, and C. Stultz, *Generative oversampling with a contrastive variational autoencoder*, in 2019 IEEE International Conference on Data Mining (ICDM), IEEE, 2019, pp. 101–109.

[36] K. Deb and H.-G. Beyer, *Self-adaptive genetic algorithms with simulated binary crossover*, Evolutionary Computation, 9 (2001), pp. 197–221.

[37] K. Deja, P. Wawrzyński, W. Masarczyk, D. Marczak, and T. Trzciński, *Multiband vae: Latent space alignment for knowledge consolidation in continual learning*, arXiv preprint arXiv:2106.12196, (2021).

[38] L. Deng, *The MINST Database of Handwritten Digit Images for Machine Learning Research*, IEEE Signal Processing Magazine, 29 (2012), pp. 141–142.

[39] P. Dhariwal and A. Nichol, *Diffusion models beat gans on image synthesis*, Advances in neural information processing systems, (2021).

[40] J. Domke and D. R. Sheldon, *Divide and couple: Using monte carlo variational objectives for posterior approximation*, Advances in neural information processing systems, 32 (2019).

[41] S. DONG, H. HU, D. LIAN, W. LUO, Y. QIAN, AND S. GAO, *Weakly supervised video representation learning with unaligned text for sequential videos*, IEEE Conference on Computer Vision and Pattern Recognition, (2023).

[42] Y. DUAN, L. WANG, Q. ZHANG, AND J. LI, *Factorvae: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 4468–4476.

[43] C. EASTWOOD AND C. K. WILLIAMS,*A Framework for the Quantitative Evaluation of Disentangled Representations*, ICLR, (2018).

[44] B. ESMAEILI, R. WALTERS, H. ZIMMERMANN, AND J.-W. VAN DE MEENT, *Topological Obstructions and How to Avoid Them*, NeurIPS, (2023).

[45] B. ESMAEILI, H. WU, S. JAIN, A. BOZKURT, N. SIDDHARTH, B. PAIGE, D. H. BROOKS, J. DY, AND J.-W. MEENT, *Structured disentangled representations*, in The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 2525–2534.

[46] B. EVERETT,*An introduction to latent variable models*, Springer Science & Business Media, 2013.

[47] V. FORTUIN, M. HÜSER, F. LOCATELLO, H. STRATHMANN, AND G. RÄTSCH, *Som-vae: Interpretable discrete representation learning on time series*, arXiv preprint arXiv:1806.02199, (2018).

[48] J.-Y. FRANCESCHI, A. DIEULEVEUT, AND M. JAGGI, *Unsupervised Scalable Representation Learning for Multivariate Time Series*, NeurIPS, (2019).

[49] H. FU, C. LI, X. LIU, J. GAO, A. CELIKYILMAZ, AND L. CARIN, *Cyclical annealing schedule: A simple approach to mitigating KL vanishing*, arXiv preprint arXiv:1903.10145, (2019).

[50] F. FUTAMI, I. SATO, AND M. SUGIYAMA, *Variational inference based on robust divergences*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2018, pp. 813–822.

[51] S. GAO, R. BREKELMANS, G. VER STEEG, AND A. GALSTYAN,*Auto-encoding total correlation explanation*, in The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1157–1166.

[52] P. Ghosh, M. S. Sajjadi, A. Vergari, M. Black, and B. Schölkopf, *From variational to deterministic autoencoders*, arXiv preprint arXiv:1903.12436, (2019).

[53] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, Advances in Neural Information Processing Systems, (2014).

[54] A. Graves, R. K. Srivastava, T. Atkinson, and F. Gomez, *Bayesian Flow Networks*, arXiv preprint arXiv:2308.07037, (2023).

[55] R. M. Gray, *Source coding theory*, vol. 83, Springer Science & Business Media, 1989.

[56] J. Guo, X. Xu, Y. Pu, Z. Ni, C. Wang, M. Vasu, S. Song, G. Huang, and H. Shi, *Smooth diffusion: Crafting smooth latent spaces in diffusion models*, IEEE Conference on Computer Vision and Pattern Recognition, (2024).

[57] R. Hadsell, S. Chopra, and Y. LeCun, *Dimensionality reduction by learning an invariant mapping*, in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, IEEE, 2006, pp. 1735–1742.

[58] J. D. Havtorn, J. Frellsen, S. Hauberg, and L. Maaløe, *Hierarchical VAEs know what they don‚Äôt know*, in International Conference on Machine Learning, PMLR, 2021, pp. 4117–4128.

[59] E. Hedlin, G. Sharma, S. Mahajan, H. Isack, A. Kar, A. Tagliasacchi, and K. M. Yi, *Unsupervised Semantic Correspondence Using Stable Diffusion*, NeurIPS, (2024).

[60] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, *Prompt-to-prompt image editing with cross attention control*, arXiv preprint arXiv:2208.01626, (2022).

[61] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, *beta-vae: Learning basic visual concepts with a constrained variational framework.*, International Conference on Learning Representations, (2017).

[62] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, Advances in Neural Information Processing Systems, (2020).

[63] J. Ho and T. Salimans, *Classifier-free diffusion guidance*, arXiv preprint arXiv:2207.12598, (2022).

[64] M. Hou, C. Xu, Z. Li, Y. Liu, W. Liu, E. Chen, and J. Bian, *Multi-granularity residual learning with confidence estimation for time series prediction*, in Proceedings of the ACM Web Conference 2022, 2022, pp. 112–121.

[65] D. A. Hudson, D. Zoran, M. Malinowski, A. K. Lampinen, A. Jaegle, J. L. McClelland, L. Matthey, F. Hill, and A. Lerchner, *Soda: Bottleneck diffusion models for representation learning*, IEEE Conference on Computer Vision and Pattern Recognition, (2024).

[66] G. Hwang, J. Choi, H. Cho, and M. Kang, *Maganet: Achieving combinatorial generalization by modeling a group action*, International Conference on Machine Learning, (2023).

[67] M. Ilse, J. M. Tomczak, C. Louizos, and M. Welling, *Diva: Domain invariant variational autoencoders*, in Medical Imaging with Deep Learning, PMLR, 2020, pp. 322–348.

[68] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, *Variational deep embedding: An unsupervised and generative approach to clustering*, arXiv preprint arXiv:1611.05148, (2016).

[69] P. Jiao, X. Guo, X. Jing, D. He, H. Wu, S. Pan, M. Gong, and W. Wang, *Temporal network embedding for link prediction via vae joint attention mechanism*, IEEE Transactions on Neural Networks and Learning Systems, 33 (2021), pp. 7400–7413.

[70] X.-B. Jin, W.-T. Gong, J.-L. Kong, Y.-T. Bai, and T.-L. Su, *Pfvae: a planar flow-based variational auto-encoder prediction model for time series data*, Mathematics, 10 (2022), p. 610.

[71] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen, *Variational autoencoders and nonlinear ICA: A unifying framework*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2207–2217.

[72] T. Kieu, B. Yang, C. Guo, R.-G. Cirstea, Y. Zhao, Y. Song, and C. S. Jensen, *Anomaly detection in time series with robust variational quasi-recurrent au-*

toencoders, in 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, 2022, pp. 1342–1354.

[73] T. KIEU, B. YANG, AND C. S. JENSEN, *Outlier detection for multidimensional time series using deep neural networks*, in 2018 19th IEEE international conference on mobile data management (MDM), IEEE, 2018, pp. 125–134.

[74] H. KIM AND A. MNIH, *Disentangling by Factorising*, ICML, (2018).

[75] D. P. KINGMA, S. MOHAMED, D. JIMENEZ REZENDE, AND M. WELLING, *Semi-supervised learning with deep generative models*, Advances in neural information processing systems, 27 (2014).

[76] D. P. KINGMA AND M. WELLING, *Auto-encoding variational bayes*, arXiv preprint arXiv:1312.6114, (2013).

[77] A. KLUSHYN, N. CHEN, R. KURLE, B. CSEKE, AND P. VAN DER SMAGT, *Learning hierarchical priors in VAEs*, Advances in neural information processing systems, 32 (2019).

[78] R. KRISHNAN, D. LIANG, AND M. HOFFMAN, *On the challenges of learning with inference networks on sparse, high-dimensional data*, AISTATS, (2018).

[79] A. KRIZHEVSKY AND G. HINTON, *Learning Multiple Layers of Features from Tiny Images*, technical report, University of Toronto, 2009.

[80] A. KUMAR, P. SATTIGERI, AND A. BALAKRISHNAN, *Variational inference of disentangled latent concepts from unlabeled observations*, arXiv preprint arXiv:1711.00848, (2017).

[81] O. KVIMAN, H. MELIN, H. KOPTAGEL, V. ELVIRA, AND J. LAGERGREN, *Multiple importance sampling ELBO and deep ensembles of variational approximations*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 10687–10702.

[82] J. KWOK AND R. P. ADAMS, *Priors for Diversity in Generative Latent Variable Models*, NeurIPS, (2012).

[83] C.-Y. LAI, F.-K. SUN, Z. GAO, J. H. LANG, AND D. S. BONING, *Nominality score conditioned time series anomaly detection by point/sequential reconstruction*, arXiv preprint arXiv:2310.15416, (2023).

[84] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, *A tutorial on energy-based learning*, Predicting structured data, 1 (2006).

[85] Y. LeCun, F. J. Huang, and L. Bottou, *Learning methods for generic object recognition with invariance to pose and lighting*, in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., vol. 2, IEEE, 2004, pp. II–104.

[86] H. Lee, E. Seong, and D.-K. Chae, *Self-supervised learning with attention-based latent signal augmentation for sleep staging with limited labeled data*, in Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, LD Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, vol. 7, 2022, pp. 3868–3876.

[87] T. Li, H. Chang, S. K. Mishra, H. Zhang, D. Katabi, and D. Krishnan, *MAGE: MAsked Generative Encoder to Unify Representation Learning and image synthesis*, CVPR, (2023).

[88] X. Li, J. Lu, K. Han, and V. A. Prisacariu, *Sd4match: Learning to prompt stable diffusion model for semantic matching*, IEEE Conference on Computer Vision and Pattern Recognition, (2024).

[89] Y. Li, W. Chen, B. Chen, D. Wang, L. Tian, and M. Zhou, *Prototype-oriented unsupervised anomaly detection for multivariate time series*, in ICML 2023, vol. 202 of Proceedings of Machine Learning Research, 2023, pp. 19407–19424.

[90] Y. Li, X. Lu, Y. Wang, and D. Dou, *Generative time series forecasting with diffusion, denoise, and disentanglement*, Advances in Neural Information Processing Systems, 35 (2022), pp. 23009–23022.

[91] Y. Li, C. Wang, X. Xia, T. Liu, B. An, et al., *Out-of-distribution detection with an adaptive likelihood ratio on informative hierarchical vae*, Advances in Neural Information Processing Systems, 35 (2022), pp. 7383–7396.

[92] Z. Li, J. V. Murkute, P. K. Gyawali, and L. Wang, *Progressive learning and disentanglement of hierarchical representations*, arXiv preprint arXiv:2002.10549, (2020).

[93] W. Liao, Y. Guo, X. Chen, and P. Li, *A unified unsupervised gaussian mixture variational autoencoder for high dimensional outlier detection*, in 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 1208–1217.

[94] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts, *Anomaly detection for time series using vae-lstm hybrid model*, in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Ieee, 2020, pp. 4322–4326.

[95] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, *Flow matching for generative modeling*, arXiv preprint arXiv:2210.02747, (2022).

[96] X. Liu, C. Gong, and Q. Liu, *Flow straight and fast: Learning to generate and transfer data with rectified flow*, arXiv preprint arXiv:2209.03003, (2022).

[97] Z. Liu, P. Luo, X. Wang, and X. Tang, *Deep Learning Face Attributes in the Wild*, ICCV, (2015).

[98] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, and O. Bachem, *On the fairness of disentangled representations*, Advances in neural information processing systems, 32 (2019).

[99] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, *Challenging common assumptions in the unsupervised learning of disentangled representations*, in International Conference on Machine Learning, 2019, pp. 4114–4124.

[100] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, *The Variational Fair Autoencoder*, ICLR, (2016).

[101] G. Luo, L. Dunlap, D. H. Park, A. Holynski, and T. Darrell, *Diffusion hyperfeatures: Searching through time and space for semantic correspondence*, Advances in Neural Information Processing Systems, (2024).

[102] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, *Lstm-based encoder-decoder for multi-sensor anomaly detection*, arXiv preprint arXiv:1607.00148, (2016).

[103] M. A. Marcinkiewicz, *Building a large annotated corpus of english: The penn treebank*, Using Large Corpora, (1994).

[104] H. Meng, Y. Zhang, Y. Li, and H. Zhao, *Spacecraft anomaly detection via transformer reconstruction error*, in Proceedings of the International Conference on Aerospace System Science and Engineering 2019, Springer, 2020, pp. 351–362.

[105] C. Meo, L. Mahon, A. Goyal, and J. Dauwels, *αTC-VAE: On the relationship between Disentanglement and Diversity*, ICLR, (2024).

[106] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, *Nerf: Representing scenes as neural radiance fields for view synthesis*, Communications of the ACM, 65 (2021), pp. 99–106.

[107] G. Mita, M. Filippone, and P. Michiardi, *An identifiable double VAE for disentangled representations*, in International Conference on Machine Learning, PMLR, 2021, pp. 7769–7779.

[108] S. Mukhopadhyay, M. Gwilliam, Y. Yamaguchi, V. Agarwal, N. Padmanabhan, A. Swaminathan, T. Zhou, and A. Shrivastava, *Do text-free diffusion models learn discriminative visual representations?*, arXiv preprint arXiv:2311.17921, (2023).

[109] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, *Do deep generative models know what they don't know?*, arXiv preprint arXiv:1810.09136, (2018).

[110] A. Nazabal, P. M. Olmos, Z. Ghahramani, and I. Valera, *Handling incomplete heterogeneous data using vaes*, Pattern Recognition, 107 (2020), p. 107501.

[111] R. B. Nelsen, *An introduction to copulas*, Springer science & business media, 2007.

[112] A. v. d. Oord, Y. Li, and O. Vinyals, *Representation learning with contrastive predictive coding*, arXiv preprint arXiv:1807.03748, (2018).

[113] K. Oublal, S. Ladjal, D. Benhaiem, E. L. BORGNE, and F. Roueff, *Disentangling Time Series Representations via Contrastive Independence-of-Support on l-Variational Inference*, ICLR, (2024).

[114] Z. Pan, J. Chen, and Y. Shi, *Masked diffusion as self-supervised representation learner*, arXiv preprint arXiv:2308.05695, (2023).

[115] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference*, Journal of Machine Learning Research, 22 (2021), pp. 1–64.

[116] D. Park, Y. Hoshi, and C. C. Kemp, *A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder*, IEEE Robotics and Automation Letters, 3 (2018), pp. 1544–1551.

[117] G. Parmar, R. Zhang, and J.-Y. Zhu, *On Aliased Resizing and Surprising Subtleties in GAN Evaluation*, CVPR, (2022).

[118] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn, *Diffusion autoencoders: Toward a meaningful and decodable representation*, IEEE Conference on Computer Vision and Pattern Recognition, (2022).

[119] X. Ran, M. Xu, L. Mei, Q. Xu, and Q. Liu, *Detecting out-of-distribution samples via variational auto-encoder with reliable uncertainty estimation*, Neural Networks, 145 (2022), pp. 199–208.

[120] A. Razavi, A. Van den Oord, and O. Vinyals, *Generating Diverse High-Fidelity Images with VQ-VAE-2*, NeurIPS, (2019).

[121] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee, *Deep visual analogy-making*, Advances in neural information processing systems, 28 (2015).

[122] D. Rezende and S. Mohamed, *Variational inference with normalizing flows*, in International conference on machine learning, PMLR, 2015, pp. 1530–1538.

[123] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-resolution image synthesis with latent diffusion models*, IEEE Conference on Computer Vision and Pattern Recognition, (2022).

[124] F. J. Ruiz, M. K. Titsias, T. Cemgil, and A. Doucet, *Unbiased gradient estimation for variational auto-encoders using coupled markov chains*, in Uncertainty in Artificial Intelligence, PMLR, 2021, pp. 707–717.

[125] D. Salinas, M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus, *High-dimensional multivariate forecasting with low-rank gaussian copula processes*, Advances in neural information processing systems, 32 (2019).

[126] H. SHAO, Y. YANG, H. LIN, L. LIN, Y. CHEN, Q. YANG, AND H. ZHAO, *Rethinking controllable variational autoencoders*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19250–19259.

[127] H. SHAO, S. YAO, D. SUN, A. ZHANG, S. LIU, D. LIU, J. WANG, AND T. ABDELZAHER, *Controlvae: Controllable variational autoencoder*, International Conference on Machine Learning, (2020).

[128] S. SHARMA, S. CHAUDHURY, ET AL., *Block sparse variational bayes regression using matrix variate distributions with application to ssvep detection*, IEEE Transactions on Neural Networks and Learning Systems, 33 (2020), pp. 351–365.

[129] S. N. SHUKLA AND B. M. MARLIN, *Heteroscedastic temporal variational autoencoder for irregularly sampled time series*, arXiv preprint arXiv:2107.11350, (2021).

[130] C. K. SØNDERBY, T. RAIKO, L. MAALØE, S. K. SØNDERBY, AND O. WINTHER, *Ladder variational autoencoders*, Advances in Neural Information Processing Systems, (2016).

[131] J. SONG, C. MENG, AND S. ERMON, *Denoising Diffusion Implicit Models*, ICLR, (2021).

[132] Y. SONG, C. DURKAN, I. MURRAY, AND S. ERMON, *Maximum likelihood training of score-based diffusion models*, Advances in neural information processing systems, 34 (2021), pp. 1415–1428.

[133] Y. SONG AND S. ERMON, *Generative modeling by estimating gradients of the data distribution*, Advances in neural information processing systems, 32 (2019).

[134] ——, *Improved techniques for training score-based generative models*, Advances in neural information processing systems, 33 (2020), pp. 12438–12448.

[135] Y. SONG, J. GONG, H. ZHOU, M. ZHENG, J. LIU, AND W.-Y. MA, *Unified Generative Modeling of 3D Molecules with Bayesian Flow Networks*, ICLR, (2024).

[136] Y. SONG, J. SOHL-DICKSTEIN, D. P. KINGMA, A. KUMAR, S. ERMON, AND B. POOLE, *Score-based generative modeling through stochastic differential equations*, arXiv preprint arXiv:2011.13456, (2020).

[137] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, *Robust anomaly detection for multivariate time series through stochastic recurrent neural network*, in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2828–2837.

[138] H. Takahashi, T. Iwata, A. Kumagai, S. Kanai, M. Yamada, Y. Yamanaka, and H. Kashima, *Learning Optimal Priors for Task-Invariant Representations in Variational Autoencoders*, KDD, (2022).

[139] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan, *Emergent correspondence from image diffusion*, Advances in Neural Information Processing Systems, (2023).

[140] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, *Wasserstein auto-encoders*, arXiv preprint arXiv:1711.01558, (2017).

[141] J. Tomczak and M. Welling, *Vae with a vampprior*, International Conference on Artificial Intelligence and Statistics, (2018).

[142] S. Tonekaboni, C.-L. Li, S. O. Arik, A. Goldenberg, and T. Pfister, *Decoupling local and global representations of time series*, International Conference on Artificial Intelligence and Statistics, (2022).

[143] M. Tschannen, O. Bachem, and M. Lucic, *Recent advances in autoencoder-based representation learning*, arXiv preprint arXiv:1812.05069, (2018).

[144] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel, *Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation*, CVPR, (2023).

[145] A. Vahdat and J. Kautz, *Nvae: A deep hierarchical variational autoencoder*, Advances in Neural Information Processing Systems, (2020).

[146] A. Van Den Oord, O. Vinyals, et al., *Neural Discrete Representation Learning*, NeurIPS, (2017).

[147] C. Wang, J. Li, X. Sun, F. Zhang, Y. Yu, and Y. Wang, *Learning domain-agnostic representation for disease diagnosis*, in The Eleventh International Conference on Learning Representations, 2022.

[148] P. Z. Wang and W. Y. Wang, *Neural gaussian copula for variational autoencoder*, arXiv preprint arXiv:1909.03569, (2019).

[149] X. WANG AND J. YIN, *Relaxed multivariate bernoulli distribution and its applications to deep generative models*, in Conference on Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 500–509.

[150] Y. WANG, Y. SCHIFF, A. GOKASLAN, W. PAN, F. WANG, C. DE SA, AND V. KULESHOV, *InfoDiffusion: Representation Learning Using Information Maximizing Diffusion Models*, ICML, (2023).

[151] Y. WANG, H. ZHANG, Z. LIU, L. YANG, AND P. S. YU, *Contrastvae: Contrastive variational autoencoder for sequential recommendation*, in Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 2056–2066.

[152] Z. WANG, X. XU, W. ZHANG, G. TRAJCEVSKI, T. ZHONG, AND F. ZHOU, *Learning latent seasonal-trend representations for time series forecasting*, Advances in Neural Information Processing Systems, 35 (2022), pp. 38775–38787.

[153] C. WEI, K. MANGALAM, P.-Y. HUANG, Y. LI, H. FAN, H. XU, H. WANG, C. XIE, A. YUILLE, AND C. FEICHTENHOFER, *Diffusion models as masked autoencoders*, IEEE International Conference on Computer Vision, (2023).

[154] H. WEN, Y. LIN, Y. XIA, H. WAN, R. ZIMMERMANN, AND Y. LIANG, *Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models*, arXiv preprint arXiv:2301.13629, (2023).

[155] R. WEN AND K. TORKKOLA, *Deep generative quantile-copula models for probabilistic forecasting*, arXiv preprint arXiv:1907.10697, (2019).

[156] G. WOO, C. LIU, D. SAHOO, A. KUMAR, AND S. HOI, *Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting*, arXiv preprint arXiv:2202.01575, (2022).

[157] D. WU, L. WANG, AND P. ZHANG, *Solving statistical mechanics using variational autoregressive networks*, Physical Review Letters, 122 (2019), p. 080602.

[158] Z. WU, L. CAO, AND L. QI, *eVAE: Evolutionary Variational Autoencoder*, TNNLS, (2024).

[159] W. XIANG, H. YANG, D. HUANG, AND Y. WANG, *Denoising Diffusion Autoencoders are Unified self-supervised Learners*, ICCV, (2023).

[160] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-MINST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*, arXiv preprint arXiv:1708.07747, (2017).

[161] T. Z. Xiao and R. Bamler, *Trading information between latents in hierarchical variational autoencoders*, arXiv preprint arXiv:2302.04855, (2023).

[162] Z. Xiao, Q. Yan, and Y. Amit, *Likelihood regret: An out-of-distribution detection score for variational auto-encoder*, Advances in neural information processing systems, 33 (2020), pp. 20685–20696.

[163] Z. Xie, C. Liu, Y. Zhang, H. Lu, D. Wang, and Y. Ding, *Adversarial and contrastive variational autoencoder for sequential recommendation*, in Proceedings of the Web Conference 2021, 2021, pp. 449–459.

[164] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, et al., *Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications*, in Proceedings of the 2018 world wide web conference, 2018, pp. 187–196.

[165] J. Xu and L. Cao, *Copula variational lstm for high-dimensional cross-market multivariate dependence modeling*.
10.1109/TNNLS.2023.3293131, 2023.

[166] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello, *Open-vocabulary panoptic segmentation with text-to-image diffusion models*, IEEE Conference on Computer Vision and Pattern Recognition, (2023).

[167] J. Xu, Y. Ren, H. Tang, X. Pu, X. Zhu, M. Zeng, and L. He, *Multi-VAE: Learning Disentangled View-Common and View-Peculiar Visual Representations for Multi-View Clustering*, ICCV, (2021).

[168] J. Xu, W. Wei, and L. Cao, *Copula-based high dimensional cross-market dependence modeling*, in 2017 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2017, Tokyo, Japan, October 19-21, 2017, 2017, pp. 734–743.

[169] J. Xu, H. Wu, J. Wang, and M. Long, *Anomaly transformer: Time series anomaly detection with association discrepancy*, arXiv preprint arXiv:2110.02642, (2021).

[170] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, *Modeling tabular data using conditional gan*, Advances in neural information processing systems, 32 (2019).

[171] K. Xue, Y. Zhou, S. Nie, X. Min, X. Zhang, J. Zhou, and C. Li, *Unifying Bayesian Flow Networks and Diffusion Models through Stochastic Differential Equations*, ICML, (2024).

[172] L. Yang, W. Fan, and N. Bouguila, *Deep clustering analysis via dual variational autoencoder with spherical latent embeddings*, IEEE Transactions on Neural Networks and Learning Systems, (2021).

[173] L. Yang and S. Hong, *Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion*, in International Conference on Machine Learning, PMLR, 2022, pp. 25038–25054.

[174] T. Yang, X. Ren, Y. Wang, W. Zeng, and N. Zheng, *Towards building a group-based unsupervised representation disentanglement framework*, arXiv preprint arXiv:2102.10303, (2021).

[175] T. Yang, Y. Wang, Y. Lu, and N. Zheng, *Disdiff: Unsupervised disentanglement of diffusion probabilistic models*, Advances in Neural Information Processing Systems, (2023).

[176] F. Ye and A. G. Bors, *Lifelong mixture of variational autoencoders*, IEEE Transactions on Neural Networks and Learning Systems, (2021).

[177] ——, *Continual variational autoencoder learning via online cooperative memorization*, in Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII, Springer, 2022, pp. 531–549.

[178] E. Yeats, F. Liu, and H. Li, *Disentangling learning representations with density estimation*, arXiv preprint arXiv:2302.04362, (2023).

[179] E. Yeats, F. Liu, D. Womble, and H. Li, *Nashae: Disentangling representations through adversarial covariance minimization*, European Conference on Computer Vision, (2022).

[180] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, *Ts2vec: Towards universal representation of time series*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 8980–8987.

[181] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, *A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data*, in Proceedings of the AAAI conference on artificial intelligence, vol. 33, 2019, pp. 1409–1416.

[182] J. Zhang, C. Herrmann, J. Hur, L. Polania Cabrera, V. Jampani, D. Sun, and M.-H. Yang,*A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence*, NeurIPS, (2024).

[183] K. Zhang, Q. Wen, C. Zhang, R. Cai, M. Jin, Y. Liu, J. Zhang, Y. Liang, G. Pang, D. Song, et al., *Self-supervised learning for time series analysis: Taxonomy, progress, and prospects*, arXiv preprint arXiv:2306.10125, (2023).

[184] W. Zhang, C. Zhang, and F. Tsung, *Grelen: Multivariate time series anomaly detection from the perspective of graph relational learning*, in Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, vol. 7, 2022, pp. 2390–2397.

[185] Y. Zhang, Y. Wang, L. Zhang, Z. Zhang, and K. Gai, *Improve diverse text generation by self labeling conditional variational auto encoder*, in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 2767–2771.

[186] H. Zhao, D. Wang, and H. Lu, *Representation Learning for Visual Object Tracking by Masked Appearance Transfer*, CVPR, (2023).

[187] S. Zhao, J. Song, and S. Ermon, *Infovae: Information maximizing variational autoencoders*, arXiv preprint arXiv:1706.02262, (2017).

[188] ——, *InfoVAE: Balancing learning and inference in variational autoencoders*, in Proceedings of the aaai conference on artificial intelligence, vol. 33, 2019, pp. 5885–5892.

[189] W. ZHAO, Y. RAO, Z. LIU, B. LIU, J. ZHOU, AND J. LU, *Unleashing text-to-image diffusion models for visual perception*, IEEE International Conference on Computer Vision, (2023).

[190] B. ZHOU, S. LIU, B. HOOI, X. CHENG, AND J. YE, *Beatgan: Anomalous rhythm detection using adversarially generated time series.*, in IJCAI, vol. 2019, 2019, pp. 4433–4439.

[191] M. ZHOU, T. CHEN, Z. WANG, AND H. ZHENG, *Beta Diffusion*, NeurIPS, (2023).