THE UNIVERSITY *of* EDINBURGH

Geometry for Deep Representation Learning

Mohammad Asif Khan

Doctor of Philosophy
Institute for Adaptive and Neural Computation
School of Informatics
University of Edinburgh
2023

Abstract

Deep representation learning has achieved remarkable success in discovering meaningful low-dimensional features from high-dimensional data in recent years. In datasets containing face images, these features can capture underlying factors of variations, such as age, eye colour, and hairstyle. We can employ learned representations for solving tasks such as face detection. By capturing these factors of variations, the representations aim to build a model of the real world, reflecting its inherent regularities. However, current approaches still face challenges when it comes to discovering complex regularities of the world in a data-efficient way, resulting in a lack of interpretability, robustness and limited generalisation.

Recognising that real-world data spaces often exhibit regularities characterised by various symmetries that need appropriate modelling assumptions is crucial. Consider an image of an apple; we know its transformation under a translation operator will not change its identity as an apple. Such properties that do not change under a broad family of transformations are known as *invariants*.

"Geometry is a study of invariants"– Felix Klein (Klein, 1872).

In this thesis, we utilise geometry as a fundamental principle to account for relevant properties in learning representation space. Specifically, we propose novel methodologies to address three main challenges in deep representation learning: learning disentangled latent factors for image sequences, investigating the robustness of deep latent factor models to adversarial perturbations, and learning representations that account for hierarchical dependencies in heterophilic graphs.[1]

The first project focuses on learning to disentangle *content* and *motion* information into separate latent components for image sequences. Here, *content* refers to information shared across all frames, for example, the identity of an object undergoing the dynamics, and *motion* refers to information expressed in a given sequence frame. The temporal structure in image sequences traces a path in a higher dimensional data space that takes the form of a 1-dimensional manifold. A key challenge in learning representations from this data is designing a latent dynamical model that accounts for the temporal structure of image sequences. In this work, we utilise symplectic geometry in latent space for modelling the dynamics of various motions; this structure in latent space associates a motion with a constant energy term that captures the manifold of the dynamics of sequences. For a set of dynamical actions, we associate each with a unique subspace that reflects the energy preservation of a respective dynamical action. Our results demonstrate that we can disentangle factors of variations, facilitating tasks such as controlled generation and motion transfer.

The second contribution proposes a robustness analysis of an oft-used representation learning framework, namely variational autoencoders (VAEs). It is vital that VAEs are built to be reliable,

[1] A heterophily is a property where two non-local nodes in a graph exhibit similarity due to their local neighbourhood connectivity structure.

primarily for their real-world applications, such as latent space control in robotics or in a medical domain for designing novel molecules by exploring the latent space. We examine latent space from a geometric standpoint and establish a connection between the vulnerability of VAEs to adversarial perturbations and the structure of the latent space. Our findings show that the learned latent manifold has a high curvature with low/zero density regions, making VAEs susceptible to adversarial attacks. We propose quantitative scores for measuring robustness and a simple training mechanism for enhancing it.

Lastly, we target the challenge of representation learning for data on graph domains with a heterophily property. In heterophilic graphs, the nodes not in an immediate vicinity may share the same label due to their similar local connectivity structure. For example, in an academic network, two researchers in different countries can exhibit similar local connectivity due to the nature of their profession. We use diffusion geometry to explicitly model hierarchical dependencies in a graph in the form of augmentations. We then use these augmentations in a contrastive setup for learning representations of nodes in a graph. These representations can facilitate various downstream tasks, including graph classification, link prediction, and community detection. Our results showcase the effectiveness of augmentations in allowing the encoder to capture hierarchical dependencies, demonstrated by improved performance on several benchmark datasets.

In summary, through three core contributions, this thesis shows the importance of incorporating geometry-based inductive biases into deep representation learning models to develop efficient and reliable applications.

Lay Summary

Deep representation learning aims to extract useful features from complex high-dimensional data. For example, it can figure out elemental features like age, eye colour, and hairstyle from a data set of facial images that we can use for end tasks such as face detection. However, to learn such features, it is essential to build assumptions that account for the regularities in the data set. For example, a person's identity will not change under a transformation like translation in a data set of face images. Such properties that do not change under a broad family of transformations are known as *invariants*. Existing methods often fail to encode suitable invariants in the features, which can limit their generalisation performance or require more training data, leading to sample inefficiency.

This thesis emphasises the importance of modelling invariants in complex data spaces utilising geometric principles, addressing three primary challenges: smoothness of latent space, disentanglement of factors of variation, and encoding hierarchical dependencies in the representation space.

In a non-smooth latent space, a slight change in the data sample can arbitrarily map it to different features in the latent space. The first research focuses on learning a latent space with smoothness property using the time axis in image sequences data. This approach further disentangles content and motion information in image sequences, facilitating controlled generation and motion transfer. The second project investigates the non-smoothness of latent space using differential geometry for the oft-used variational autoencoders (VAEs) and introduces evaluation scores to quantify such effects. The final project targets encoding hierarchical dependencies in the representation space, which often appear in graph data with a heterophily property where nodes not in an immediate vicinity have the same label. The proposed approach learns to embed nodes in a graph to feature space so that nodes with heterophily are mapped nearby in the latent space.

In summary, this thesis highlights the significance of integrating geometry-based inductive biases into deep representation learning to encourage suitable properties in the latent space, improving sample efficiency and generalisation performance.

Acknowledgements

I am grateful to acknowledge the support of various individuals who made it possible for me to write this thesis. I am incredibly thankful to my advisor, Professor Amos Storkey, for his mentorship and regular interaction, which helped sharpen my research skills. I thank Professor Chris Williams and Professor Subramanian Ramamoorthy for their insightful feedback during my annual review meetings.

I am thankful to the Bayesian and Neural Systems research group for their helpful comments in group meetings and the enjoyable conversations at the informal gatherings at the pub. I express my gratitude to William Toner and Dr Joe Mellor for their feedback and discussions, which significantly contributed to the success of my research. I am also thankful to my office mates, Andreas Grivas, Nick McKenna, Eric Munday, and Jesse Sigal, for their constant support and enjoyable conversations during my PhD journey.

I thank the Institute of Adaptive and Neural Computation, University of Edinburgh, for providing a dynamic research environment and resources to conduct my research. I thank Dr Haitham Bou-Ammar for hosting me as an intern in his research team at Huawei R&D in London. I am grateful to Professor Jens Lehmann, Dr Fabien Cardinaux, and Professor Asja Fischer, who provided invaluable mentorship during my MSc studies and supported my research career.

I would like to take this opportunity to express my deepest gratitude to my parents, who have always been my unwavering source of support and unconditional love. I feel incredibly fortunate to have such caring and loving sisters who have been my pillars of strength throughout my life. Their endless encouragement and care have been a constant source of motivation for me and helped me reach where I am today. I am also deeply grateful to my brother, nieces, and nephews for their love and support, which has brought immense joy and reminded me of the importance of family in my academic pursuits.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Mohammad Asif Khan*)

Table of Contents

List of Figures

List of Tables

List of Symbols and Notations

$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
$\mathbb{E}$	Expectation
$\mathbf{X}$	Input space
$\mathbf{Z}$	Latent space
$\mathbf{x}$	Data point in $\mathbf{X}$
$\mathbf{x}_{1:T}$	Sequence of input data points $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$
$\mathbf{z}$	Latent representation of $\mathbf{x}$
K	Number of actions
$\mathbf{u}$	Action label of an input sequence
$\mathbf{S}$	Motion space
$\mathbf{Q}$	Position phase space
$\mathbf{P}$	Momentum phase space
$\mathbf{s}$	Phase coordinate
$\mathbf{p}$	Momentum coordinate
$\mathbf{q}$	Position coordinate
$\mathbf{S}^k$	k^{th} subspace of $\mathbf{S}$
G	Group
$SO(2)$	Group of rotations in a 2-dimensional plane
$R(\theta)$	Rotation matrix that rotates by an angle θ in a dimensional plane
$\mathfrak{g}$	Lie algebra of a Lie group G
$Sp(2d)$	Symplectic Lie group in a 2d-dimensional space
$\mathfrak{sp}$	Lie algebra of a $Sp(2d)$
$\circ$	Group action
$\mathbf{H}$	Real Hamiltonian Matrix of shape $2d \times 2d$
$\mathbf{I}_m$	Identity matrix of shape $m \times m$
$\mathbf{M}$	Symmetric matrix
$\mathbf{E}$	Hamiltonian energy
$\mathbf{J}$	Skew-symmetric bilinear form matrix of shape $2d \times 2d$

$\mathcal{N}$	Gaussian distribution
μ	Mean of Gaussian distribution
σ	Standard deviation of Gaussian distribution
$\mathcal{U}$	Uniform distribution
$\mathcal{B}$	Beta distribution
ε	Spherical Gaussian noise drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
f_θ	Neural-network mapping with parameters θ
p	Probability distribution
q	Variational distribution
Tr	Trace of a matrix
H	Entropy
KL	Kullback Leibler divergence
$d(.,.)$	Distance function
$\|\|.\|\|_p$	L^p norm
$\mathcal{M}$	Riemannian manifold
$\mathcal{T}_x\mathcal{M}$	Local tangent space at x of a Riemannian manifold
$\mathbf{J}_{f_\theta(\mathbf{x})}$	Jacobian matrix under mapping f_θ, where i, j entry is a gradient of $i-$th output unit w.r.t $j-$th input unit
$\mathbf{G}_\mathbf{x}$	Pullback metric tensor at input point $\mathbf{x}$
$\hat{\mathbf{G}}_\mathbf{x}$	Stochastic pullback metric tensor at input point $\mathbf{x}$
η	Adversarial noise
λ	Eigenvalue
η_0, δ, υ	Real-valued constant numbers
$\rho(.)$	Spectral radius of a matrix
$\mathcal{S}(.)$	Von Neumann Entropy of a matrix
$\mathbf{A}$	Adjacency matrix
$\tilde{\mathbf{A}}$	Column normalised adjacency matrix
$\mathbf{F}$	Node feature representation matrix
$\mathbf{T}$	Lazy diffusion filter
$\mathbf{\Phi}$	Diffusion filters
C	Graph corruption function
D	Discriminator function
$\mathbb{I}(.)$	Indicator function
α	Learning rate

Chapter 1

Introduction

Learning low-dimensional representations of high-dimensional data sits at the core of machine learning. The goal of representation learning is to design a mapping to project data into an embedding space that ultimately captures sufficient information required for solving a broad range of downstream tasks. For example, in a face detection task, we desire representations to encode features such as eye colour, skin tone and eyebrows. However, we must make suitable assumptions about the data space to facilitate learning of a meaningful representation space. To illustrate this point, imagine we are given a set of greyscale images, each with 256×256 pixels; we can view these images as points in $\mathbb{R}^{65536}$ dimensional space. Now, if each pixel can take 1 of 256 greyscale values, there are 256^{65536} possible image configurations needed to describe dependencies among data variables, making it an impossible task.

Deep neural networks (DNNs) build on various assumptions about the regularities in data space, a.k.a *inductive biases* to design complex non-linear transformations for learning representations. The properties of these learned representations are closely tied to the *inductive biases* used in designing DNN architectures or optimisation strategies. For instance, convolutional neural networks (CNNs) exploit local connectivity in input and are translation invariant, which is desirable for tasks such as face classification. However, when encountering data with invariants to other symmetry transformations like rotation or reflection, CNNs are unsuitable. Similarly, graph neural networks (GNNs) operate on discrete objects such as graphs and can account for the complex relationship among nodes. GNNs are invariant to permutations of nodes, which again is a desired property for graph datasets. Therefore, it is critical to identify properties we want in representations for their better generalisability and applications to a broad range of tasks. By exploring the suitable inductive biases, we can tailor representations to capture the desired properties effectively, including invariance to symmetry transformations.

This thesis adopts geometric principles as a foundation for exploring properties that facilitate learning good representations. Each chapter identifies a problem within a representation learning paradigm and proposes a method for addressing it. Before outlining the technical contributions

in detail, we will briefly examine essential properties that characterise good representations.

1.1 What Constitutes a Good Representation?

Bengio et al. (2013a) provided an overview of various prior assumptions characterising good representations. These assumptions encompass essential properties, to name a few, such as *manifold hypothesis*, which states that high-dimensional data generally lives near a much lower dimensional manifold;[1] *smoothness*, dictating that small changes in inputs should correspond to small changes in their encoding; *disentanglement*, which implies that individual or groups of latent factors capture the explanatory factors of the data; *simplicity*, ensuring that features are easily interpretable such as linked by linear dependencies; and *temporal and spatial coherence*, meaning that latent factors should change gradually for observations that are spatially or temporally close. Achille and Soatto (2018) have discussed some of these properties from an information-theoretic perspective. Much recently, Higgins et al. (2018) emphasised that representations should capture symmetry transformations as a crucial aspect of modelling the real world.

Our work builds in the context of the above properties with applications to three distinct scenarios. Chapter 3 explores the representation learning of image sequences, where we target disentanglement and temporal coherence in the latent space using a symplectic structure.[2] Moving forward, Chapter 4 employs Riemannian geometry to examine the smoothness of learned latent manifolds, aiming to establish their robustness against adversarial attacks. Last, Chapter 5 tackles the challenge of capturing hierarchical dependencies in graph-structured datasets using diffusion geometry. All three studies demonstrate the relevance of geometry principles for learning informative representations that lead to improved performance across a range of tasks.

Next, we will provide a brief overview of some of the properties relevant to the scope of this thesis. Afterwards, we will present a detailed discussion of specific research contributions.

1.1.1 Manifold Hypothesis

The manifold hypothesis implies the presence of geometric structure in a data space that emerges due to various degrees of freedom. Consider an example of a set of images recorded by a camera revolving in a fixed orbit around an individual. Let us assume this individual is enacting an action of squats. Squatting is a periodic motion where dynamics periodically return to the same

[1] In lay terms, we can think of a manifold as a topological space where for all points, a local line segment around those looks like Euclidean space. Consider a circle; it locally looks like $\mathbb{R}^1$; therefore, we call it a $1-$manifold.

[2] To put simply, symplectic refers to the geometry of the phase-space of smooth manifolds and provides a framework for understanding the conservation of specific quantities in dynamical systems. We introduce a formal definition in Background Chapter 2.

pose. Likewise, the revolving camera returns to the same position. The two periodicities mean the data manifold takes the form of a torus. Developing neural networks that can capture such regularities in data could be vital for understanding their generalisation and robustness. Let us assume we are given a trained representation learning model (on a finite set of sequences) and were to use it for the downstream application of action classification. Will the representations generalise to sequences recorded from unseen viewpoints of a camera? Such generalisation is possible if the representations reflect the geometry of data. The presence of regularities in the dataset allows us to build appropriate inductive biases in our learning framework, making it possible to learn from data.

1.1.2 Smoothness

The notion of smoothness for a DNN generally means that when the input is subjected to slight variations, the output should change slowly. If the representation space is smooth, small perturbations in the input correspond to a smooth transition locally in the representation space. This property allows representation learning models to generalise well to noisy data, reduces the risk of overfitting and serves as a defence mechanism against adversarial perturbations that target to achieve abrupt changes in the latent encodings.

In Chapter 4, we use pullback geometry to demonstrate the non-smoothness of the latent space of VAEs that potentially makes them vulnerable to adversarial attacks. Furthermore, we show that an alternative optimisation strategy of $\beta-$VAE that smooths the latent space for an increasing value of β improves the robustness of representations to such attacks.

1.1.3 Disentanglement

Representations serve as models of the world, capturing complex interactions between entities and their hierarchical relationships. A crucial aspect of representations is their ability to associate diverse input data attributes with a distinct group of latent components. This association provides a physical understanding of the underlying data using latent factors of variations. To illustrate this, let us consider a scenario where images are mapped to a feature representation using a black-box model. Imagine we are presented with two images of a ball, one coloured red and the other blue, placed against a white background. We can map these images using the black-box model to obtain their corresponding feature representations. In an ideal scenario, the resulting representations should maintain relevant *invariances* and *equivariance*. The *invariance* is a property that does not change under nuisance transformations, such as the class information (i.e., identifying an image as a ball) to differentiate it from other objects. The *equivariance* is a property that maintains the sensitivity of representations to certain transformations, such as colour information, to distinguish a red ball from balls of different colours. An approach where factors of variations reflect salient features of the data can facilitate improved performance. For

example, different colours or shapes, even if they were not encountered during training. This generalisation capacity is crucial for real-world applications where the model needs to handle novel data instances.

Analysing the components of the representation corresponding to the ball's colour can provide insights into the interpretability of the neural network used for parameterising the representation space. This interpretability allows for a deeper understanding of the learned features and their capacity to reflect the complex dependencies among attributes in the dataset.

Discovering factors of variation is particularly important in deep generative modelling. Probabilistic frameworks, such as VAEs, are widely used for this purpose. VAEs simultaneously learn the encoding of data and its generative model. The disentangled representations enable us to manipulate and control specific attributes of the data point. In the case of the ball example, having distinct components encode the class and colour allows us to generate new instances of balls with different colours or classify balls based on their colour more accurately.

1.1.4 Temporal Coherence

The general idea here is that the representations should change slowly for temporally nearby frames. This property is of particular relevance in temporal data such as image sequences. For example, let us say we have a sequence describing an individual's walking motion; the dynamics evolve gradually, with previous poses influencing the pose at the next step. Representations should reflect such a notion of proximity to avoid abrupt and implausible changes, ensuring that sudden disappearances or physically unrealistic actions do not occur.

By ensuring that temporal variations in a sequence correspond to gradual changes in the latent space, temporal coherence promotes the inherent property of smoothness, enabling the learning of coherent and robust representation space.

Chapter 3 of this thesis demonstrates the importance of symplectic geometry in capturing this temporal coherence in image sequences. By incorporating the Hamiltonian operator as a latent dynamics model, a plausible future timestep is the one given by constant energy, eliminating any abrupt changes. Moreover, the symplectic phase space geometry utilised in representation learning proves valuable in disentangling temporally independent information from time-dependent details. This disentanglement separates content-related factors, such as the individual's appearance, from time-dependent factors, such as the specific motion being performed. By capturing this distinction, the representations provide a richer understanding of the underlying dynamics and facilitate the modelling and analysis of temporal data.

Using symplectic geometry to enforce temporal coherence and disentangle relevant information in the latent factors enhances the fidelity and interpretability of representations in the context of image sequences, contributing to improved performance on tasks such as action recognition, video synthesis, and motion transfer.

1.1.5 Hierarichal Dependencies

In several domains, particularly ones characterised by graph-structured data, the heterophily property of graph often results in multiple levels of hierarchical dependencies (Bronstein et al., 2017; Liu et al., 2021). The heterophily is a property of a graph where nodes that are not directly connected share the same role in a graph due to their local connectivity pattern given by a k-hop neighbourhood of a node. The ability of representations to effectively encode these complex dependencies is crucial for addressing various downstream tasks. For example, consider the task of fraudster detection in social network graphs. In such networks, scammers may exhibit similar patterns of behaviour reflected in their local connectivity within a network. Although direct connections between scammers may be rare, they can still be grouped based on their shared activity. Capturing such long-term hierarchical dependencies is essential for detecting and avoiding scams in social networks.

Chapter 5 employs diffusion geometry to learn node representations that capture the long-term complex interdependencies among nodes in heterophilic graphs, achieving improved performance on downstream classification tasks.

1.2 Research Contributions

In summary, the central theme of this thesis is to demonstrate the role of geometry in learning effective representations that capture desirable properties. By leveraging geometric tools, we can encode appropriate invariances, ensure smoothness in the learned models, and effectively handle data with complex geometries, improving performance across diverse tasks. This section outlines the specific research contributions building upon these foundational concepts. First, we look at disentangling latent factors of variation in the image sequences dataset. Second, we investigate the robustness of the latent space in variational autoencoders (VAEs). Last, we propose a novel augmentation technique inspired by diffusion geometry for graph representation learning. Specifically, we consider the following issues.

1.2.1 Disentanglement of *Content* and *Motion* in Image Sequences

Latent variable models that can effectively decompose *content* and *motion* factors in image sequences offer practical applications in areas such as motion transfer and controlled generation. However, a significant challenge lies in capturing the one-dimensional manifold that arises from the temporal evolution of the sequence. It is vital to develop models that can capture the underlying dynamics to generate sequences with coherent frames and enable the separation of various actions within a motion space. By addressing this challenge, we can unlock the potential of latent variable models in producing high-quality and interpretable sequential data.

We introduce *HALO* – a deep generative model utilising Hamiltonian latent Operators to reliably disentangle content and motion information in image sequences. The *content* represents summary statistics of a sequence, and *motion* is a dynamic process that determines how information is expressed in any part of the sequence. By modelling the dynamics as a Hamiltonian motion, important desiderata are ensured: (1) the motion is reversible, (2) the symplectic, volume-preserving structure in phase space means paths are continuous and are not divergent in the latent space. Consequently, the nearness of sequence frames is realised by the nearness of their coordinates in the phase space, which proves valuable for disentanglement and long-term sequence generation. The sequence space is generally comprised of different types of dynamical motions. To ensure long-term separability and allow controlled generation, we associate every motion with a unique Hamiltonian that acts in its respective subspace. We demonstrate the utility of *HALO* by swapping the motion of a pair of sequences, controlled generation, and image rotations.

1.2.2 Robustness of Deep Latent Variable Models

Variational Autoencoders (VAEs) (Kingma and Welling, 2014) are frequently used deep latent variable models; several recent works have investigated the robustness of their representation space to adversarial attacks (Willetts et al., 2021; Kuzina et al., 2021). Given a pre-trained VAE in an unsupervised learning problem, an adversary aims to find a small perturbation in an input sample that significantly changes its latent space encoding, thereby compromising the reconstruction. A known reason for such vulnerability is the distortions in the latent space resulting from a mismatch between approximated latent posterior and a prior distribution. Consequently, a slight change in an input sample can move its encoding to a low/zero density region in the latent space, resulting in an unconstrained generation. This project demonstrates that an optimal way for an adversary to attack VAEs is to exploit a directional bias of a stochastic pullback metric tensor induced by the encoder and decoder networks.

The pullback metric tensor of an encoder measures the change in infinitesimal latent volume from an input to a latent space. Thus, it can be viewed as a lens to analyse the effect of input perturbations leading to latent space distortions. We propose robustness evaluation scores using the eigenspectrum of a pullback metric tensor. Moreover, we empirically show that the scores correlate with the robustness parameter β of the $\beta-$VAE. Since increasing β also degrades reconstruction quality, we demonstrate a simple alternative using *mixup* training to fill the empty regions in the latent space, thus improving robustness with improved reconstruction.

1.2.3 Hierarchical Dependencies in Heterophilic Graphs

Learning node-level representations of heterophilic graphs is crucial for various applications, including fraudster detection and protein function prediction. In such graphs, nodes share

structural similarity identified by the equivalence of their connectivity, which is implicitly encoded in the form of higher-order hierarchical information in the graphs.

The contrastive methods are popular choices for learning the representation of nodes in a graph. However, existing contrastive methods struggle to capture higher-order graph structures. To address this limitation, we propose a novel multiview contrastive learning approach that integrates diffusion filters on graphs. By incorporating multiple graph views as augmentations, our method captures the structural equivalence in heterophilic graphs, enabling the discovery of hidden relationships and similarities not apparent in traditional node representations. Our approach outperforms baselines on synthetic and real structural datasets, surpassing the best baseline by 16.06% on Cornell, 3.27% on Texas, and 8.04% on Wisconsin. Additionally, it consistently achieves superior performance on proximal tasks, demonstrating its effectiveness in uncovering structural information and improving downstream applications.

1.3 Publications

The thesis encompasses three technical contributions, two previously published in conference proceedings. The respective publications of chapters are:

- **Chapter 3** introduces an approach for disentanglement of content and motion in image sequences based on the publication,

  Khan, Asif, and Amos Storkey, "Hamiltonian Latent Operators for Content and Motion Disentanglement in Image Sequences". *The Thirty-sixth Conference on Neural Information Processing Systems, 2022.*

- **Chapter 4** presents a geometrical perspective on the robustness of VAEs based on,

  Khan, Asif, and Amos Storkey, "Adversarial robustness of VAEs through the lens of local geometry." *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics, PMLR 206:8954-8967, 2023.*

- **Chapter 5** proposes an approach for representation learning of nodes in heterophilic-graph based on a preprint,

  Khan, Asif, and Amos Storkey, "Contrastive Learning for Non-local Graphs with Multi-Resolution Structural Views", *arXiv preprint (2023)*

- During my PhD, I had an opportunity to work with the Bayesian Optimisation and Reinforcement Learning Team at Huawei's Noah's Ark Lab, London. I developed a combinatorial Bayesian optimisation for the protein engineering problem with application to designing antibody sequences. The short version of the findings was presented at the Computation Biology Workshop at ICML 2022.

Khan, Asif, Alexander I. Cowen-Rivers, Antoine Grosnit, Philippe A. Robert, Victor Greiff, Eva Smorodina, Puneet Rawat, Rahmad Akbar, Kamil Dreczkowski, Rasul Tutunov, Dany Bou-Ammar, Jun Wang, Amos Storkey, Haitham Bou-Ammar, "Toward real-world automated antibody design with combinatorial Bayesian optimisation." *Cell Reports Methods (2023): 100374.*

1.4 Thesis Structure

The structure of the rest of the dissertation is as follows:

Chapter 2 provides the necessary background for the thesis. It begins with a concise introduction to latent variable models and their integration with deep learning techniques. The chapter also covers the fundamental concepts of geometry and its application in representation learning. Additionally, the evaluation metrics used in this work are explained.

Chapter 3 focuses on addressing the problem of latent factor disentanglement in image sequence datasets. A novel methodology is introduced, which employs symplectic geometry in the latent space to model the dynamics of image sequences. The chapter highlights the significance of incorporating symplectic geometry for achieving coherent frame generation and the separability of different actions in the motion space.

Chapter 4 investigates the robustness of VAEs, a commonly used deep latent factor model. From a geometrical perspective, the chapter explores the robustness of VAEs against adversarial perturbations and proposes a novel mechanism for measuring robustness using pullback geometry.

Chapter 5 presents a representation learning framework tailored for heterophilic graphs. It proposes a methodology that uses diffusion geometry to leverage the hierarchical information embedded in graphs. This approach captures the structural equivalence among nodes and learns representations that unveil hidden relationships and similarities.

Chapter 6 concludes the thesis by summarising the contributions made, discussing their broader impact, and outlining potential directions for future research.

1.5 Summary

This thesis explores the role of geometrical inductive biases in understanding the structure of complex data spaces and developing sample-efficient and robust deep representation learning approaches. It focuses on three significant problems. First, learning disentangled representations for image sequences enables controlled generation and motion transfer. Second, is a robustness analysis of VAEs using pullback geometry to enhance reliability and resilience to adversarial attacks. Last, the diffusion-based augmentation strategy for learning multiscale information in a

graph, which we demonstrate, is essential for classification problems on heterophilic graphs.

Chapter 2

Background

This chapter presents essential background information to build the foundation of this thesis. It is organised into two sections. The first section introduces the fundamental mathematical concepts used as a foundation across the three projects conducted in this thesis. The second section offers a comprehensive overview of different representation learning approaches encompassing traditional and deep learning paradigms. We provide a broader overview here. The terminologies and related work specific to individual projects are introduced in their respective chapters.

2.1 Mathematical Preliminaries

In this section, we introduce the necessary definitions that hold relevance across all three projects of the thesis. The technical description draws inspiration from Tu (2011) and Lee (2006).

2.1.1 Manifolds

The geometrical spaces we are most interested in our work are known as *manifolds*. Informally, a manifold is a topological space that locally looks like Euclidean space. To establish it formally, we first introduce key definitions.

Definition 1. A topology on a set $\mathcal{X}$ is a subset $\mathcal{U}$ of the power set of $\mathcal{X}$: $\mathcal{U} \subseteq 2^{\mathcal{X}}$ satisfying:

- the empty set $\emptyset$ and a set $\mathcal{X}$ are in the topology: $\emptyset, \mathcal{X} \in \mathcal{U}$
- given any pair of set U_1, U_2 in topology their intersection $U_1 \cap U_2$ is also in the topology $\mathcal{U}$.
- given a collection of arbitrary sets $\{U_\alpha\} \subseteq \mathcal{U}$ in the topology the union $\cup_{i \in \alpha} U_i \in \mathcal{U}$ is also in the topology.

The set $O \in \mathcal{X}$ is called an open set, and the complement $\mathcal{U} \setminus O \in \mathcal{X}$ is called a closed set.

Definition 2. The pair $(\mathcal{U}, \mathcal{X})$ is a topological space with $\mathcal{U}$ a topology of $\mathcal{X}$.

By utilising the above axioms of topology, we can define various structures on the set $\mathcal{X}$. For instance, consider the set $\mathcal{X} = \{1,2,3\}$. The set $\mathcal{U} = \{\emptyset, \mathcal{X}, \{1\}, \{1,2\}\}$ is an example of one such topology on the set $\mathcal{X}$.

Definition 3. For a pair of topological spaces $(\mathcal{U}, \mathcal{X})$ and $(\mathcal{V}, \mathcal{Y})$ a map $f : \mathcal{X} \to \mathcal{Y}$ is called continuous w.r.t topology $\mathcal{U}$ and $\mathcal{V}$ if for every open set $Y \in \mathcal{Y}$, $f^{-1}(Y) \in \mathcal{X}$ is an open set.

Definition 4. For a pair of spaces $\mathcal{X}$ and $\mathcal{Y}$, a homeomorphism $f : \mathcal{X} \to \mathcal{Y}$ is a one-one onto function where f and f^{-1} are continuous. Then, the space $\mathcal{X}$ and $\mathcal{Y}$ are said to be homeomorphic.

Definition 5. A topological space is called an $n-$dimensional manifold $\mathcal{M}$ if $\forall m \in \mathcal{M}$ there exists a neighborhood region V_m of m such that we can define a homeomorphism $f : V_m \to \mathbb{R}^n$.

A *local neighborhood* V_m of m is a pair (V_m, ϕ), where V_m is an open set containing m and ϕ is a chart on V_m, i.e., $\phi : \hat{V}_m \to V_m$, where $\hat{V}_m$ is an open set in $\mathbb{R}^n$ for some n and V_m is an open set in $\mathbb{R}^n$, such that m is mapped to the origin, i.e., $\phi(m) = \mathbf{0}$ in $\mathbb{R}^n$. Some commonly known examples are $\mathbb{R}^n$, which is homeomorphic to itself, the 1-Sphere $\mathbb{S}^1$, which locally looks like $\mathbb{R}$ or a 2-Sphere $\mathbb{S}^2$ which locally looks like $\mathbb{R}^2$. The local regions are also known as charts; we will use homeomorphisms to define them explicitly.

Definition 6. Given a point m on a n-dimensional manifold $\mathcal{M}$ and its neighbourhood set V_m, a chart is a homeomorphism $\phi_m : V_m \to \mathbb{R}^n$. The collection of charts $\{(V_m, \phi_m)\}$ that covers the manifold is called an atlas $\mathcal{A}$, and for $V_m \cap V_k \neq \emptyset$, the chart transition map is defined as $\phi_{m \to k} : \phi_m(V_m \cap V_k) \to \phi_k(V_m \cap V_k)$.

A chart of a manifold is a coordinate representation in Euclidean space; thus, we can use calculus to establish properties like continuity or differentiability locally on a manifold. If a chart representation is continuous and differentiable, we can conclude the manifold is continuous and differentiable. An important point to note is that such properties of the manifold are independent of the choice of charting function.

The chart transition map is used to understand how a representation of manifold changes from one coordinate system to another. Thus, it contains the structural information on how to combine or glue the charts of an atlas to obtain a global representation of a manifold in $\mathbb{R}^n$.

Definition 7. A smooth manifold $\mathcal{M}$ is a topological space iff two charts (ϕ_1, V_1), (ϕ_2, V_2) are smoothly compatible. That is to say chart transition map $\phi_{1 \to 2} : \phi_1(V_1 \cap V_2) \to \phi_2(V_1 \cap V_2)$ is a diffeomorphism (if it is smooth, bijective and invertible meaning the inverse $\phi_{1 \to 2}^{-1}$ exists and is smooth).

In general, topological manifolds can have a notion of curvature where using an Euclidean metric as a measure of distance can lead to a misleading interpretation of its underlying geometry.

Riemannian geometry focuses on the study of smooth manifolds equipped with metrics for measuring distance on an arbitrary curved space.

Definition 8. A Riemannian metric for a smooth manifold $\mathcal{M}$ is a bilinear, symmetric, positive definite map $\mathbf{G}_{\mathbf{x}} : \mathcal{T}_{\mathbf{x}}\mathcal{M} \times \mathcal{T}_{\mathbf{x}}\mathcal{M} \to \mathbb{R}$ for all $\mathbf{x} \in \mathcal{M}$, where $T_{\mathbf{x}}\mathcal{M}$ is a tangent plane at point $\mathbf{x}$ on the manifold (Lee, 2006).

Definition 9. A smooth manifold $\mathcal{M}$ with a Riemannian metric $\mathbf{G}$ defined on every point of a manifold is called a Riemannian Manifold (Lee, 2006).

Definition 10. Given a mapping $f : \mathcal{M} \to O$ from smooth manifold $\mathcal{M}$ to smooth manifold O, for any $\mathbf{x} \in \mathcal{M}$ the pull-back metric $\mathbf{G}_{\mathbf{x}}$ induced by the mapping f is given as $\mathbf{G}_{\mathbf{x}} = \mathbf{J}_{f(\mathbf{x})}^T \mathbf{G}_{f(\mathbf{x})} \mathbf{J}_{f(\mathbf{x})}$, where $\mathbf{J}_{f(\mathbf{x})} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$.

2.1.2 Lie Groups

The concept of symmetry refers to a transformation that preserves certain properties of an object, such as its shape or arrangement. Common examples of symmetries include translation (shifting in space) and rotation. The study of symmetries is significant in understanding physical systems and uncovering the constants that govern them. For instance, the presence of space translation symmetry corresponds to the conservation of linear momentum, while rotation symmetry corresponds to the conservation of angular momentum. Groups are employed as fundamental tools to analyse and describe these symmetries. Groups provide a formal framework for studying the transformations and symmetries that underlie various physical phenomena. Formally, we say,

Definition 11. A group G is a set with a binary operation $*$ satisfying the following conditions:

- closure under $*$, i.e., $x * y \in G$ for all $x, y \in G$.
- there is an identity element $e \in G$, satisfying $x * e = e * x = x$ for all $x \in G$.
- for each element $x \in G$ there exist an inverse $x^{-1} \in G$ such that $x * x^{-1} = x^{-1} * x = e$.
- for all $x, y, z \in G$ the associative law holds i.e. $x * (y * z) = (x * y) * z$.

Classifying a group as discrete or continuous depends on the nature of the symmetry exhibited by the system.

A discrete group possesses a finite number of elements. For example, the dihedral group is generated by three elements: the identity e, a rotation r by an angle of π, and a reflection f along the x-axis. The group consists of the finite elements $\{e, r, f, rf\}$. The group generators are a set of elements that can produce other elements within the group through the group multiplication rule. In the case of a dihedral group, the generators are $\{e, r, f\}$. In contrast, a continuous group

is characterised by infinitesimal transformations and is referred to as a Lie group. Lie groups play a crucial role in studying continuous symmetries and are widely used in various branches of physics and mathematics.

Definition 12. A Lie group, G, is a group which also forms a smooth manifold structure, where the group operations under multiplication $G * G \rightarrow G$ and its inverse $G \rightarrow G$ are smooth maps.

For e.g., a group of rotations in a 2D plane known as $SO(2)$,

$$SO(2) = \{R \in \mathbb{R}^{2\times 2} | R^T R = I, det(R) = 1\}. \tag{2.1}$$

$SO(2)$ is a single parameter group simply given by a 2D rotation matrix $R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$.

Definition 13. A Lie algebra $\mathfrak{g}$ of a Lie group G is the tangent space to a group defined at its identity element e with an exponential map $exp : \mathfrak{g} \rightarrow G$ and a binary operation $\mathfrak{g} * \mathfrak{g} \rightarrow \mathfrak{g}$.

The structure of Lie groups holds significant importance due to the implications of Noether's theorem. This theorem establishes a profound connection between differentiable symmetries and conservation laws. According to Noether's theorem, for every differentiable symmetry exhibited by a system, there exists a corresponding conservation law. In physics, these conservation laws are investigated by identifying the Hamiltonian of the physical system (Easton, 1993).

Our research explores two specific choices of Hamiltonians that give rise to distinct structures: the symplectic group $Sp(2d)$ and the symplectic orthogonal group $SpO(2d)$. These structures are particularly interesting as they offer valuable insights into the symmetries and conservation laws associated with the studied physical systems.

Definition 14. A symplectic group $Sp(2d)$ is a Lie group formed by the set of real symplectic matrices defined as $Sp(2d) = \{\mathbf{H} \in \mathbb{R}^{2d\times 2d} | \quad \mathbf{H}^T \mathbf{J} \mathbf{H} = \mathbf{J}\}$, where $\mathbf{J} = \begin{pmatrix} 0 & \mathbf{I_d} \\ -\mathbf{I_d} & 0 \end{pmatrix}$.

Definition 15. The Lie algebra $\mathfrak{sp}$ of a symplectic group $Sp(2d)$ is a vector space defined by, $\mathfrak{sp} = \{\mathbf{H} \in \mathbb{R}^{2d\times 2d} | \quad \mathbf{J}\mathbf{H} = (\mathbf{J}\mathbf{H})^T\}$.

Definition 16. A symplectic orthogonal group $SpO(2d)$ is defined by restricting the Hamiltonian matrices to orthogonal form.

Definition 17. A group action is a map $\circ : G \circ \mathcal{X} \rightarrow \mathcal{X}$ iff (i) $e \circ x = x, \forall x \in \mathcal{X}$, where e is the identity element of G, (ii) $(g_1 * g_2) \circ x = g_1 * (g_2 \circ x), g_1, g_2 \in G, \forall x \in \mathcal{X}$ where $*$ is a group operation.

Having established the above definition, we introduce various evaluation scores used across the projects in this thesis. The challenges of deep representation learning discussed in this thesis depend on deep learning techniques. The proposed research contributions involve utilising

deep neural networks as the foundational frameworks for representation learning, which are trained using state-of-the-art stochastic optimisation methods. To assess the generalisation and reliability of these models, we report various metrics across the thesis. In this section, we provide precise definitions for the metrics employed in our research.

2.1.3 Evaluation Metrics

Here, we introduce the technical details of evaluation metrics.

Accuracy is the ratio of the number of correctly classified instances to the total number of instances in the dataset.

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \mathbb{I}[\mathrm{y}_i = \hat{\mathrm{y}}_i]}{N}. \tag{2.2}$$

where for a sample $\mathbf{x}_i$, y_i is a target class, $\hat{y}_i$ is a predicted class and N is the number of evaluation samples.

Inter Entropy (Radford et al., 2016) measures the uncertainty associated with assigning samples to different classes. It assesses the distinguishability of samples across various classes, providing insights into the diversity of samples. Formally,

$$H(\mathbf{y}|\mathbf{x}) = -\sum_{i=1}^{K}\sum_{j=1}^{N} p(y_i|\mathbf{x}_j)\log p(\mathbf{y}_i|\mathbf{x}_j). \tag{2.3}$$

where $p(\mathbf{y}|\mathbf{x})$ is a classification model that assigns a probability vector $\mathbf{y} = (\mathrm{y}_1, \cdots, \mathrm{y}_K)$ over K classes for an input data point $\mathbf{x}_j$. A lower inter entropy indicates distinct samples across classes. This thesis uses $H(\mathbf{y}|\mathbf{x})$ to evaluate the quality of samples drawn from generative models, where we assume access to a pre-trained classifier $p(\mathbf{y}|\mathbf{x})$.

Intra Entropy (Radford et al., 2016) provides a measure of uncertainty associated with assigning generated samples within the same class. It evaluates the compactness and homogeneity of samples within each class. Mathematically,

$$H(\mathbf{y}) = -\sum_{i=1}^{K}\frac{N_i}{N}\sum_{j=1}^{N} p(\mathrm{y}_i|\mathbf{x}_j)\log(p(\mathbf{y}_i|\mathbf{x}_j)). \tag{2.4}$$

where N_i is the number of samples of class i, N is the total number of samples and K is the number of classes. A lower $H(\mathbf{y})$ means consistent samples within each class, reflecting the ability of the generative model to capture class-specific characteristics.

Inception Score (Salimans et al., 2016) is used to evaluate samples drawn from generative models. It measures the quality and diversity of generated images based on the conditional and marginal class probabilities.

$$\mathbf{IS} = \exp\left(\mathbb{E}_{\text{data}} D_{\text{KL}}(p(\mathbf{y}|\mathbf{x}) \parallel p(\mathbf{y}))\right). \tag{2.5}$$

which can be rewritten as a combination of inter-Entropy and intra-Entropy terms as $\mathbf{IS} = \exp(H(\mathbf{y}) - H(\mathbf{y}|\mathbf{x}))$. A higher value of **IS** indicates better data generation capabilities of the model.

Structural Similarity Index (SSIM) (Wang et al., 2004) is used to measure the structural similarity between two images. It considers the luminance, contrast, and structure of the images to compute a similarity score ranging from -1 to 1.

$$\text{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_\mathbf{x}\mu_\mathbf{y} + C_1)(2\sigma_\mathbf{xy} + C_2)}{(\mu_\mathbf{x}^2 + \mu_\mathbf{y}^2 + C_1)(\sigma_\mathbf{x}^2 + \sigma_\mathbf{y}^2 + C_2)}. \tag{2.6}$$

where $\mu_\mathbf{x}$ and $\mu_\mathbf{y}$ are sample means and $\sigma_\mathbf{x}^2$ and $\sigma_\mathbf{y}^2$ are sample variance of $\mathbf{x}$ and $\mathbf{y}$, and C_1, C_2 are normalising constants.

Mean Squared Error (MSE) measures the average squared difference between the noisy $\hat{\mathbf{x}}$ and a target sample $\mathbf{x}_i$ as,

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2. \tag{2.7}$$

Peak Signal-to-Noise Ratio (Wang et al., 2004) is a metric used to evaluate the quality of reconstructed or generated images by measuring the ratio between the maximum possible magnitude of a sample to the magnitude of the distortion present in a generated sample.

$$\text{PSNR}(\mathbf{x},\mathbf{y}) = 10\log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}(\mathbf{x},\mathbf{y})}\right). \tag{2.8}$$

Higher PSNR values indicate better quality and less distortion.

Homogeneity (Rosenberg and Hirschberg, 2007) is a clustering metric that measures the extent to which each cluster contains only samples from a single class. It quantifies the similarity of samples within each cluster as,

$$\text{Homogeneity}(C,K) = 1 - \frac{H(C|K)}{H(C)}. \tag{2.9}$$

where $H(C|K)$ is the conditional entropy of assigning cluster C given cluster K, and $H(C)$ is the entropy of cluster assignment C. This score is useful in assessing the purity of clusters in clustering algorithms. A higher value indicates that each cluster predominantly contains samples from a single class.

Completeness (Rosenberg and Hirschberg, 2007) measures the extent to which all samples from a single class are assigned to the same cluster. It quantifies the similarity of clusters within each class,

$$\text{Completeness}(C,K) = 1 - \frac{H(K|C)}{H(K)}. \tag{2.10}$$

where $\mathbf{H}(K)$ is the entropy of cluster assignment K. This score assesses how well each class is grouped in clustering. A higher value indicates that samples from the same class are assigned to the same cluster.

Silhouette (Rousseeuw, 1987) is used to assess the quality and separation of clusters in clustering. It is given as the average distance between a data point and its cluster compared to other clusters.

$$\text{Silhouette}(\mathbf{x}_i) = \frac{\mu(\mathbf{x}_i) - \rho(\mathbf{x}_i)}{\max(\mu(\mathbf{x}_i), \rho(\mathbf{x}_i))}. \tag{2.11}$$

where $\mu(\mathbf{x}_i)$ is the distance between a given sample $\mathbf{x}_i$ and all other data points within the same cluster, and $\rho(\mathbf{x}_i)$ is the smallest distance of $\mathbf{x}_i$ to the points in other clusters. A higher silhouette score indicates well-separated and distinct clusters, while lower scores suggest overlapping or poorly separated clusters.

With that, we conclude the formal background details required for the thesis. We now proceed to survey a range of representation learning literature. Our exploration begins with traditional approaches and subsequently looks into modern deep-learning methods. Furthermore, we explore recent advancements in geometric principles for representation learning. It is important to note that this discussion provides a general overview of the field, while more extensive coverage of related work can be found in the respective chapters.

2.2 Representation Learning: A Historical Perspective

Learning representations for high-dimensional data holds a significant interest in machine learning, owing to the prevalence of complex datasets such as images, audio, social networks, and genomics, among others (Deng et al., 2009; Panayotov et al., 2015; Leskovec and Krevl, 2014; Eraslan et al., 2019). Embedding these datasets into low-dimensional spaces that retain the data's inherent structure offers immense potential for addressing various challenges. For instance, it facilitates the development of efficient embedding-based databases for streamlined data storage and retrieval (Wang et al., 2021; Johnson et al., 2019), enables the discovery of novel patterns in genomics data with implications for personalised medicine (Eraslan et al., 2019), and much more.

Central to representation learning is the well-established *manifold hypothesis*, which posits that high-dimensional data distributions tend to be concentrated along lower-dimensional manifolds. By learning embeddings, these approaches construct models of the world that encode the factors of variations of the data distribution through latent components. This section provides an overview of diverse representation learning techniques, focusing on approaches relevant to research undertaken in this thesis. Given the breadth and depth of the literature in this field, we aim to cover the most pertinent works while referencing more extensive surveys when appropriate.

2.2.1 Linear Methods

Linear factor models are among the widely employed techniques for learning the low-dimensional probability distribution of data. The linear structure offers an interpretable perspective on the underlying factors of variation within high-dimensional data. This class of models proves particularly useful when the higher-dimensional data points lie on a low-dimensional linear subspace. Additionally, the simplicity of linear factor models makes them suitable for constructing complex mixture models. Formally, for a set of D variables in higher-dimensional data, denoted as $\mathbf{x} = \{\mathbf{x}^1, ..., \mathbf{x}^m\}$, linear factor models represent $\mathbf{x}$ in a low-dimensional latent space $\mathbf{z} = \{\mathbf{z}^1, ..., \mathbf{z}^d\}$ using d variables, according to the equation,

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \varepsilon \qquad \varepsilon \sim \mathcal{N}(\varepsilon|0, \mathbf{\Psi}), \tag{2.12}$$

where $\mathbf{W}$ is a $m \times d$ matrix known as the factor loading matrix, and ε denotes a Gaussian-distributed noise with zero mean and covariance $\mathbf{\Psi}$, capturing the variability in the data variables. A common assumption over a prior distribution over latent variable $\mathbf{z}$ is a zero mean and unit variance Gaussian distribution. This assumption enables the specification of the conditional distribution over data variables as $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z}, \mathbf{\Psi})$. The gradient descent and expectation maximisation (EM) algorithms are widely used for training these models.

The choice of a probability distribution of variables $\mathbf{x}$, $\mathbf{z}$, and ε gives rise to different models that explain data on varying levels, and their choice depends on the downstream application. Assuming the diagonal covariance matrix ($\Psi = \sigma^2\mathbf{I}$) for the noise variable results in a probabilistic PCA model. An independence assumption for components $\mathbf{z}^i$ of the latent variable $\mathbf{z}$ in a noise-free setting ($\varepsilon = 0$) describes a class of models called independent component analysis (ICA).

In a Bayesian framework, we can define the *prior* distribution $p(\mathbf{W})$ over parameters $\mathbf{W}$ as $p(\mathbf{W}) = \mathcal{N}(\mathbf{W}|0, \mathbf{I})$, which gives a conditional distribution over data variables $p(\mathbf{x}|\mathbf{z}, \mathbf{W}) = \mathcal{N}(\mathbf{x}|\mathbf{z}, \mathbf{W}\mathbf{W}^T + \Psi)$, and by integrating out the parameters we obtain $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{z}, \mathbf{Z}\mathbf{Z}^T + \mathbf{\Psi})$, where $\mathbf{Z}$ is a matrix where each row is the latent encoding of a data point $\mathbf{x}$.

2.2.2 Spectral Methods

A general recipe of spectral methods is first to utilise local distance measures to construct a data-dependent graph adjacency or kernel matrix. Subsequently, a spectral decomposition is used to obtain low-dimensional representations, as described in Algorithm 1.

Over the years, various families of methods have been developed based on the approach outlined in Algorithm 1. These methods differ in their definition of the similarity matrix. For instance, Laplacian eigenmaps proposed by Belkin and Niyogi (2003) utilise ε-neighbourhood and n-nearest neighbour graph. Multidimensional scaling (MDS) introduced by Cox and Cox

Algorithm 1: A generic framework of manifold learning using spectral methods

Input: $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, where $\mathbf{x}_i \in \mathbb{R}^D$, D is the dimensionality of data space $\mathbf{X}$ and N is number of data points.

Output: $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$, where $\mathbf{z}_i \in \mathbb{R}^k$, $k < D$ is the dimensionality of latent space $\mathbf{Z}$.

Construct a similarity matrix $\mathbf{W} = [w_{ij}]_{N \times N}$ where w_{ij} is a neighbourhood or local distance score obtained using a distance function $d(\cdot, \cdot)$;

Optionally transform $\mathbf{W}$ to a Laplacian or a normalised matrix. $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ where $\mathbf{D} = diag(\sum_j \mathbf{W}_{:j})$;

Compute the first k eigenvectors $\{\mathbf{e}^1, .., \mathbf{e}^k\}$ of $\mathbf{L}$;

Construct a matrix $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$, where the latent representation $\mathbf{z}_i$ is obtained by taking i^{th} entry of eigenvectors i.e. $\mathbf{z}_i = \{\mathbf{e}^1(i), ..., \mathbf{e}^k(i)\}$. **return** $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$

(2008) employs a double centring formula to define a distance matrix, originally used in cognitive science studies to model similarity in the human brain. However, the Euclidean distance measure in MDS does not generalise well to non-Euclidean data. Building upon MDS, Balasubramanian et al. (2002) extend the method to manifolds by replacing the local Euclidean distance in the double centring formula with a geodesic distance. Similarly,Cox and Cox (1991) focuses on hyperspherical manifolds and defines a distance measure on the surface of the sphere. Another approach, known as a local linear embedding (LLE) Roweis and Saul (2000), learns embeddings by preserving the local geometry of each data point within its neighbourhood.

However, these approaches suffer from scalability issues when confronted with high-dimensional data, primarily due to the computation of the distance or kernel matrix required in Step 1 of Algorithm 1, which has a cubic complexity in terms of the number of data points. Similarly, eigenfunction computation is again cubic in the size of data points. Some solutions have been proposed to address the challenge of high-dimensional distance or kernel matrices (Cayton and Dasgupta, 2006; Chen and Buja, 2009). Another setback is the reliance on local distance measures, which become problematic in higher-dimensional spaces where data sparsity makes it challenging to define meaningful distances (Friedman, 1997; Aggarwal et al., 2001). Recent work by Pfau et al. (2019) introduced a spectral inference network that approximates eigenfunctions of high-dimensional spaces using neural networks, enabling a promising revenue of designing efficient approaches through stochastic optimisation.

2.2.3 Non-Parameteric Methods

Linear methods are often insufficient for capturing the intricate geometric properties of high-dimensional data that do not reside in a linear space. While local linear models have been employed to address this limitation, as outlined in the previous section, they tend to struggle with sparse data with limited support on a manifold. To overcome these challenges, Gibson (1960)

and McDonald (1962) proposed a natural extension of linear methods to nonlinear models. Their approach involves incorporating a nonlinear function to transform a low-dimensional latent space, followed by a linear probabilistic model which can be mathematically represented as Equation 2.13, where $\phi : \mathbb{R}^d \to \mathbb{R}^d$.

$$\mathbf{x} = \mathbf{W}\phi(\mathbf{z}) + \varepsilon \qquad \varepsilon \sim \mathcal{N}(\varepsilon|0, \Psi). \tag{2.13}$$

Non-parametric methods offer a means of constructing the nonlinear function without explicit parameters. For instance, consider a two-dimensional latent variable $\mathbf{z} = (\mathbf{z}^{(1)}, \mathbf{z}^{(2)})$. By employing monomials of degree two, we can define $\Phi(\mathbf{z}) = (\mathbf{z}^{(1)}\mathbf{z}^{(2)}, \mathbf{z}^{(1)2}, \mathbf{z}^{(2)2})$, thereby augmenting the dimensionality of $\mathbf{z}$. This dimensionality increase can be interpreted as lifting variables to a higher-dimensional space where a linear relationship holds. However, it is important to note that using monomials involves a combinatorial explosion, as representing d-dimensional variables using monomials of degree k requires computing $\binom{d+k}{k}$ functions, which grows exponentially with d.

An alternative approach for nonlinear generalisation is kernel transformations, which map the latent space to an infinite-dimensional Hilbert space where linear relationships can be established. The kernel trick enables the direct construction of an inner product in the kernel space, avoiding the explicit computation of infinite-dimensional feature representations. Kernel-PCA, for instance, leverages this technique to compute a covariance matrix in the Hilbert space (Mika et al., 1999).

In the case of linear models, a Bayesian approach leads to a conditional distribution of the form $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{z}, \mathbf{Z}\mathbf{Z}^T + \psi)$ which can be interpreted as a Gaussian process (GP) with a linear covariance/kernel. By replacing the linear kernel $\mathbf{Z}\mathbf{Z}^T$ with an appropriate kernel function $\mathbf{K}$, such as a radial basis function (RBF), exponential, Matern, or periodic kernel, we can model the nonlinearities in the data, resulting in $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|0, \mathbf{K} + \boldsymbol{\Psi})$. For a comprehensive overview of GPs, refer to Rasmussen (2003). Lawrence (2005) proposed a GP-LVM which defines a mapping from latent variables to the data space by maximising the GP likelihood. For highly complex data, selecting an appropriate kernel or covariance function becomes crucial in capturing important factors of variation. However, in practical scenarios, such information is often unavailable a priori and relies heavily on domain-specific prior knowledge.

2.2.4 Autoencoders

Deep neural networks (DNNs) have emerged as powerful nonlinear methods for learning low-dimensional latent representations. Their effectiveness stems from their capacity to capture complex functions using backpropagation with stochastic gradient descent for training. The simplest form of a neural network for learning representations can be described by a three-layer

architecture consisting of an input layer, a hidden layer, and an output layer. The mapping from the input to the hidden layer, called the *encoder*, is a nonlinear parametric transformation $\phi_\theta : \mathbb{R}^D \to \mathbb{R}^d$. Similarly, the mapping from the hidden to the output layer, known as the *decoder*, is defined as $\phi_\omega : \mathbb{R}^d \to \mathbb{R}^D$. The network parameters are optimised by solving the least-squares objective function:

$$\min_{\theta,\omega} ||\mathbf{X} - \phi_\omega \circ \phi_\theta(\mathbf{X}))||_2^2. \tag{2.14}$$

The mappings ϕ_θ and ϕ_ω are realised as a linear map followed by a nonlinear activation function: $\phi_\theta = \mathbf{h} \circ \mathbf{W}_1$, $\phi_\omega = \mathbf{h} \circ \mathbf{W}_2$. Here, $\mathbf{W}_1$ and $\mathbf{W}_2$ represent linear maps, where the (i, j) entry of $\mathbf{W}_1$ corresponds to the strength of the connection between the i-th input and the j-th hidden neuron and the (i, j) entry of $\mathbf{W}_2$ corresponds to the strength of the connection between the i-th hidden and the j-th output neuron. The activation function $\mathbf{h}$ introduces nonlinearity to the network. The optimisation problem defined in Equation 2.14 is non-convex and is commonly tackled using gradient descent or similar iterative methods. In modern deep learning approaches, the mappings ϕ_θ and ϕ_ω are typically implemented as compositions of multiple layers, allowing for hierarchical representations and increased modelling capacity.

2.2.5 Deep Latent Variable Models

The recent success in deep latent variable models like variational autoencoders (Kingma and Welling, 2014) (VAE), normalising flows (Rezende and Mohamed, 2015) (NFs), and many other similar methods (Pu et al., 2017; Liu et al., 2017; Arjovsky et al., 2017) have raised a great interest in the modelling probability distribution of high dimensional data using deep latent variables. The key reason for this success is using cheap stochastic optimisation methods like gradient descent, which can efficiently solve the cost function and update parameters of multiple nonlinear layers using backpropagation.

We first present a summary of variational inference before introducing the variational autoencoder (VAE) framework.

2.2.5.1 Variational Inference

The objective of the latent variables model is to express the probability distribution of high-dimensional data in terms of low-dimensional latent variables. Specifically, the probability distribution of the high-dimensional data variables $\mathbf{x} = (\mathsf{x}^1, ..., \mathsf{x}^D)$ can be represented in terms of low dimensional latent variables $\mathbf{z} = (\mathsf{z}^1, ..., \mathsf{z}^d)$ as:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}. \tag{2.15}$$

where, $p(\mathbf{z})$ represents the prior over latent variables, and $p(\mathbf{x}|\mathbf{z})$ is the data likelihood term conditioned on latent variables $\mathbf{z}$. A generative model is parameterised as a distribution $p_\theta(\mathbf{x}|\mathbf{z})$, where the goal is to find optimal parameters by maximising the data log-likelihood $\log p_\theta(\mathbf{x})$ on training data. Once the model is trained, inference can be performed by computing the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ using Bayes' rule:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}}. \tag{2.16}$$

However, learning such a model is challenging; we first need to identify the appropriate distribution over latent variables, and then there is a computational intractability of evaluating the integral over latent variables $\mathbf{z}$. Many approaches resort to approximate inference methods like Markov chain Monte Carlo (MCMC) to sample from the posterior distribution. MCMC approaches have limitations in terms of speed and scalability for high-dimensional complex distributions.

An alternative solution is variational inference (VI), which offers a faster and more scalable way to perform inference. VI assumes the existence of a family of variational distributions $q(\mathbf{z})$ that can provide reasonably good approximations for various real-world problems. The goal is to find the optimal $q^*(\mathbf{z})$ that best approximates the posterior distribution $p(\mathbf{z}|\mathbf{x})$. VI achieves this by optimising the Kullback-Leibler divergence (KLD) between $q(\mathbf{z})$ and $p(\mathbf{z}|\mathbf{x})$:

$$\min_{q(\mathbf{z})\in\mathcal{Q}} KL[q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})]. \tag{2.17}$$

where $\mathcal{Q}$ represents the family of variational distributions, VI approximates the posterior distribution by solving an optimisation problem, enabling efficient inference through stochastic optimisation methods.

The KLD is non-negative, and the optimal value of the objective is zero, which occurs when $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x})$. Importantly, the KLD in the VI optimisation problem depends on the posterior $p(\mathbf{z}|\mathbf{x})$, which involves the computation of an intractable integral.

In the following steps, we will explore how the variational distribution $q(\mathbf{z})$ can address this issue and be reformulated as an alternative optimisation problem. By applying the non-negativity of the KL term, we can rewrite the expression as:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})||p(\mathbf{z})]. \tag{2.18}$$

The term on the right side of the inequality is known as the evidence lower bound (ELBO). Maximising the log-likelihood of the VI objective is equivalent to maximising the ELBO. The complexity of the variational distribution $\mathcal{Q}$ determines the solution to the ELBO. Typically, there is a tradeoff between the expressive power of $\mathcal{Q}$ and the computational cost required to obtain a solution. Therefore, it is common practice to restrict the space of $\mathcal{Q}$. One class of models, known as mean-field variational family, factorises the multivariate latent distribution

$q(\mathbf{z})$ into the product of marginals: $q(\mathbf{z}) = \prod_{i=1}^{d} q(\mathrm{z}^{(i)})$. This assumption of independence limits the model's capacity. Alternatively, more complex distributions can be defined, such as structured factorisations that capture dependencies among the latent components.

The marginal distributions $q(\mathrm{z}^{(d)})$ are typically defined using a parametric family, such as a normal distribution $q_{\theta_d}(\mathbf{z}^{(d)}) = \mathcal{N}(\mathrm{z}^{(d)}|\mu_d, \sigma_d^2)$, where $\theta_d = \{\mu_d, \sigma_d^2\}$ are the parameters of the distribution representing the mean μ_d and variance σ_d^2 of the d^{th} variable. The term $\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})]$ represents the expected value of the likelihood of the data conditioned on the latent variables. This can be modelled by a parametric model trained to maximise the data likelihood $p_\phi(\mathbf{x}|\mathbf{z})$. Variational expectation maximisation (VEM) is one of the methods used to optimise the ELBO. The E-step involves maximising the parameters of the variational distribution with respect to θ, while the M-step concerns maximising the parameters ϕ of the likelihood model. We refer to Blei et al. (2017) for a detailed overview of the topic.

2.2.5.2 Variational Autoencoder

The family of deep generative models known as Variational Autoencoders (VAEs) address the variational problem introduced in Equation 2.17 using deep learning and stochastic optimisation methods. VAEs implement the approximate posterior distribution of the latent variables, denoted as $q_\theta(\mathbf{z})$, as a parametric neural network conditioned on the data variables. This distribution, which takes the form of an encoder, is trained to model $q_\theta(\mathbf{z}|\mathbf{x})$. Similarly, the generative distribution $p_\phi(\mathbf{x}|\mathbf{z})$ is implemented as a parametric neural network interpreted as a decoder. The VAE objective, also known as ELBO, is given by:

$$ELBO(p,q) = \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]. \tag{2.19}$$

The encoder-decoder neural networks are composed of multiple layers. A main issue in optimising ELBO is the non-differentiability of the sampling step $\mathbf{z} \sim q_\theta(\mathbf{z}|\mathbf{x})$. This issue is overcome using a reparameterisation trick, which involves introducing an auxiliary random variable ε and expressing the latent variable $\mathbf{z}$ as a deterministic function of ε. Mathematically it is written as $\mathbf{z} = \mu + \sigma \odot \varepsilon$, where $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The inference distribution $q_\theta(\mathbf{z}|\mathbf{x})$ is typically modeled as a normal distribution $q_\theta(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mu_\theta, \sigma_\theta^2\mathbf{I})$, where $\mu_\theta = \{\mu^1, ..., \mu^d\}$ and $\sigma_\theta = \{\sigma^1, ..., \sigma^d\}$ are the mean and standard deviation of the d-dimensional latent variable obtained from a neural network mapping θ.

For each sample $\mathbf{x}_i$, the variational distribution is defined, and the expectation in Equation 2.19 can be approximated using a Monte Carlo estimate:

$$\mathbb{E}_{\mathbf{z}_{i,j}\sim\mathcal{N}(\mathbf{z}_i|\mu_i,\sigma_i^2\mathbf{I})}[\log p_\theta(\mathbf{x}_i|\mathbf{z}_{i,j})] = \frac{1}{J}\sum_{j=1}^{J} \log p_\theta(\mathbf{x}_i|\mathbf{z}_{i,j}) \tag{2.20}$$

$$\mathbf{z}_{i,j} = \mu_i + \sigma_i \odot \varepsilon_j \qquad \varepsilon_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2.21}$$

Under a unit normal prior $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$, the KL divergence between two multivariate normal distributions is computed as:

$$KL[\mathcal{N}(\mathbf{z}_i|\mu_i, \sigma_i^2)||\mathcal{N}(0, \mathbf{I})] = \frac{1}{2}\sum_{i=1}^{d}(1 + \log((\sigma^i)^2) - (\mu^i)^2 - (\sigma^i)^2). \tag{2.22}$$

The differentiability of the full objective allows us to optimise it using stochastic minibatch gradient descent. The parameters θ and ϕ are chosen from popular neural network architectures such as multilayer perceptrons (MLPs), convolutional neural networks (CNNs), or residual neural networks (ResNets). For more comprehensive details, we refer to Kingma and Welling (2014).

2.2.6 Geometry for Representation Learning

In this section, we explore representation learning methods that leverage geometry as a prior for learning representations or focus on learning representations of geometric objects like manifolds or graphs. We present a comprehensive overview of this field and discuss relevant literature. Further details and in-depth discussions on related work can be found in the respective chapters.

2.2.6.1 Representation Learning

The seminal work by Rifai et al. (2011b) demonstrated that low-dimensional representations learned using autoencoder networks are consistent with the manifold hypothesis and preserve the structure of the data space. In a subsequent study, Rifai et al. (2011a) explored the sensitivity of the tangent direction of a data manifold to the tangent directions of the low-dimensional latent space. They utilised this observation to describe the data manifold using a tangent bundle obtained by computing principal components of the Jacobian of the latent space with respect to the data space. The tangent bundle was then used to train classifiers insensitive to local directional changes along the manifold.

The recent integration of deep neural networks (DNNs) with generative models (GMs) has led to the emergence of deep generative models (DGMs), which offer an expressive framework for modelling complex probability distributions of high-dimensional data in lower-dimensional latent spaces. However, a drawback of this type of density estimation is that using nonlinear functions hinders the derivation of meaningful interpretations of the latent space. Consequently, several challenges arise, such as inferring the manifold structure of the latent space, navigating the latent space to find points corresponding to meaningful samples in a data space, and identifying latent directions that reveal subspaces of symmetries like rotation and reflection.

The VAE family represents a well-studied class of DGMs. Recent research efforts have focused on investigating the geometric properties of VAEs' latent spaces. Shao et al. (2017) introduced a method to unveil the Riemannian geometry of DGMs' latent spaces. They assumed

that the latent space possesses a smooth manifold structure, enabling them to leverage Riemannian geometry tools to analyse the latent space's metric properties. Their proposed algorithms encompassed (i) computing geodesic paths between pairs of latent points, facilitating smooth interpolation between data points, (ii) parallel transport of tangent vectors for computing analogies between data points, and (iii) exploring curvature properties of the latent space. Building upon this work, Chen et al. (2018a) developed a neural network approach for generating geodesic paths between data points. Their key idea involved parameterising a geodesic path using a neural network and optimising it by minimising the length or energy of the path.

In a recent study, Pfau et al. (2020) presented an algorithm for symmetry-based disentanglement of DGMs' latent spaces. Their approach generalised spectral decomposition on manifolds by identifying invariant subspaces under random walk diffusion. Concurrently, some of the works have explored the utilisation of Riemannian geometry to model the latent spaces of DGMs. Davidson et al. (2018) employed the von Mises-Fisher (vMF) distribution as a prior on latent variables, resulting in a hyperspherical latent space. This choice proved effective for representing data with rotational symmetry. Falorsi et al. (2019) and Zhou et al. (2019) focused on data exhibiting SO(3) symmetry group and leveraged the Lie group structure within the latent space of VAEs. Zhou et al. (2019) demonstrated that a smooth manifold structure is advantageous for modelling rotational symmetries encountered in kinematics problems prevalent in graphics and computer vision.

Many real-world problems involve complex topological properties. For instance, the hierarchical relationships observed in natural language, social networks, or the human genome can be better explained by employing priors that are more expressive in capturing regularities in data space. Dai et al. (2021) applied hyperbolic geometry and modelled the distribution of the latent space using the Poincaré ball, employing a wrapped normal distribution for language generation tasks. Nickel and Kiela (2018) utilised a Lorentz model and showed its relevance in capturing hyperbolic geometry in tasks such as discovering conceptual hierarchies in relational data. Khrulkov et al. (2020) demonstrated the utility of hyperbolic geometry for learning image representations, yielding a significant boost in classification performance in a few-shot learning setting. Skopek et al. (2020) proposed modelling the VAE's latent space as a product of constant curvature Riemannian manifolds, enhancing the expressiveness of VAEs and facilitating the modelling of latent spaces with arbitrary constant curvature. Pfau et al. (2020) proposed an algorithm that provides geometric insights of representation space and disentanglement of latent factors of variations using subspace diffusion.

In conclusion, real-world data exhibits complexity and encompasses various symmetries that give rise to topological and geometrical properties. It is essential to consider these properties when designing expressive representation learning approaches. As discussed, the study of these properties can be approached in two main ways. Firstly, there is a focus on identifying the

complex structure learned by latent variable models. This approach entails the development of methods that enable us to comprehend the metric properties of the latent space. By understanding the underlying geometry, we can gain insights into the organisation and relationships within the data. Secondly, an alternative approach involves imposing an expressive prior or structure in the latent space. Unlike the previous approach, this method requires prior knowledge of the domain. For example, if the data exhibits rotational symmetries, a hyperspherical latent space (Davidson et al., 2018) might be more suitable than a hyperbolic space.

Overall, by accounting for complex symmetries and incorporating appropriate inductive biases, we can improve the capacity of representations to capture the intricate characteristics of real-world data.

2.2.7 Geometric Deep Learning

The central goal of geometric deep learning (GDL) is to extend deep learning approaches to handle non-Euclidean data. The seminal work of Bronstein et al. (2017) classify geometric problems into two broad categories: i) *structure of the domain*, which assumes a manifold hypothesis and focuses on learning the underlying structure of the data, as discussed in previous sections, and ii) *structure on the domain*, which directly deals with non-Euclidean data without embedding it in a higher-dimensional Euclidean space. GDL methods belong to this category, where the emphasis is on developing techniques to analyse data in its inherent non-Euclidean form. The data is typically represented as graphs, simplicial complexes or point clouds, commonly encountered in domains such as network analysis (where social networks are represented as graphs) and computer graphics (where 3D shapes are represented as point clouds).

Bruna et al. (2014) introduced a generalisation of convolution operator for graphs, which provides a frequency-like interpretation of the eigenfunctions of a graph Laplacian. This enables the design of parametric filters in the frequency domain, facilitating the suppression of unwanted frequencies. However, the computational cost of eigendecomposition makes this approach impractical for larger graphs. To address scalability Defferrard et al. (2016) proposed ChebNet, an approximate filter based on a Chebyshev polynomial approximation of the graph Laplacian. This approximation allows for scalability when dealing with large datasets. Additionally, Kipf and Welling (2017) proposed graph convolutional networks (GCNs) that utilise a first-order approximation of the graph Laplacian to construct deep and scalable networks. This concept has been extended to a broader class of methods known as graph neural networks (GNNs), where diverse aggregation functions are defined to perform filtering operations on graph signals (Dehmamy et al., 2019). Recent advancements in constructing graph filters have incorporated ideas from heat diffusion kernels, wavelet filter banks, and other related topics of interest in the signal processing and computer graphics community (Coifman and Maggioni, 2006; Hammond et al., 2011).

The above concepts have been applied to embedding graphs in a low-dimensional latent space, offering various applications such as link prediction and node clustering. Perozzi et al. (2014) combined random walk-over graphs with language models to learn low-dimensional latent representations. Numerous variants have been proposed over the years to learn low-dimensional embeddings of graph data, and for a detailed overview, we refer to Hamilton et al. (2017b).

DGMs have also seen significant development in representation learning for graph-structured data. GraphVAEs (Kipf and Welling, 2016) proposed a variational graph autoencoder that offers a probabilistic framework for learning distribution over latent representations. Such models have also found applications in several real-world applications, including molecular property prediction (Simonovsky and Komodakis, 2018) and molecule generation Jin et al. (2018).

Graphs as a data structure encode complex hierarchical relationships between nodes or entities. A good representation space would capture this multitude of information for its applicability to a wide range of downstream tasks. Nickel and Kiela (2017) as well as Chamberlain et al. (2017), highlighted the importance of graph geometry for learning latent variable models of graph data and proposed a framework that learns graph embeddings in a hyperbolic space. The hyperbolic model provides a tree-like structure for embeddings, allowing for the modelling of complex relationships, which is vital for identifying missing links and achieving better compression. Dhingra et al. (2018) and Leimeister and Wilson (2018) also utilised a similar idea for word embeddings to capture complex syntactic and semantic properties of language. GNNs have also been extended to handle more complex graph structures, such as knowledge graphs (Schlichtkrull et al., 2018), graphs-in-graphs (Zaripova et al., 2023) and temporal graphs (Rossi et al., 2020), showcasing their applicability to a broader range of problems including brain imaging classification. In many scenarios, there is not enough labelled data for tasks on the graph domain. Hamilton et al. (2017a) proposed graph-based semi-supervised learning to take advantage of both labelled and unlabeled data, exploiting the graph's local structure to propagate information across nodes and improve the representation learning process.

Much of the application of GDL approaches revolves around data where underlying graph topology is available. However, in several tasks where the data domain is not known apriori, the above approaches are not applicable. Transformers have emerged as a potential solution in such cases where an attention (Vaswani et al., 2017) mechanism dynamically learns a fully connected graph over a set of input tokens. However, in applications such as processing point clouds, the computational cost of constructing a fully connected graph results in scalability issues. Recent developments (Wang et al., 2019; Kazi et al., 2022; de Ocáriz Borde et al., 2023) have led to an interest in approaches that can dynamically infer an inherent graph structure in a latent space where an underlying topology is not known. For a more technical overview, we refer to Chen and Wu (2022).

Bronstein et al. (2021) provides a comprehensive overview of GDL, discussing the basic concepts of geometric representations, the importance of equivariance as well as advanced topics in the field of GDL, including various recent developments and their broad range of applications.

2.3 Conclusion

We conclude this chapter by introducing the required background and covering the literature survey. In the following chapters, we will present the technical contributions of this thesis. To remind the readers, the collective contributions of our work are to demonstrate the importance of geometric principles and inductive biases for deep representation learning across different applications, including disentanglement, robustness, and generalisation capabilities for complex data domains.

Throughout the thesis, we will refer to this chapter whenever necessary to provide relevant context and foundational information.

Chapter 3

Hamiltonian Latent Operators for Content and Motion Disentanglement in Image Sequences

Developing deep generative models (DGMs) for image sequences data that can decompose the latent factors as a combination of *content* and *motion* components is of significant interest in computer vision (Cremers and Yuille, 2003; Kannan et al., 2005). The image sequence data space generally comprises objects with varying attributes that do not change under the dynamical evolution referred to as *content*. Similarly, the aspect of sequences that evolve under temporal dynamics – that is, how information is expressed in any frame of the sequence is referred to as *motion*.

The manifold hypothesis for image data states that the high-dimensional images live on a low-dimensional manifold. The temporal aspect of sequence data is a 1-dimensional submanifold of image manifold. The critical challenge in modelling image sequences is how to design a latent dynamical model that effectively captures this one-dimensional manifold created by the arrow of time (Brand, 2003; Teh and Roweis, 2003; Bird et al., 2022). Additionally, how such a model can account for the separability of a set of dynamic actions present in data space, existing deep generative models for sequences typically rely on dynamical models that accumulate errors in predicting the trajectory and are limited in their capacity to capture one-dimensional manifold of time. Moreover, they can't ensure that a training strategy cannot channel static information into the motion space. Consequently, the learned latent factors in these models lack disentanglement and long-term motion prediction often results in the generation of incoherent frames within the sequence. An essential requirement in designing a DGM is a mechanism that would introduce *temporal coherence* in motion subspace and an ability to prevent any content information from leaking into the motion subspace.

This chapter addresses the aforementioned problem by introducing a DGM framework that

leverages Hamiltonian operators (Easton, 1993) as a latent dynamic model for image sequences. Hamiltonian dynamics provides a physics viewpoint of understanding short-term and long-term temporal dependencies in a sequence, allowing us to generate more realistic and natural-looking frames. Importantly, this formulation introduces a *symplectic structure* in the latent space, wherein an energy term expresses the interdependency among latent factors. This energy can intuitively be understood as a physical property of an underlying motion and provides a meaningful notion of the nearness between the latent encodings of frames in the image sequence. That is to say, the energy level determines the accessibility of future representation given the representation of the past. As a result, even though the latent coordinates may be spatially close to each other, only the coordinates that satisfy the principle of energy conservation are considered reachable. We demonstrate such nearness is helpful for a broad space of image sequences such as human motions where it is vital to consider, “What is the space of feasible motion?" In the action of walking, what are plausible ways in which the pose of a person can vary?

Furthermore, the choice of Hamiltonian dynamics enables the model to be transferred to unseen scenarios. By preserving the individual energy, we can interpolate by predicting the missing frames in a sequence or extrapolate a whole sequence even from a frame taken from a novel viewpoint. This property allows learning a representation space that is more robust to noise and occlusions in a sequence. Moreover, by traversing the respective symplectic structure, any motion can be unrolled from an arbitrary starting frame, resulting in a natural and visually coherent transition.

Section 3.1 introduces sequential DGMs aimed at disentangling content and motion in image sequences. We highlight the limitations of existing DGMs, which fail to capture the temporal nearness and separability of motions adequately. This observation motivates the need for a novel framework that utilises a set of Hamiltonian operators to model the dynamics of various actions in the latent space. Section 3.2 discusses the related literature work. The relevant technical definitions are introduced in Section 2.1.2 of a background Chapter 2.

Moving forward, we present the formulation of our proposed framework in Section 3.3, providing detailed explanations of the generative, dynamic, and inference aspects in Sections 3.3.1, 3.3.2 and 3.3.3. We outline the empirical evaluation setup in Section 3.4 and present the results in Section 3.5. These results showcase the efficacy of our framework by demonstrating motion swapping between a pair of sequences, controlled generation, and image rotations. To further investigate the benefits of our framework, we conduct extensive ablation studies in Section 3.6. We summarise our findings and highlight the limitations of our proposed framework in Section 3.7. This comprehensive exploration of the topic provides insights into the potential of DGMs for disentangling content and motion in image sequences while shedding light on issues that require further improvement.

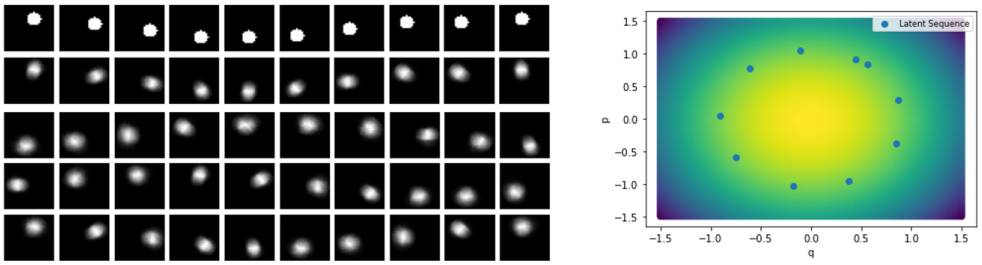


Figure 3.1: Here, we empirically demonstrate the benefits of learning the Hamiltonian operator in the latent space. Consider $(32, 32)$ image sequence of a ball rotating in a fixed orbit; the (i, j) centre of a ball moves under constraint $i^2 + j^2 = c$, where c is a constant. We generated a data set of sequences with different initial conditions and the same number of time steps. Next, we trained a VAE with a Hamiltonian operator in the latent space. We use an encoder to transform sequences to 2-dimensional phase space, then unroll a trajectory using a learnable Hamiltonian operator, and finally use a decoder to obtain the sequence. On the Left side, the top row is the original sequence, the second is the reconstructed sequence, and the last three are sequences generated from random initial states. On the right, we plot the coordinates in the phase space coloured by their energy value, along with a representation of a sequence generated from a random initial coordinate.

3.1 Introduction

The ability to learn to generate artificial image sequences has diverse uses, from animation, keyframe generation, and summarisation to restoration that has been explored in previous work over many decades (Hogg, 1983; Hurri and Hyvärinen, 2003; Cremers and Yuille, 2003; Storkey and Williams, 2003; Kannan et al., 2005). However, learning to generate arbitrary sequences is not enough; to hold a practical application, the user must be able to control aspects of the sequence generation, such as the motion being enacted or the characteristics of the agent doing an action. To enable this, we must learn to decompose image sequences into *content* and *motion* characteristics so that we can apply learnt motions to new objects or vary the motions being applied.

Deep generative models (DGMs) such as variational autoencoders (VAEs) (Kingma and Welling, 2014) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) use neural networks (NNs) to transform the samples from a prior distribution over lower-dimensional latent factors to samples from the data distribution itself. Recent developments (Chung et al., 2015; Srivastava et al., 2015; Hsu et al., 2017; Yingzhen and Mandt, 2018) extend VAEs to sequences using Recurrent Neural Networks (RNNs) on the representation of temporal frames. Similar approaches have been taken for GAN models (Tulyakov et al., 2018; Yoon et al., 2019; Dandi et al., 2020).

The dynamical processes creating the evolution of image sequences are highly constrained. Consider the simplistic case of a person walking in a scene with a camera moving around that individual. The walking poses will return to similar positions periodically, and likewise, the revolving camera will revisit previous positions. Even without strict periodicity, many dynamical processes are reversible. Any time that a dynamic could conceivably return to an earlier state suggests an implicit conservation law — the conservation of information in the underlying scene generator—as it must be capable of returning to and regenerating the same scene with non-negligible probability. The critical observation we wish to capture in this chapter is that understanding the conservation of information occurring in the context of a set of sequences is vital to decomposing *content* from *motion*.

Our objective is to capture a motion within a sequence as a conserved quantity. In physics, such motions are referred to as *Hamiltonian dynamics*; that keep the corresponding Hamiltonian function constant. Hence, we argue that a flexible latent Hamiltonian model provides a good inductive bias to learn a representation space that enables the conservation of the right quantities (which are themselves learnt) and models the dynamic evolution. This is mathematically equivalent to saying that we can learn to represent the underlying motions as combinations of differentiable symmetry groups; all differentiable symmetry transformations follow a conservation law (Noether, 1918).

We next illustrate the benefit of using Hamiltonian dynamics in the latent space of DGM in Figure 3.1.[1] Here, we observe using the Hamiltonian dynamics model can discover constant energy in latent space from a set of image sequences, proving critical for generating novel energy-preserving sequences. This example demonstrates identifying symmetries is a suitable inductive bias for developing expressive DGMs that understand the motion constraints and generalise beyond the training data. Higgins et al. (2018); Toth et al. (2020); Botev et al. (2021) have discussed the benefits of such inductive biases for learning disentangled representation.

The existing sequential DGMs do not impose any structural prior for constraining the dynamics in motion space and, therefore, accumulate errors as the sequence length grows, quickly deviating from the relevant path (Karl et al., 2017; Fraccaro et al., 2017; Yildiz et al., 2019; Bird et al., 2022). The attractive property of Hamiltonian dynamics is that they are *symplectic* that is, the divergence of a vector field is zero, and the evolving dynamics preserve the infinitesimal volume element. Consequently, the motion paths are restricted to a low-dimensional manifold in the latent space, and we can predict the dynamics forward and backward in time.

In this chapter, we intimate the more general applicability of latent Hamiltonian models; previous applications have been limited to somewhat constrained physical systems. We propose a VAE framework – Hamiltonian latent operators *HALO* to model the dynamics of image sequences using a collection of learnable linear Hamiltonian operators in the latent space.

[1] We discuss the specifics of dynamical operators in Section 3.3.2.

Specifically, for any motion sequence, we model the transition from a time step t to a step $t+1$ using a group action of a Hamiltonian operator. The evolution of the dynamics of a sequence leaves certain information unchanged, identified as *content*, and specific properties that evolve in conjunction (i.e. *motion*). Since the space of image sequences can comprise various types of dynamical actions, we split the *motion space* into subspaces where each subspace models a unique action and is unaffected by other actions. This formulation explicitly ensures the separability of dynamics. It further reduces the computational cost since the Hamiltonian of the space is now in a block diagonal form where each block is a Hamiltonian of a symmetry subgroup. Here, we focus on a discrete, identified set of actions that we can then compose at generation time. We want to remark our method can also work without action labels, as empirically demonstrated in the results. The benefit of identifying actions apriori is that we can use it for a controlled generation. We empirically demonstrate the advantages of our approach through i) the generation of diverse dynamics from a starting frame and ii) the successful disentanglement of the content and motion representation.

3.2 Related Work

3.2.1 Hamiltonian Neural Networks

Several deep learning (DL) methods have recently been proposed to learn the dynamics of physical systems using Hamiltonian mechanics. Greydanus et al. (2019) use NNs to predict Hamiltonian from phase-space coordinates $\mathbf{s} = (\mathbf{p}, \mathbf{q})$ and their derivatives. Another similar work Bondesan and Lamacraft (2019) used NNs to discover symmetries of Hamiltonian mechanical systems. More recently, Hamiltonian NNs have been used for simulating complex physical systems (Sanchez-Gonzalez et al., 2019, 2020). The key idea of this work is to represent the states of particles as a graph and use a graph neural network (GNN) to predict the change from the current state to the next state. In a follow-up work Cranmer et al. (2020), introduce sparsity on the messages in a graph and use the symbolic regression method to search for physical laws that describe the messages in the graph. Recently Toth et al. (2020) developed the Hamiltonian generative network (HGN), where they proposed to learn a Hamiltonian from image sequences. HGN maps a sequence to a latent representation and then projects it to the phase space to unroll the dynamics using a symplectic ODE integrator with Hamilton's equation. In a recent paper, Yildiz et al. (2019) use second-order ODE parameterised as a BNN for modelling dynamics of high dimensional sequence data in the latent space of VAE. Most of the developments are built on the neural ODE (Chen et al., 2018c), an idea to view layers of NNs as internal states of an ODE. These methods rely on the numerical integration scheme and the stability of the ODE solver. A Hamiltonian formalism dictates an additional requirement that the dynamics of an ODE should be volume-preserving and reversible. We want to clarify that, unlike HGN,

which mainly focuses on sequence generation and relies on symplectic ODE integrators in the latent space, we use linear Hamiltonian operators with matrix exponentials and demonstrate its relevance for disentanglement.

3.2.2 Latent Space Models

There is a long history of latent state space models for modelling sequences (Kalman, 1960; Starner and Pentland, 1997; Roweis and Ghahramani, 1999; Elliott and Krishnamurthy, 1999; Pavlovic et al., 2000). More recently, these methods have been combined with deep generative models for generating high dimensional sequences as well as learning a disentangled representation (Karl et al., 2017; Villegas et al., 2017; Tulyakov et al., 2018; Hsieh et al., 2018; Yingzhen and Mandt, 2018; Miladinović et al., 2019; Minderer et al., 2019; Franceschi et al., 2020; Zhu et al., 2020). MoCoGAN (Tulyakov et al., 2018) developed an adversarial framework, combining a random content noise with a sequence of random motion noise to generate videos. More recently, DSVAE (Yingzhen and Mandt, 2018) proposed to split a latent space into time-variant and invariant representations and use LSTM (Hochreiter and Schmidhuber, 1997) to learn the prior on time-variant representation. S3VAE (Zhu et al., 2020) improves the disentanglement of DSVAE by minimising a mutual information loss between content and motion variables. Some Hamiltonian methods (Toth et al., 2020; Yildiz et al., 2019) also model the dynamics of high dimensional sequential data in a latent space. However, the focus in those cases is only on sequence generation; to our knowledge, this has not been investigated for disentanglement.

3.2.3 Group Transformations in Latent Space Models

Rao and Ruderman (1998) proposed the algorithm to model the infinitesimal movement on data manifold using learnable Lie group operators.Culpepper and Olshausen (2009) use the matrix exponents to learn the transport operators for modelling the manifold trajectory. Many other similar methods have investigated the use of geometric operators for learning the manifold representation from data (Rao and Ruderman, 1998; Culpepper and Olshausen, 2009; Memisevic, 2012; Sohl-Dickstein et al., 2010; Cohen and Welling, 2014). The use of symmetries for learning disentangled factors of variations has recently been considered. A disentanglement is generally identified as learning representations with independent latent factors. The main goal is that each latent factor should control a distinct data factor, and a single latent variable should control no two data factors (Bengio et al., 2013b; Lake et al., 2017; Eastwood and Williams, 2018). Higgins et al. (2018) have proposed a symmetry-based definition of disentanglement. The goal in these settings was to decompose a latent space into subspaces and, in each subspace, learn a unique group transformation such that the subspace is unchanged by the action of other groups. Caselles-Dupré et al. (2019), build such a model using interaction with the environment. A few other similar approaches were recently proposed to learn group transformations in a latent

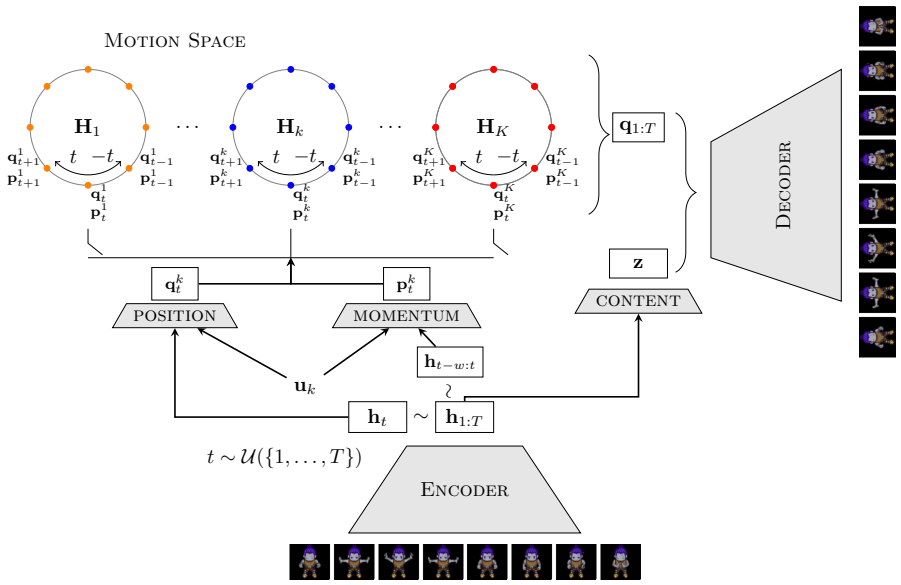


Figure 3.2: The framework for our model. We first encode each time step of a sequence to a respective feature vector $\mathbf{h}_{1:T}$. Next, to unroll the dynamics of an action k, we map the feature representation to the respective phase space. Specifically, we sample a starting index t and map $\mathbf{h}_t$ to position coordinate $\mathbf{q}_t^k$. For momentum $\mathbf{p}_t^k$, we use temporal convolution with a kernel size w on $\mathbf{h}_{t-w:t}$. We then use the operator $\mathbf{H}_k$ to trace the forward and backward trajectories. At last, we combine the position coordinates of all timesteps $\mathbf{q}_{1:T}$ with the content representation $\mathbf{z}$ and pass it through the decoder network to generate the sequence.

space (Connor and Rozell, 2020; Quessard et al., 2020; Dupont et al., 2020). However, the applications were restricted to relatively toy problems and, to our knowledge, have not been investigated on higher dimensional videos.

3.3 Method

In this section, we introduce *HALO* for sets of sequential image data. Each set of sequences depicts the temporal evolution associated with one of several *actions*. In this context, an *action* is simply a label associated with a particular sequence set. Still, where it is understood, the sequences within a set may have very different content but the same dynamic form, e.g. in the sprites data (discussed later), the actions are 'walking', 'spell cast', and 'slash' and the sequences within a set are different individuals performing the relevant action. In the following, we assume the separation into action sets is known, but that assumption is relaxed later.

Let $\mathbf{x}_{1:T}^i$ denote the ith image sequence, with $\mathbf{x}_t^i$ the tth frame in the sequence. Let $\mathbf{u}^i$ be an indicator vector denoting the action associated with the ith sequence; i.e. $u_k^i = 1$ iff sequence i follows action k and $u_{k'}^i = 0$ for all other $k' \neq k$. These sequences and corresponding actions are collected into a dataset $\{(\mathbf{x}_{1:T}^i, \mathbf{u}^i)\}_{i=1}^N$ of size N, where, for the sake of simplicity in description,

we assume they all are of same length T. In this work, we use a latent space to aid the modelling of each sequence and decompose that latent space into two parts, which we call a *content* space (denoted by $\mathbf{Z}$) and a *motion* space (denoted by $\mathbf{S}$). As the data comprises sequences of various actions that take different dynamical forms, we further decompose the latent motion space $\mathbf{S} = \mathbf{S}^1 \oplus \mathbf{S}^2 \oplus \ldots \oplus \mathbf{S}^K$, with one subspace for each action. In modelling a sequence corresponding to action k, only the subspace $\mathbf{S}^k$ will be allowed to change across the length of that sequence. Each motion subspace is further decomposed into generalised *position* $\mathbf{Q}^k$ and *momentum* $\mathbf{P}^k$ parts: $\mathbf{S}^k = (\mathbf{Q}^k, \mathbf{P}^k)$. For a frame $\mathbf{x}_t$ of an action k, the coordinates in motion space $\mathbf{s}_t^k \in \mathbf{S}^k$ are expressed as a combination of the position coordinates $\mathbf{q}^k \in \mathbf{Q}^k$ coordinate and momentum coordinates $\mathbf{p}^k \in \mathbf{P}^k$. Only the position component of this phase space is used together with content to create individual images in a sequence generatively. The momentum part *only* affects the dynamics.

The above formulation provides many advantages; it prevents the neural network from leaking constant *content* information via the motion representation. It also ensures the possibility of preserving key conservation quantities that we argue are implicit in the constraints of most motion dynamics. This is discussed further in Paragraph 3.3.2. The entire framework of our model is illustrated in Figure 3.2. Next, we introduce the generative model, followed by variational formalism for inference and learning.

3.3.1 Generative model

For completeness, we first present the full probabilistic model in 3.1-3.4 before describing each component. Each dynamic is categorised by a particular *action* enumerated by k, encoded in an indicator vector $\mathbf{u}$ (i.e. $u_k = 1$ for action k). The generative model is conditioned on this action vector. First, in (3.1), we sample the *content* variable $\mathbf{z}$ from a prior $p(\mathbf{z})$. The content variable will describe the characteristics of constant appearance expressed throughout the sequence. Next, we sample a starting position from a prior $p(\mathbf{q}_1^k)$ and momentum from a prior $p(\mathbf{p}_1^k)$ (we initialise the actions not represented in the sequence to zero). The full state-space representation for the dynamic of action k is then given by $\mathbf{s}_1^k = (\mathbf{q}_1^k, \mathbf{p}_1^k)$. The dynamical model Equations 3.3-3.7 then traces out the forward trajectory in the phase space. Finally, we combine the position trajectory with the content representation and use a decoder neural network to get the emission distribution of the data space sequence. In summary,

GIVEN: k denoting action label for a sequence,

$$\mathbf{z} \sim p(\mathbf{z}), \quad \mathbf{q}_1^k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_\mathsf{d}), \quad \mathbf{p}_1^k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_\mathsf{d}), \tag{3.1}$$

$$\mathbf{s}_1^k = (\mathbf{q}_1^k, \mathbf{p}_1^k), \quad \mathbf{s}_1^{k'} = \mathbf{0}, \quad \forall k' \neq k, \tag{3.2}$$

$$\mathbf{s}_t^k = f(\mathbf{s}_{t-1}^k; \omega_k, t), \quad \mathbf{s}_t^{k'} = \mathbf{s}_{t-1}^{k'}, \quad \forall t > 1, k' \neq k, \tag{3.3}$$

$$\mathbf{q}_t = [\mathbf{q}_t^1, \ldots, \mathbf{q}_t^K], \mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_t | \phi(\mathbf{z}, \mathbf{q}_t), \alpha^2 \mathbf{I}_\mathsf{m}), \ \forall t. \tag{3.4}$$

where d is the dimensionality of k^{th} subspace, m is the dimensionality of data space, f is a dynamical model 3.7 and ω_k are the parameters of f to be used for the k^{th} subspace. We use an emission distribution that is a spherical Gaussian, with a parameterised mean $\phi(\cdot,\cdot)$, and a covariance $\alpha^2\mathbf{I_m}$.

3.3.2 Dynamical Model

In image sequences, we can view each frame of a sequence as a point in an abstract representation space; the temporal dynamics trace a path connecting the frames, forming a 1-submanifold of the image manifold. Most dynamical models either try to capture this geometry deterministically (Srivastava et al., 2015) or probabilistically (Chung et al., 2015; Hsu et al., 2017; Yingzhen and Mandt, 2018) via linear or nonlinear state-space models. In either case, small errors in dynamical steps can accumulate and result in a significant deviation from the manifold when unrolling long-term trajectories during inference (Karl et al., 2017; Fraccaro et al., 2017). Interestingly, Hamiltonian systems alleviate these issues by constraining the dynamics to be symplectic and reversible. The symplectic geometry ensures the dynamics are volume-preserving, preventing any deviation from the manifold, and reversibility helps understand how the state of an object changes under dynamical evolution. By reversing the arrow of time, the object could return to its previous state; this awareness provides a sense of accountability to an object for its actions. In our work, without significant loss of generality, we propose a linear Hamiltonian system in the latent layer, relying on deep neural network mapping to data space to handle all nonlinear aspects. The linearity of dynamics also enhances their interpretability.

Definition 18. A matrix $\mathbf{H} \in \mathbb{R}^{2d\times 2d}$ is an Hamiltonian matrix if $\mathbf{H}^T\mathbf{J}\mathbf{H} = \mathbf{J}$, where $\mathbf{J}$ is a skew-symmetric matrix $\mathbf{J} = \begin{pmatrix} 0 & \mathbf{I_d} \\ -\mathbf{I_d} & 0 \end{pmatrix}$ and $\mathbf{I_d}$ is an identity matrix.

Consider a coordinate vector $\mathbf{s} \in \mathbb{R}^{2d}$ in the phase space $\mathbf{S}$ at a time t that evolves under constant Hamiltonian energy $\mathbf{E}$,

$$\mathbf{E} = \frac{1}{2}\mathbf{s}^T\mathbf{M}(t)\mathbf{s} \tag{3.5}$$

where $\mathbf{M}(t)$ is a symmetric matrix. In Hamiltonian mechanics, the coordinates are specified in terms of position $\mathbf{q}$ and momentum $\mathbf{p}$ variables as $\mathbf{s} = (\mathbf{q}, \mathbf{p})$. Using the fact energy $\mathbf{E}$ is constant over time, we can express the equation of motion as,

$$\frac{d\mathbf{s}(t)}{dt} = \mathbf{H}(t)\mathbf{s} \tag{3.6}$$

where $\mathbf{H}(t) = \mathbf{J}\mathbf{M}(t)$. The closed-form solution of the above system is given by matrix exponential $\mathbf{s}(t) = e^{t\mathbf{H}}\mathbf{s}(0)$. We use fast Taylor approximation (Bader et al., 2019) to compute matrix exponential that provides a stable solution under matrix norms. The real Hamiltonian matrices form a symplectic Lie group under multiplication $Sp(2d)$ with $2d^2 + d$ independent

elements. For small t, we can interpret matrix exponent $e^{t\mathbf{H}}$ as an infinitesimal transformation of state $\mathbf{s}(0)$ under the action of a Lie group $Sp(2d)$ formed by $\mathbf{H}$ matrix. We also consider the symplectic orthogonal group $SpO(2d)$ that further restricts the Hamiltonian matrix to a skew-symmetric form with $(d^2-d)/2$ independent elements. The benefit of this restriction is that the Hamiltonian matrix reduces to a rotation operator that is easy to interpret. The formal definitions are introduced in background Chapter 2.1.2. We refer readers to Easton (1993) for a more comprehensive overview of the Lie group structure of Hamiltonian matrices.

In this work, we consider K Hamiltonians $\mathbf{H}_1,\ldots,\mathbf{H}_K$, each acting on a unique subspace of the phase space $\mathbf{S}^1,\mathbf{S}^2,\ldots,\mathbf{S}^K$. To unroll the trajectory of motion k, we use the group action defined by the matrix exponent of the operator $\mathbf{H}_k$ on a starting phase space representation $\mathbf{s}_1^k \in \mathbf{S}^k$ given by,

$$\mathbf{s}_t^k = f(\mathbf{s}_{t-1}^k;\omega_k,t) = e^{t\mathbf{H}_k}\mathbf{s}_{t-1}^k, \forall t>1; \quad \mathbf{s}_t^{k'} = \mathbf{0}, \quad \forall t,k' \neq k. \tag{3.7}$$

The backward dynamics can be obtained by negating time, i.e., replacing t with $-t$ above. We assume all time steps are equally spaced. The above formulation provides an explicit disentanglement of the motion space. It further allows us to parallelise the computation of the matrix exponential by leveraging the block diagonal form of $\mathbf{H}$. Specific to our work, we parameterise a symmetric matrix $\mathbf{M}_k$ and obtain its Hamiltonian matrix as $\mathbf{H}_k = \mathbf{J}\mathbf{M}_k$ where $\mathbf{J}$ is a fixed skew-symmetric matrix as stated in definition 18.The top part of Figure 3.2 demonstrates the trajectory unrolling using Hamiltonian dynamics.

3.3.3 Inference

In order to learn the model parameters, we need to infer the distribution over latent variables. We use variational inference to learn the model parameters that lead to maximising the evidence lower bound (ELBO) objective,

$$\max_q \mathbb{E}_{q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})} \log\left[\frac{p(\mathbf{x}_{1:T},\mathbf{z},\mathbf{s}_{1:T}|\mathbf{u})}{q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})}\right]. \tag{3.8}$$

where $q(.|.)$ is the approximate posterior distribution and $\mathbf{s}_t = (\mathbf{q}_t,\mathbf{p}_t)$. It remains to define the approximate posterior we use. Since the Hamiltonian dynamics are reversible, at inference time, we randomly sample a choice of frame t and use forward and backward action of Hamiltonian to trace the trajectory of states after and before that frame for the respective action as stated in Equation 3.7.

For a sequence $\mathbf{x}_{1:T}$, we use the process in Equation 3.9-3.13 to draw samples from a variational distribution $q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})$. Simply, we sample the content variable $\mathbf{z}$ conditioned on the observed data and independently sample the motion states $\mathbf{s}_t^k = (\mathbf{q}_t^k,\mathbf{p}_t^k)$ for the reference frame t conditioned on the observed data and the relevant action k. Motion states corresponding

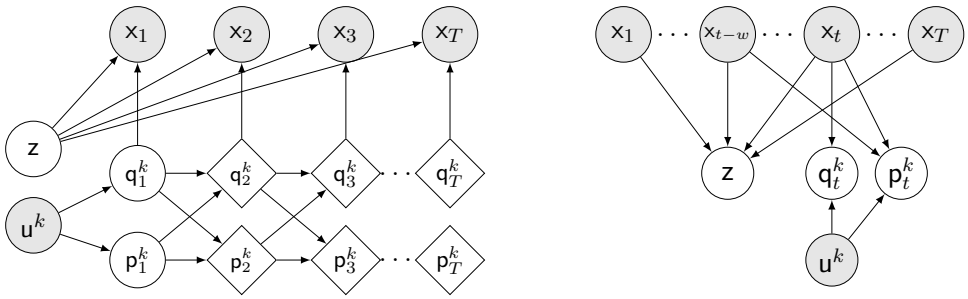


Figure 3.3: On the right is the probabilistic graph of our generative model, and on the left of the inference model. In a generative model we first sample a content variable $\mathbf{z}$ and action variable u^k. Next, the initial position and momentum coordinates are sampled from a distribution conditioned on the action variable, which a dynamical model uses to unroll a deterministic trajectory of motion coordinates. Finally, the position coordinates are combined with content variables to generate respective frames. In an inference model, the position and momentum coordinates of action u^k at time step t are determined using an inference network. Additionally, a content variable is sampled from another inference network.

to other actions are set to zero. In equations, this is,

$$\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}_{1:T}), \tag{3.9}$$

$$t \sim \mathcal{U}(\{1,\ldots,T\}), \tag{3.10}$$

$$\mathbf{q}_t^k \sim q(\mathbf{q}_t^k|\mathbf{x}_t,\mathbf{u}), \tag{3.11}$$

$$\mathbf{p}_t^k \sim q(\mathbf{p}_t^k|\mathbf{x}_{t-w:t},\mathbf{u}), \tag{3.12}$$

$$\mathbf{s}_t^k = (\mathbf{q}_t^k,\mathbf{p}_t^k). \tag{3.13}$$

where t is a starting index, $q(\mathbf{q}_t^k|\mathbf{x}_t,\mathbf{u})$, is the posterior distributions of k^{th} position subspace conditioned on the frame $\mathbf{x}_t$ and action variable $\mathbf{u}$, $q(\mathbf{p}_t^k|\mathbf{x}_{t-w:t},\mathbf{u})$ is the posterior distributions of k^{th} momentum subspace conditioned on w previous frames and action variable $\mathbf{u}$ and $q(\mathbf{z}|\mathbf{x}_{1:T})$ is the posterior distribution of the content space conditioned on the entire sequence. We parameterise the factorised posterior as a spherical Gaussian distribution learned using an encoder neural network. Specifically, $q(\mathbf{z}|\mathbf{x}_{1:T})$ as a content network, $q(\mathbf{q}_t^k|\mathbf{x}_t,\mathbf{u})$ as a position network, and $q(\mathbf{p}_t^k|\mathbf{x}_{t-w:t},\mathbf{u})$ as a momentum network. We use the reparametrisation trick (Kingma and Welling, 2014) to sample from the latent distribution.

3.3.4 Learning Objective

We use maximum loglikelihood on sequence variables to derive the evidence lower bound (ELBO),

$$\begin{aligned}\log p(\mathbf{x}_{1:T}|\mathbf{u}) &= \log \int p(\mathbf{x}_{1:T},\mathbf{z},\mathbf{s}_{1:T}|\mathbf{u})d\mathbf{s}_{1:T}d\mathbf{z},\\ &= \log \int \frac{p(\mathbf{x}_{1:T},\mathbf{z},\mathbf{s}_{1:T}|\mathbf{u})}{q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})} q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})d\mathbf{s}_{1:T}d\mathbf{z},\\ &\geq \int \log\left[\frac{p(\mathbf{x}_{1:T},\mathbf{z},\mathbf{s}_{1:T}|\mathbf{u})}{q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})}\right] q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})d\mathbf{s}_{1:T}d\mathbf{z},\\ &\geq \mathbb{E}_{q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})}\log\left[\frac{p(\mathbf{x}_{1:T},\mathbf{z},\mathbf{s}_{1:T}|\mathbf{u})}{q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})}\right], \end{aligned} \tag{3.14}$$

where $\mathbf{s}_t = [\mathbf{q}_t, \mathbf{p}_t]$. The joint distribution is factorised as,

$$p(\mathbf{x}_{1:T},\mathbf{z},\mathbf{s}_{1:T}|\mathbf{u}) = p(\mathbf{z})p(\mathbf{x}_1|\mathbf{q}_1,\mathbf{z})\prod_{t=1}^{T-1} p(\mathbf{x}_{t+1}|\mathbf{q}_{t+1},\mathbf{z})p(\mathbf{q}_{t+1},\mathbf{p}_{t+1}|\mathbf{q}_t,\mathbf{p}_t,\mathbf{u}). \tag{3.15}$$

Since we transform the starting latent state $\mathbf{s}_1 = (\mathbf{q}_1, \mathbf{p}_1)$ using a deterministic transformation $f(t,\mathbf{H};\omega) = e^{t\mathbf{H}}$ (where ω are the parameters of $\mathbf{H}$ matrix), we can write our transition distribution as,

$$p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{u}) = p(\mathbf{s}_t|\mathbf{s}_{t-1},\mathbf{u})\left|\frac{df}{d\mathbf{s}_t}\right| = p(\mathbf{s}_t|\mathbf{s}_{t-1},\mathbf{u})e^{\mathrm{Tr}(\mathbf{H})} = p(\mathbf{s}_1|\mathbf{u})\prod^{t} e^{\mathrm{Tr}(\mathbf{H})}. \tag{3.16}$$

where Tr is the trace operator, $|.|$ denotes determinant, and $p(\mathbf{s}_1|\mathbf{u}) = p(\mathbf{q}_1|\mathbf{u})p(\mathbf{p}_1|\mathbf{u})$. The transition model is reversible; therefore, without loss of generality, we can replace a starting step 1 with any arbitrary t and unroll trajectory forward and backward. We next equate (3.16) in the generative model defined in (3.15) that reduces the factorisation to,

$$p(\mathbf{x}_{1:T},\mathbf{z},\mathbf{s}_{1:T}|\mathbf{u}) = p(\mathbf{z})p(\mathbf{x}_1|\mathbf{q}_1,\mathbf{z})p(\mathbf{q}_t|\mathbf{u})p(\mathbf{p}_t|\mathbf{u})\prod_{t\prime=1,\neq t}^{T-1} p(\mathbf{x}_{t\prime}|\mathbf{q}_{t\prime})e^{\mathrm{Tr}(\mathbf{H})}. \tag{3.17}$$

We factorise the variational distribution $q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u})$ as,

$$q(\mathbf{z},\mathbf{s}_t|\mathbf{x}_{1:T},\mathbf{u}) = q(\mathbf{z}|\mathbf{x}_{1:T})q(\mathbf{q}_t|\mathbf{x}_t,\mathbf{u})q(\mathbf{p}_t|\mathbf{x}_{t-w:t},\mathbf{u}), \quad \mathbf{s}_t = [\mathbf{q}_t,\mathbf{p}_t]. \tag{3.18}$$

We now use the equations (3.18) and (3.17) to rewrite the ELBO as,

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_{1:T}),q(\mathbf{q}_t|\mathbf{x}_t,\mathbf{u}),q(\mathbf{p}_t|\mathbf{x}_{t-w:t},\mathbf{u})}\log\left[\frac{p(\mathbf{z})p(\mathbf{q}_t|\mathbf{u})p(\mathbf{p}_t|\mathbf{u})p(\mathbf{x}_1|\mathbf{q}_1,\mathbf{z})\prod_{t\prime=1,\neq t}^{T} p(\mathbf{x}_{t\prime}|\mathbf{q}_{t\prime},\mathbf{z})e^{Tr(\mathbf{H})}}{q(\mathbf{z}|\mathbf{x}_{1:T})q(\mathbf{q}_t|\mathbf{x}_t,\mathbf{u})q(\mathbf{p}_t|\mathbf{x}_{t-w:t},\mathbf{u})}\right] \tag{3.19}$$

$$\begin{aligned}&\mathbb{E}_{q(\mathbf{q}_t|\mathbf{x}_t,\mathbf{u})}\log\left[\frac{p(\mathbf{q}_t|\mathbf{u})}{q(\mathbf{q}_t|\mathbf{x}_t,\mathbf{u})}\right] + \mathbb{E}_{q(\mathbf{p}_t|\mathbf{x}_{t-w:t},\mathbf{u})}\log\left[\frac{p(\mathbf{p}_t|\mathbf{u})}{q(\mathbf{p}_t|\mathbf{x}_{t-w:t},\mathbf{u})}\right] + \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_{1:T})}\log\left[\frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x}_{1:T})}\right]\\ &+ \mathbb{E}_{q(\mathbf{q}_t|\mathbf{x}_t,\mathbf{u})}\left[\sum_{t\prime}\log p(\mathbf{x}_{t\prime}|\mathbf{q}_{t\prime},\mathbf{z})\right].\end{aligned} \tag{3.20}$$

The trace of the real Hamiltonian matrix is zero; we can, therefore, omit the term $Tr(\mathbf{H})$. Since, for each motion $\mathbf{u}_k$, we associate a separate Hamiltonian $\mathbf{H}_k$ that acts on a subspace $\mathbf{S}^k$, we can view the full state space $\mathbf{S}$ as a partition of symmetry groups $\mathbf{S} = \mathbf{S}_1 \oplus \cdots \oplus \mathbf{S}_K$ where

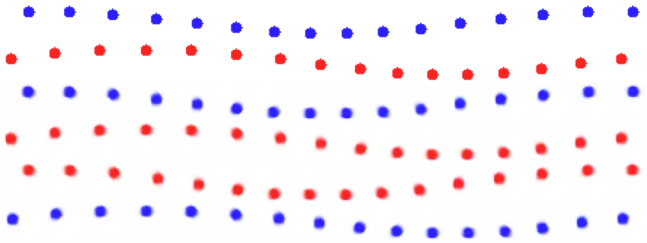


Figure 3.4: First two rows are original sequences, the next two rows are respective reconstructions, and the last two rows are generated by swapping the content variables. Swapping the content changes the colour, but the dynamics are intact.

the Hamiltonian $\mathbf{H}$ is in the block diagonal form $\mathbf{H} = diag(\mathbf{H}_1, \cdots, \mathbf{H}_K)$. We, therefore, express the distributions in terms of the variables of their respective subspaces to obtain the final ELBO,

$$-KL[q(\mathbf{q}_t^k|\mathbf{x}_t, \mathbf{u})||p(\mathbf{q}_t^k)] - KL[q(\mathbf{p}_t^k|\mathbf{x}_{t-w:t}, \mathbf{u})||p(\mathbf{p}_t^k)] - KL[q(\mathbf{z}|\mathbf{x}_{1:T}, \mathbf{u})||p(\mathbf{z})], \\ + \mathbb{E}_{q(\mathbf{q}_t^k|\mathbf{x}_t, \mathbf{u})}\left[\sum_{t'} \log p(\mathbf{x}_{t'}|\mathbf{q}_{t'}, \mathbf{z})\right]. \tag{3.21}$$

Figure 3.3 describes the probabilistic graph of the generative model on the left and an inference model on the right.

3.4 Experimental Setup

This section presents the wide-ranging applicability of *HALO* to disentangle content and motion in different datasets. Firstly, we showcase the effectiveness of *HALO* by applying it to a synthetic dataset of image sequences depicting rotating balls that evolve under constant energy. This synthetic dataset allows us to demonstrate the disentanglement capabilities of *HALO* in a controlled setting.

Moving forward, we extend the evaluation of *HALO* to two complex datasets: Sprites and MUG (Aifanti et al., 2010). These datasets present more challenging scenarios where the disentanglement of content and motion becomes more intricate. Furthermore, we explore the application of *HALO* in predicting rotations on the MNIST digits dataset. Here, the Hamiltonian structure employed by *HALO* proves advantageous in generating long-term sequences with varying velocities.

3.4.1 Datasets

3.4.2 Synthetic Data

We construct a set of sequences of images of a ball that moves in an orbit under constraint $i^2 + j^2 = c$, where (i, j) is the centre of a ball and c is the distance from the centre of an orbit.

Each sequence is drawn from a different initial condition decided uniformly at random, and all sequences are of length 16. To introduce the content element, we colour half of the sequence "red" and the remaining "blue". In Figure 3.4, we show the result of swapping the content variable of two held-out sequences that demonstrates the effectiveness of *HALO* in disentangling *motion* from the *content* while keeping the dynamics intact.

Next, we introduce details of the two datasets, followed by a discussion of the results in Section 3.5.

3.4.2.1 Sprites

Sprites consist of sequences of animated characters performing different actions ('walking', 'spell cast' and 'slashing' from three viewing angles: 'left', 'right' and 'straight') as per sprites sheets.[2] The sequences are of length 8 RGB images of size $64 \times 64 \times 3$. Each character's appearance has four attributes: skin colour, hairstyle, tops and pants. Each attribute can take six values, resulting in 1296 unique characters. We used 1000 characters for training and the rest for evaluation. The choice of 1000 is based on the existing baseline methods and is adopted for fair comparison.

3.4.2.2 MUG face expression

MUG (Aifanti et al., 2010) is a dataset of six facial expressions (anger, disgust, fear, happiness, sadness and surprise) of 52 individuals. Sequences are of variable lengths ranging from 50 to 160 frames. We downsample the sequences by a factor of two and then take a random subsequence of length 8, crop the face region and resize it to 64×64. The 75% of sequences are used for training and 25% for evaluation following Tulyakov et al. (2018).

3.4.2.3 Implementation details

We use Pytorch (Paszke et al., 2019) for the implementation of neural networks. We train all our models on Nvidia GeForce RTX 2080 GPUs. Our code is available on GitHub. [3]. The architecture of the encoder and decoder network is based on (Yingzhen and Mandt, 2018), also outlined in Table 3.8 and 3.9. We use the same network architecture for both sprites and the MUG dataset. The output of an encoder is fed to the content, position, and momentum network to get the variational distributions in $\mathbf{Z}$, $\mathbf{Q}$ and $\mathbf{P}$ space. Table 3.10 describes the architecture of the network. For the position and momentum network, the input action k is represented by a one-hot vector $\mathbf{u}$ that takes one at index k and is zero elsewhere. For MUG, we choose $|\mathbf{Z}| = 512$, $|\mathbf{Q}| = K \times 12$ and $|\mathbf{P}| = K \times 12$ and for sprites $|\mathbf{Z}| = 256$, $|\mathbf{Q}| = K \times 6$ and $|\mathbf{P}| = K \times 6$, where K is the number of actions. For sprites, $K = 3$ and for MUG, $K = 6$. To train all our models, we

[2] https://github.com/jrconway3/ Universal-LPC-spritesheet
[3] https://github.com/MdAsifKhan/HALO.git

Model	Dataset	SSIM↑	PSNR↑	MSE↓
	Sprites	$\mathbf{0.982 \pm 0.005}$	$\mathbf{36.76 \pm 1.096}$	$\mathbf{0.0005 \pm 0.0002}$
H	**MUG**	$\mathbf{0.797 \pm 0.003}$	$\mathbf{24.49 \pm 0.099}$	$\mathbf{0.0040 \pm 0.0001}$
	Sprites	0.950 ± 0.021	33.88 ± 2.03	0.0026 ± 0.0012
Skew-**H**	MUG	0.791 ± 0.003	24.25 ± 0.094	0.0044 ± 0.0001

Table 3.1: We can see both choices of an operator can generate sequences close to the ground truth.

use an Adam (Kingma and Ba, 2015) optimiser with a learning rate of $2e^{-4}$ and a batch size of 24.

3.5 Results and Discussion

We first compare two choices of Hamiltonian structure, a symplectic group using **H** and the symplectic orthogonal group by restricting **H** to a skew-symmetric form that we refer to as skew-**H**. Here, we map a starting frame to the latent space, unroll the trajectory and then map the timesteps to a data space using a decoder. We generate sequences of length 16 (twice the length used for training). The sprites consist of periodic sequences of length 8 where the start and end frames are identical; in this case, we duplicate the sequence to get a ground truth of length 16. For MUG, we draw a sequence of length 16 from the evaluation set. We compare the generated sequences with target sequences using per-frame structural similarity index measure (SSIM), peak signal-to-noise ratio (PSNR) and mean squared error (MSE). The SSIM scores are between -1 and 1, with a more significant score indicating more similarity between the ground truth and generated sequence. Likewise, higher PSNR and lower MSE imply better generation. Table 3.1 describes the performance under different scores, demonstrating our model can generate high-quality sequences from an input image. We observe that **H** performs better than skew-**H**. We hypothesise that the superior performance of **H** can be attributed to the fact that skew-**H** introduces additional restrictions on the parameters of **H** that might reduce the expressiveness of the model. This result is an interesting finding – the question we leave as a future scope of the work. We consider the dynamical operator **H** for the rest of the experiments.

To evaluate disentanglement we compare with the state-of-the-art baselines DSVAE (Yingzhen and Mandt, 2018), MoCoGAN(Tulyakov et al., 2018) and S3VAE (Zhu et al., 2020).

Method	Data	Accuracy↑	$H(\mathbf{y}\|\mathbf{x})\downarrow$	$H(\mathbf{y})\uparrow$	IS ↑
HALO (conditional)		**0.929**	**0.108**	**1.778**	**5.312**
HALO (unconditional)		0.750	0.187	1.762	4.830
DSVAE (Yingzhen and Mandt, 2018)	MUG	0.543	0.374	1.657	3.607
MoCoGAN (Tulyakov et al., 2018)		0.631	0.183	1.721	4.655
S3VAE (Zhu et al., 2020)		0.705	0.135	1.760	5.078
HALO (conditional)		**1.000**	**0.011**	2.009	7.374
DSVAE (Yingzhen and Mandt, 2018)	Sprites	0.907	0.072	2.192	8.331
MoCoGAN (Tulyakov et al., 2018)		0.928	0.090	2.192	8.182
S3VAE (Zhu et al., 2020)		**0.994**	0.041	**2.197**	**8.636**

Table 3.2: Here, we report the disentanglement of content and motion components of latent space. The high accuracy and Inter-Entropy $H(\mathbf{y})$ while low Intra-Entropy $H(\mathbf{y}|\mathbf{x})$ are expected from a better model. Our model performs best across all three scores on MUG. On sprites, we are comparable to S3VAE. This is due to the simplicity of classes in sprites. We want to remark that our unconditional model, as demonstrated on MUG, significantly outperforms other baselines, showing the benefit of Hamiltonian even when labels are unavailable.

3.5.1 Quantitative Evaluation

We use a pre-trained action prediction classifier for evaluating disentanglement. The architecture is provided in Table 3.14. To begin with, we draw a starting position and momentum from a prior distribution and use a dynamical model to unroll the trajectory in the phase space. Next, we sample the content variable $\mathbf{z}$ from real sequences and combine it with position variables to generate image sequences. We report the performance of the classifier in predicting the action from these generated sequences. The score helps measure the model's tendency to keep the motion intact with the modified content. We use the same classifier to report the intra-Entropy $H(\mathbf{y}|\mathbf{x})$ and inter-Entropy $H(\mathbf{y})$ that estimates the diversity of generated sequences. $H(\mathbf{y}|\mathbf{x})$ measures the closeness of generated sequences to the real sequences, and $H(\mathbf{y})$ measures the diversity of generated sequences (He et al., 2018). The two scores can be combined to obtain inception score (IS) (Salimans et al., 2016), a commonly used evaluation criterion for the diversity of generated samples($\mathbf{IS} = \exp(H(\mathbf{y}) - H(\mathbf{y}|\mathbf{x}))$ Barratt and Sharma (2018)). The use of IS in this work is mainly based on its prevalence in baseline approaches. There are several caveats, such as its sensitivity to weights of a classification model used to report the performance (Barratt and Sharma, 2018). We therefore recommend that readers not put emphasis

Sprites (Attr.)	Accuracy↑
Skin Color	0.925
Shirt	0.948
Pant	0.968
Hair	0.992
Identity (MUG)	0.998

Table 3.3: Here, we investigate the extent to which content is preserved when we switch motion variables with an arbitrary sequence. We report the accuracy of individual attributes in sprites and the identity of actors in the MUG dataset. The results show the content space can capture attributes that don't change under dynamics.

on the score. There are other possible choices of scores for evaluation, such as Fréchet inception distance (FID) Martin and Elster (2020) that use a pre-trained network to project images to a feature space and use a distance between the distribution of real and generated images. Our work focuses on disentanglement, commonly evaluated using scores chosen in this work. For the description of the evaluation scores, we refer to the background Chapter 2.1.3.

The results are reported in Table 3.2. We observe *HALO* outperforms the baselines on the MUG and is comparable with S3VAE on sprites. This improvement results from explicitly associating an action with a unique subspace that allows separability of the dynamics and avoids any mixing or ambiguity of action in the motion space. The results on sprites are comparable; we attribute this to the simplicity of sprites' classes that result in high performance across all models. We want to remark that our formulation is not constrained by action variables $\mathbf{u}$. Table 3.2 also describes the results for an unconditional version on the MUG (Aifanti et al., 2010) that significantly outperforms the baseline. The details of the unconditional model are discussed in the ablation 3.6. The benefit of incorporating action variables $\mathbf{u}$ is that it allows controlled generation of sequences, as demonstrated in Figure 3.7.

Next, we evaluate the tendency to preserve the identity of sequences. For sprites, identity refers to four different attributes, and for MUG, it is the sequence label of the individual. We pre-train a classifier on the identity prediction task and use it to evaluate the generated sequences. This way, we measure the model's ability to keep the identity intact when changing the motion. For sprites, we report the accuracy of individual attributes. Table 3.1 outlines the results. We can see on MUG that our model can accurately preserve the identity. We can make a similar observation for different attributes of sprite sequences. Thus, good performance indicates that the content is preserved when traversing the motion subspace, and the motion space is invariant when changing the content variables, also validated by the qualitative results.

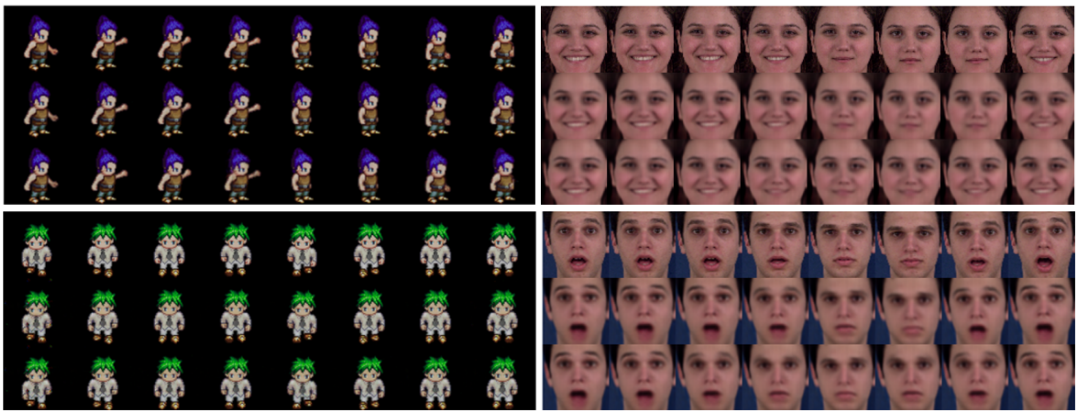


Figure 3.5: In each patch, the first row is the original sequence, the second row is reconstruction, and the third row is a sequence generated by an action of the operator on the starting time step. The reconstruction demonstrates that our model can learn good representations, and the generation demonstrates the dynamical operator can capture realistic motions from an arbitrary starting frame. More examples are in Figure 3.17.

3.5.2 Qualitative Evaluation

We first evaluate the quality of sequence prediction by comparing the original sequence, its reconstruction and the generation from an initial frame. To predict the future timesteps of a sequence, we apply the motion operator on the latent encoding of the first time step. Figure 3.5 on the left are the results for sprites, and on the right of MUG video sequences. Next, we report sequences reconstructed by swapping motion variables to evaluate disentanglement. We start by encoding two sequences $\mathbf{x}^1_{1:T}$ and $\mathbf{x}^2_{1:T}$ to their latent representations $(\mathbf{z}^1, \mathbf{q}^1_{1:T})$ and $(\mathbf{z}^2, \mathbf{q}^2_{1:T})$, next we swap the motion variables $(\mathbf{z}^1, \mathbf{q}^2_{1:T})$ and $(\mathbf{z}^2, \mathbf{q}^1_{1:T})$ between the two representation spaces, and then pass the resulting representations through the decoder to generate the sequences $\mathbf{x}^{1\to 2}_{1:T}$ and $\mathbf{x}^{2\to 1}_{1:T}$. We refer to Section 3.3 for the description of content and motion space. Figure 3.6, on the left, are the pair of consecutive rows of original sequences and on the right are the sequences generated by swapping the motion representations. We can see that swapping the motion part does not affect the identities of sequences.

We now evaluate the image-to-sequence task to investigate the suitability of our model for a controlled generation. We first encode the image to its content and phase space coordinate in different motion spaces. Next, the respective operators are used to unroll the trajectories in phase space, which are combined with content and mapped to the image space using a decoder network. Figure 3.7 shows examples of decoding different motions from the same input image. For sprites, the actions are in order 'walk', 'spell card', and 'slash'; for MUG, they are ordered 'anger', 'disgust', 'fear', 'happiness', 'sadness' and 'surprise'. We observe that the visual dynamics associated with all the operators are well separated.

We further provide extended qualitative samples. Figure 3.14 shows the results of conditional

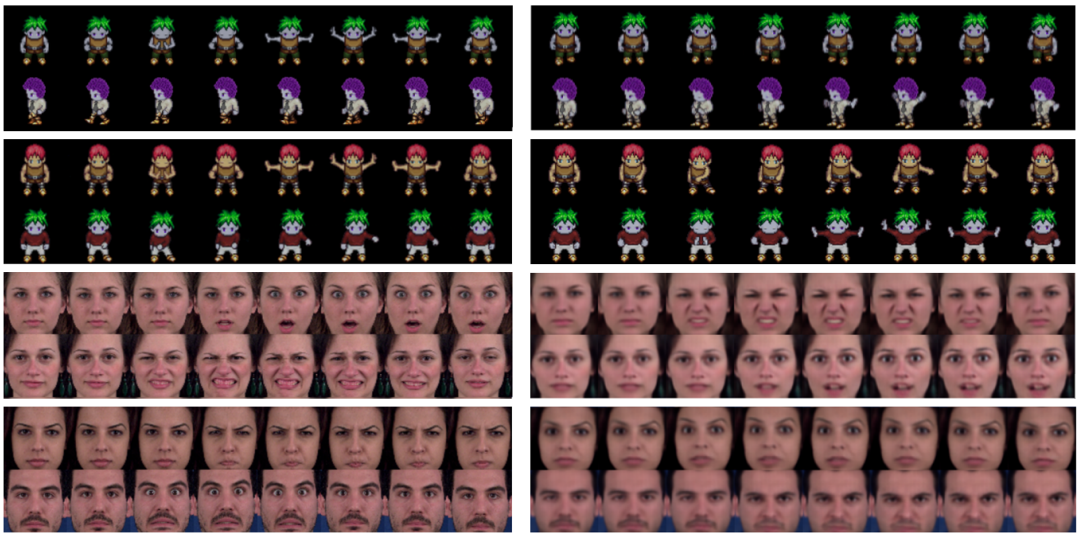


Figure 3.6: Qualitative demonstration of content and motion disentanglement. On the left side, we show rows of original sequence pairs and on the right are the reconstructions after swapping the motion variables in the latent space.

sequence generation, and Figure 3.17 shows the results of motion swapping. Figure 3.15 and 3.16 further show examples of image-to-sequence generation. We generate 16 frames in future conditioned on an initial starting frame.

3.6 Ablation Study

We perform ablation studies to investigate the effectiveness of our dynamical model. We first adapt our model to scenarios where action variables are **u** not available and compare it to other choices of the dynamical model.

3.6.1 Unconditional Dynamics

In our proposed formulation presented in Section 3.3.2, we introduce the action variable **u** to map a sequence to its corresponding phase space. This enables the separation of dynamics and facilitates the controlled generation of motion sequences. It is important to note that using action variables does not impose any restrictions on the Hamiltonian dynamics.

In this section, we extend our formulation to handle sequences where action variables are unavailable. To address this, we introduce a factorisation of the phase space into K symmetry groups, where the Hamiltonian takes the form $\mathbf{H} = \textit{block-diagonal}(\mathbf{H}_1, \ldots, \mathbf{H}_K)$. This factorization allows us to evolve the trajectory for any arbitrary sequence $\mathbf{x}_{1:T}$ by simultaneously evolving all the operators involved in each symmetry group as,

$$\mathbf{s}_t = f(\mathbf{s}^k_{t-1}; \omega_k, t) = \textit{block-diagonal}(e^{t\mathbf{H}_1}\mathbf{s}^1_{t-1}, \ldots, e^{t\mathbf{H}_K}\mathbf{s}^K_{t-1}) \quad \forall t > 1 \tag{3.22}$$

Figure 3.7: Examples of sequences generated by an action of Hamiltonian operator on the phase space representation of the starting frame. We observe that all dynamics are well separated, demonstrating the disentanglement of different actions. This result shows the benefit of the symplectic structure in motion space.

By employing this modified approach, our framework becomes applicable to a broader range of scenarios. However, it is essential to note that we do not have direct control over the action generated by the dynamics in this formulation. The type of motion generated depends on the initial position and momentum variables. Additionally, it should be noted that the operators $\mathbf{H}_k$ in this context may not correspond to specific actions. Still, instead, they can represent more general properties that are conserved and shared across different motions. For example, other operators could capture variations in the magnitude of action movements, such as expressions like smiling or surprise.

To empirically investigate this aspect, we map a starting frame to the phase space and generate a sequence using individual $\mathbf{H}_k$ operators and the overall $\mathbf{H}$. This allows us to explore the impact and contribution of each operator in generating distinct motion characteristics or capturing specific aspects of the dynamics. We acknowledge that the control over generated actions is indirect, with the motion characteristics influenced by initial conditions and the conserved properties described by the operators $\mathbf{H}_k$.

The generated motion sequences are depicted in Figure 3.9. Here, the first six rows represent sequences generated using individual $\mathbf{H}_k$ operators, while the last row represents sequences generated using the combined $\mathbf{H}$ operator. By examining the generated sequences, we can observe that the individual operators successfully capture the varying extent of motion. Additionally, Figure 3.8 illustrates the model's performance in sequence generation and motion transfer tasks, showcasing how well the model performs in generating coherent and realistic sequences and its

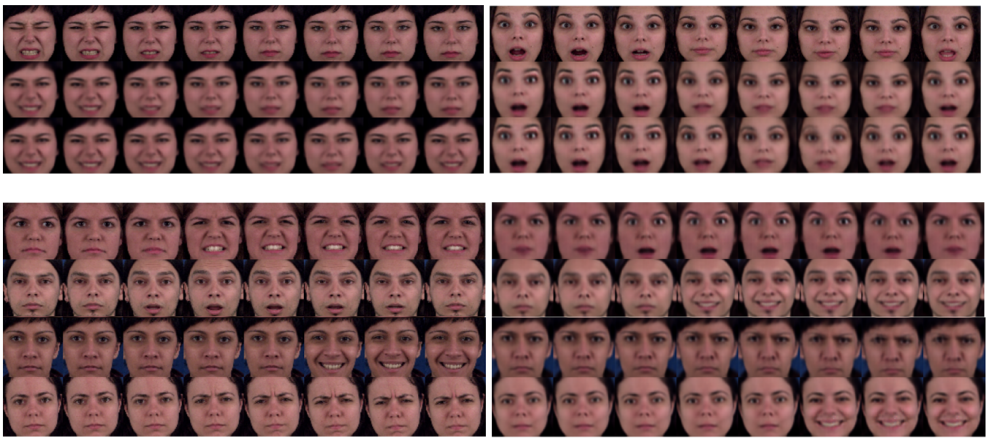


Figure 3.8: Unconditional Hamiltonian approach. On top, the first row is an original sequence, the second row is the reconstruction, and the third row is generated by an action of the Hamiltonian operator on the phase space representation of the first frame in the sequence. On the bottom is an example of a motion swap; on the left are two original motions, and on the right are sequences generated by swapping the motion variables.

ability to transfer motion from one sequence to another.

3.6.2 *HALO* vs other Dynamical Approaches

In this section, we compare *HALO* with other choices of dynamical models. We first introduce the choice of linear models and then discuss the results.

3.6.2.1 Linear Model

A linear dynamical is defined as,

$$\mathbf{h}_t = \mathbf{A}_{t-1}\mathbf{s}_{t-1} + \mathbf{B}_{t-1}\mathbf{h}_{t-1} + \mathbf{b} \tag{3.23}$$

where $\mathbf{h}_t$ is a hidden state, and $\{\mathbf{A}, \mathbf{B}, \mathbf{b}\}$ are learnable parameters. To generate the trajectory $\mathbf{x}_{1:T}$, we combine the state coordinates $\mathbf{h}_{1:T} = \{\mathbf{h}_1, \ldots, \mathbf{h}_T\}$ with the content variable $\mathbf{z}$ and pass the joint representation through the decoder network. We report the performance of a linear model in conditional and unconditional settings.

3.6.2.2 Positional Encoding

We generate a simplistic baseline using a fixed Fourier encoding representation. Specifically, for a sequence of frames $\mathbf{x}_{1:T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ we map it to a content variable $\mathbf{z}$ and a frame $\mathbf{x}_t$ to a phase $\phi_t \in [-1, 1]$. We then generate $T - t$ linearly separated phase coordinates $\{\phi_t, \ldots, \phi_T\} \in [\phi_t, 1 + \phi_t]$ and define the motion space representation as,

$$\mathbf{s}_t = \{\sin(\phi_t 2^k), \cos(\phi_t 2^k)\}_{k=1}^{k=\lfloor d/2 \rfloor}. \tag{3.24}$$

Figure 3.9: Unconditional Hamiltonian approach. An example of image-to-sequence generation. The first column is the starting frame, the first six rows correspond to the sequence generated by the action of $k-th$ block of $\mathbf{H}$, and the last row is the sequence generated by the full $\mathbf{H}$.

where d is the size of motion space. We impose a Gaussian prior on the phase coordinates $p(\phi_t) = \mathcal{N}(0,1)$.

Method	**Accuracy**↑	$H(\mathbf{y}\|\mathbf{x})$↓	$H(\mathbf{y})$ ↑	**IS** ↑
HALO	**0.929**	**0.108**	**1.778**	**5.312**
Linear	0.548	0.722	1.553	2.295
RNN	0.580	0.759	1.743	2.675
HALO (unconditional)	0.750	0.187	1.762	4.830
Linear (unconditional)	0.451	0.962	1.525	1.756
RNN (unconditional)	0.550	1.015	1.658	1.902
Positional Encoding	0.152	0.978	1.150	1.188

Table 3.4: Results of a classifier on MUG for different choices of dynamical models. The high score of accuracy and Inter-Entropy $H(\mathbf{y})$ while low scores of Intra-Entropy $H(\mathbf{y}|\mathbf{x})$ are expected from a better model.

We describe the qualitative results in Table 3.4-3.5. The Hamiltonian model achieves the best performance across all scores. We observe all models except the positional encoding achieve comparable performance on identity prediction. We speculate this could be due to the non-changing dynamics, which makes predicting the identity from a sequence of static images much easier for a classifier. Due to the failure of positional encoding, we omit it from the rest of the discussion. We restrict the qualitative analysis to conditional models. Figure 3.13 describes the results on image-to-sequence generation, further demonstrating that the Hamiltonian dynamics are consistent in long-term prediction and prevent constant information flow to motion variables. Figure 3.10 we compare the motion transfer by replacing Hamiltonian with a linear dynamical

Identity	Accuracy↑
HALO	0.998
Linear	0.996
RNN	**1.000**
HALO (unconditional)	0.994
Linear (unconditional)	0.974
RNN (unconditional)	0.998
Positional Encoding	0.009

Table 3.5: Comparison to other baselines in terms of accuracy of the identity of sequences. This shows our model can preserve content when the motion representation is changed.

model and RNN. We observe that, unlike *HALO*, the variations in dynamics get constant for Linear and RNN models. This result shows the benefit of Hamiltonian dynamics as, by definition, they ensure the phase space coordinates change over time, preventing the encoder neural network from channelling any static information in the motion space, which is vital for explicit disentanglement of content and motion variables. Overall, the Hamiltonian formulation outperforms other approaches and works best across all tasks.

Next, we investigate the benefit of constant energy resulting from the choice of *HALO* as a dynamical operator in the motion subspace of latent space.

3.6.3 What are the Benefits of Constant Energy?

The Hamiltonian formulation maintains constant energy over time. Such a choice is beneficial for generating long-term sequences. In this part, we generate long sequences using our dynamical model and look at the evaluation of energy over time.

The total Hamiltonian energy in the phase space is given by,

$$\mathsf{E} = \frac{1}{2}\mathbf{s}^T \mathbf{M}\mathbf{s}. \tag{3.25}$$

where $\mathbf{s} = (\mathbf{q}, \mathbf{p})$, and $\mathbf{M}$ is a symmetric matrix. Let $\mathbf{M}$ be a 2×2 block matrix $\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{C} \end{pmatrix}$. We can expand the energy term as,

$$\mathsf{E} = \frac{1}{2}\mathbf{q}^T \mathbf{A}\mathbf{q} + \frac{1}{2}\mathbf{p}^T \mathbf{C}\mathbf{p} + \frac{1}{2}\mathbf{q}^T \mathbf{A}\mathbf{p} + \frac{1}{2}\mathbf{p}^T \mathbf{B}\mathbf{q}. \tag{3.26}$$

The first term is potential energy (PE), the second is kinetic energy (KE), and the last two combined are non-separable terms. When B is zero, the energy is entirely separable into KE and PE terms. The non-separable Hamiltonian is common in many physical problems, for instance, rigid body dynamics and many others appearing in quantum mechanics. For details, we refer

Figure 3.10: Here, we demonstrate that our method outperforms other dynamical model baselines used for disentanglement. In each patch, rows one and two are original, and rows three and four are obtained by swapping the motion components of latent space. The top is a Linear Model, the centre is RNN, and the last is *HALO*.

to Easton (1993). The choice of the unconstrained linear form of Hamiltonian was motivated to allow more flexibility to the model to learn in a data-driven way.

In Figure 3.11, we report the plot of energy over time for an image under different motion dynamics. The change in the individual energy shows the dynamics are not constant; this is also evident from the corresponding image sequences shown in the plot. As dynamics evolve, the total energy is strictly conserved, demonstrating that the trajectory cannot diverge from the learned symplectic structure. The results show the benefit of our model in generating long-term sequences. We want to remark that the energy terms should be interpreted carefully. It might not have any equivalence to the energy of a physical system; what it does is that it provides constraints to use the time translation symmetry of the dynamics.

We summarise the key findings of ablation in Table 3.6. Our formulation proves helpful in learning disentangled representations, outperforming various baselines. We demonstrated its use for the controlled generation of image sequences. The effectiveness of our approach is a direct consequence of the *symplectic* geometry in the phase space, which prevents trajectories from deviating from the motion manifold. The choice of the quadratic form of energy provides a relationship among latent components, which we speculate is critical for the interpretability of

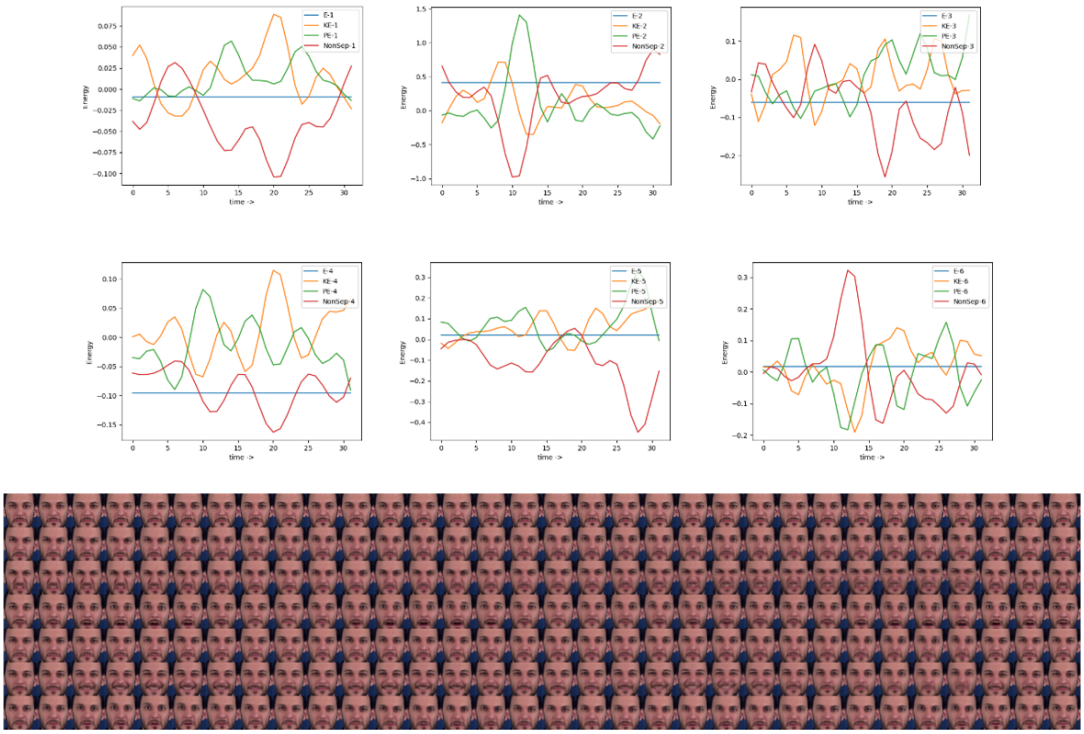


Figure 3.11: We map a starting frame to the phase space and use the operators $\mathbf{H}$ to generate the phase space trajectory, which is then mapped to data space using the decoder network. At the top is the plot of energy vs time of the operators $\mathbf{H}_k$ (E is the total energy,KE is the kinetic energy term, PE is the potential energy, and $NonSep$ is the non-separable term). Below, each row is the sequence generated by the action of $\mathbf{H}_k$.

Dynamics	Image-To-Seq.	Motion Swap	Structure
HALO	✓	✓	Symplectic
Linear	✗	✓	✗
RNN	✗	✓	✗
Positional Encoding	✗	✗	✗

Table 3.6: The benefits of our formulation over other dynamical models. The details on positional encoding are discussed in Ablation 3.6.2.2.

actions.

3.6.4 Rotating MNIST

In this section, we investigate the performance of our approach in predicting the rotations of MNIST digits. We use an unconditional version of our model for this part. Following the procedure of Casale et al. (2018), we generated sequences of 16 time steps by rotating the

images of digit “3". We followed the same training procedure. In Figure 3.12, part (a), the first row is the input sequence, and the second is a reconstruction. In part (b), we show three sequences generated by random initial phase space coordinates. The network architecture for MNIST experiments is outlined in Table 3.11, 3.12 and 3.13. In Table 3.7, we compare the mean squared error (MSE) of our model with the other related methods (Yildiz et al., 2019; Casale et al., 2018). Our model achieves comparable performance to GPPVAE. The ODE^{2}VAE performs best in terms of MSE; this can be attributed to using a second-order latent ODE model. In contrast, our formulation only uses first-order dynamics, which provides extra computational efficiency. Furthermore, compared to GPPVAE, we don’t have costly kernel computations.

We want to remark datasets such as stochastic movingMNIST (Denton and Birodkar, 2017), also used in a few disentanglement research, are not a good application of our model due to the nature of dynamics generated by an action of a random transformation. Such a dataset will be an example of a failure case of our model due to the association of a constant energy with a dynamical model in a latent space for a dataset that contains sequences with no constant of motion. The Hamiltonian model relies on a dataset of data sequences where dynamics follow a conserved quantity and can be associated with constant energy. This assumption may or may not hold for stochastic MNIST data, depending on the nature of random movements.

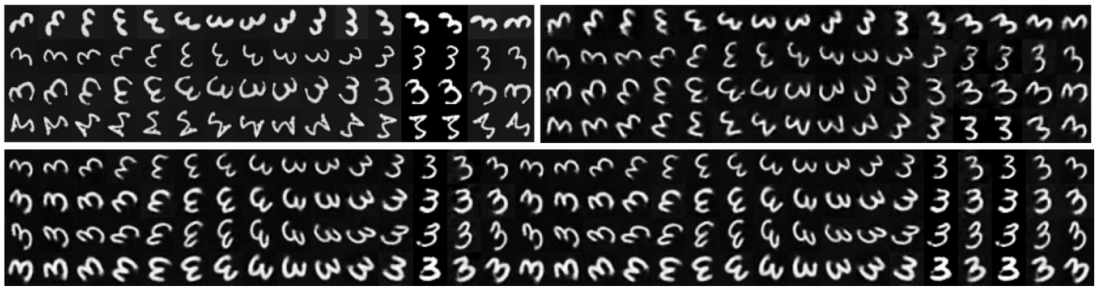


Figure 3.12: Results on Rotating MNIST with a learnable Hamiltonian operator. On the top left, we have four input sequences, and on the right, their reconstruction; on the bottom, we have four sequences generated by an action of Hamiltonian on the state space coordinate of the frame in the first column.

3.7 Conclusion

This chapter introduced a novel sequential VAE, namely *HALO*, that aims to disentangle motion from content in image sequences. *HALO* formulation leverages Hamiltonian latent operators to associate conserved quantities with the dynamics of the sequences. By the principles of Hamiltonian dynamics, the non-zero energy term ensures that the motion space varies over time, preventing the encoder neural network from channelling any static information in the motion variables and providing a meaningful disentanglement of content and motion. The quantitative

Model	MSE ↓
GPPVAE-dis (Casale et al., 2018)	0.0306
GPPVAE-joint (Casale et al., 2018)	0.0280
ODE^{2}VAE (Yildiz et al., 2019)	0.0204
ODE^{2}VAE-KL (Yildiz et al., 2019)	0.0184
HALO (Ours)	0.0208

Table 3.7: Mean squared error on a test set of rotating MNIST. The 75% of sequences of rotating digit 3 were used for training, and the rest for testing purposes.

results of our experiments, both with conditional and unconditional models, surpassed the performance of existing baselines. Additionally, we demonstrated the disentanglement qualitatively by showcasing motion-swapping capabilities. In the conditional model, we associate each action with a unique Hamiltonian, which is critical for the controlled generation task of image-to-sequence generation. HALO exhibits the ability to generate long-term trajectories and traverse the motion manifolds associated with different actions in the latent space. However, we acknowledge that HALO may have limitations in handling irregularly sampled sequences, changes in tempo, or reversals.

In conclusion, this chapter underscored the importance of incorporating geometry-based inductive biases to facilitate learning disentangled latent space—particularly its relevance in introducing temporal coherence in motion subspace. Our findings were presented at the NeurIPS conference in December 2022, highlighting the significance and novelty of our research.

DGMs, particularly VAEs, have gained considerable popularity as effective frameworks for learning latent spaces in high-dimensional data manifolds, as evidenced by our contributions in this chapter. However, one notable concern is the vulnerability of VAEs to adversarial attacks, where imperceptible perturbations in input samples can lead to the reconstruction of meaningless samples. These vulnerabilities expose the limitations of VAEs and necessitate further investigation into their robustness to ensure their reliability and effectiveness.

In the subsequent chapter, our focus will shift towards studying the robustness of VAEs by employing tools from differential geometry. Differential geometry offers a solid mathematical framework for analysing the properties of smooth manifolds and comprehending their geometric structures. By utilising these tools, we aim to establish the smoothness of the latent space of encoder-decoder neural networks in VAEs and show its connection to their adversarial vulnerabilities.

Encoder Architecture of Sprites and MUG	
Conv2d	kernels: 256, kernelSize: $(5,5)$, stride: $(1,1)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
Conv2d	kernels: 256, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
Conv2d	kernels: 256, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
Conv2d	kernels: 256, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
Conv2d	kernels: 256, kernelSize: $(5,5)$, stride: $(1,1)$, padding: $(2,2)$
BatchNorm2d $\rightarrow$ LeakyReLU(0.2) $\rightarrow$ Rearrange('b c w h -> b (c w h)')	
Linear	in:=(c w h), out: 4096
	BatchNorm1d $\rightarrow$ LeakyReLU(0.2)
Linear	in: 4096, out: 2048
	BatchNorm1d $\rightarrow$ LeakyReLU(0.2)
Linear	in: 2048, out: h
	BatchNorm1d $\rightarrow$ LeakyReLU(0.2)

Table 3.8: Encoder network

Decoder Architecture of Sprites and MUG	
Linear	in: h, out: 4096
	BatchNorm1d $\rightarrow$ LeakyReLU(0.2)
Linear	in: 4096, out:(c w h)
BatchNorm1d $\rightarrow$ LeakyReLU(0.2) $\rightarrow$ Rearrange('b (c w h) -> b c w h')	
ConvTranspose2d	kernels: 256, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
ConvTranspose2d	kernels: 256, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
ConvTranspose2d	kernels: 256, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
ConvTranspose2d	kernels: 256, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
ConvTranspose2d	kernels: 256, kernelSize: $(5,5)$, stride: $(1,1)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ Tanh()

Table 3.9: Decoder network

Content and Motion Architecture of Sprites and MUG					
Content		Position		Momentum	
LSTM	in: h, out: z	Linear	in: $h+k$, out: v	Linear	in: $h+k$, out: v
Linear_{μ}	in: z, out: z	BatchNorm1d $\rightarrow$ LeakyReLU(0.2)		BatchNorm1d $\rightarrow$ LeakyReLU(0.2)	
$\text{Linear}_{\log\sigma}$	in: z, out: z	Linear	in: v, out: v	Linear	in: v, out: v
		BatchNorm1d $\rightarrow$ LeakyReLU(0.2)		BatchNorm1d $\rightarrow$ LeakyReLU(0.2)	
		Linear_{μ}	in: v, out: q	TCN	kernelSize: 3, pad: 2, stride: 1
		$\text{Linear}_{\log\sigma}$	in: v, out: q	Linear_{μ}	in: v, out: p
				$\text{Linear}_{\log\sigma}$	in: v, out: p

Table 3.10: Content and Motion network. TCN stands for temporal convolution network.

Encoder Architecture MNIST	
Conv2d	kernels: 32, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ ReLU()
Conv2d	kernels: 64, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ ReLU()
Conv2d	kernels: 128, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(2,2)$
	BatchNorm2d $\rightarrow$ ReLU()
Linear	in: $(c \times w \times h)$, out: 4096
	BatchNorm1d $\rightarrow$ ReLU()
Linear	in: 4096, out: 256
	BatchNorm1d $\rightarrow$ ReLU()

Table 3.11: Encoder network MNIST

Decoder Architecture MNIST	
Linear	in: 20, out: 4096
BatchNorm1d $\rightarrow$ ReLU()	
Linear	in: 4096, out: $(c \times w \times h)$
BatchNorm1d $\rightarrow$ ReLU() $\rightarrow$ Rearrange('b (c w h) -> b c w h')	
ConvTranspose2d	kernels: 128, kernelSize: $(3,3)$, stride: $(1,1)$, padding: $(0,0)$
BatchNorm2d $\rightarrow$ ReLU()	
ConvTranspose2d	kernels: 64, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(1,1)$
BatchNorm2d $\rightarrow$ ReLU()	
ConvTranspose2d	kernels: 32, kernelSize: $(5,5)$, stride: $(2,2)$, padding: $(1,1)$
BatchNorm2d $\rightarrow$ ReLU()	
ConvTranspose2d	kernels: 1, kernelSize: $(5,5)$, stride: $(1,1)$, padding: $(2,2)$
BatchNorm2d $\rightarrow$ Sigmoid()	

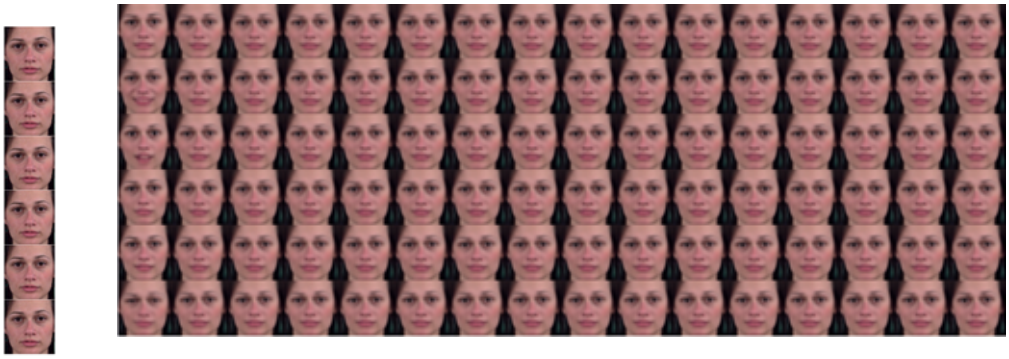
Table 3.12: Decoder network MNIST

Motion Network MNIST			
Position		Momentum	
Linear	in: 256, out: 320	Linear	in: 256, out: 320
BatchNorm1d $\rightarrow$ LeakyReLU(0.2)		BatchNorm1d $\rightarrow$ LeakyReLU(0.2)	
Linear	in: 320, out: 20	Linear	in: 320, out: 20
BatchNorm1d $\rightarrow$ LeakyReLU(0.2)		BatchNorm1d $\rightarrow$ LeakyReLU(0.2)	
Linear_{μ}	in: 20, out: 20	TCN	kernelSize: 4, pad: 3, stride: 1
$\text{Linear}_{\log\sigma}$	in: 20, out: 20	Linear_{μ}	in: 20, out: 20
		$\text{Linear}_{\log\sigma}$	in: 20, out: 20

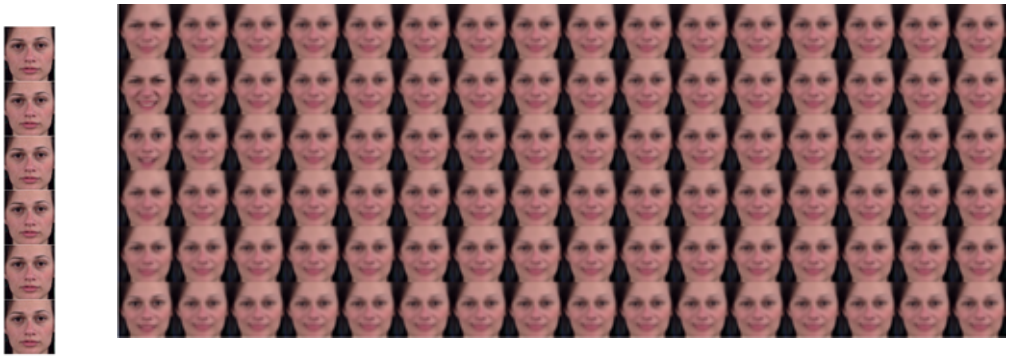
Table 3.13: Motion network MNIST. TCN stands for temporal convolution network.

Classifier Architecture	
Conv2d	kernels: 64, kernelSize: $(5,5)$, stride: $(4,4)$, padding: $(1,1)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
Conv2d	kernels: 128, kernelSize: $(5,5)$, stride: $(4,4)$, padding: $(1,1)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
Conv2d	kernels: 256, kernelSize: $(5,5)$, stride: $(4,4)$, padding: $(1,1)$
	BatchNorm2d $\rightarrow$ LeakyReLU(0.2)
Linear	in: $(c \times w \times h)$, out: 1024
	BatchNorm1d $\rightarrow$ LeakyReLU(0.2)
LSTM	in: 1024, out: 512
	BatchNorm1d $\rightarrow$ LeakyReLU(0.2)
Linear	in: 512, out: 256
	BatchNorm1d $\rightarrow$ LeakyReLU(0.2)
Linear	in: 256, out: K

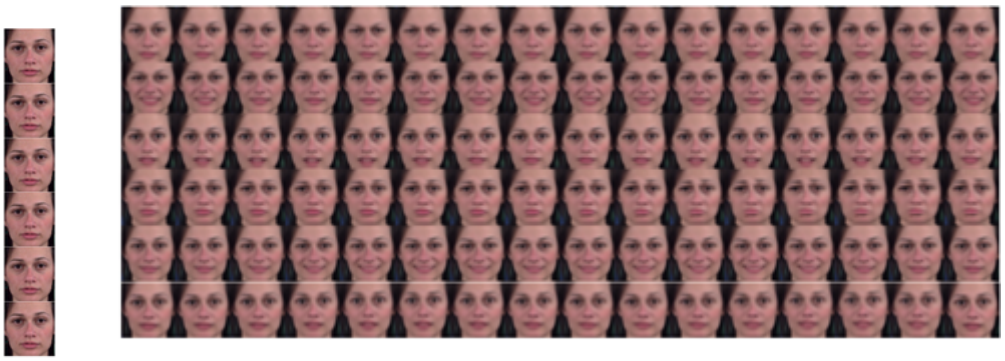
Table 3.14: Classifier network used for evaluation. For the attribute classification task, K is set to the number of attributes; for the action classification, it is set to the number of actions.



(a) Linear Model

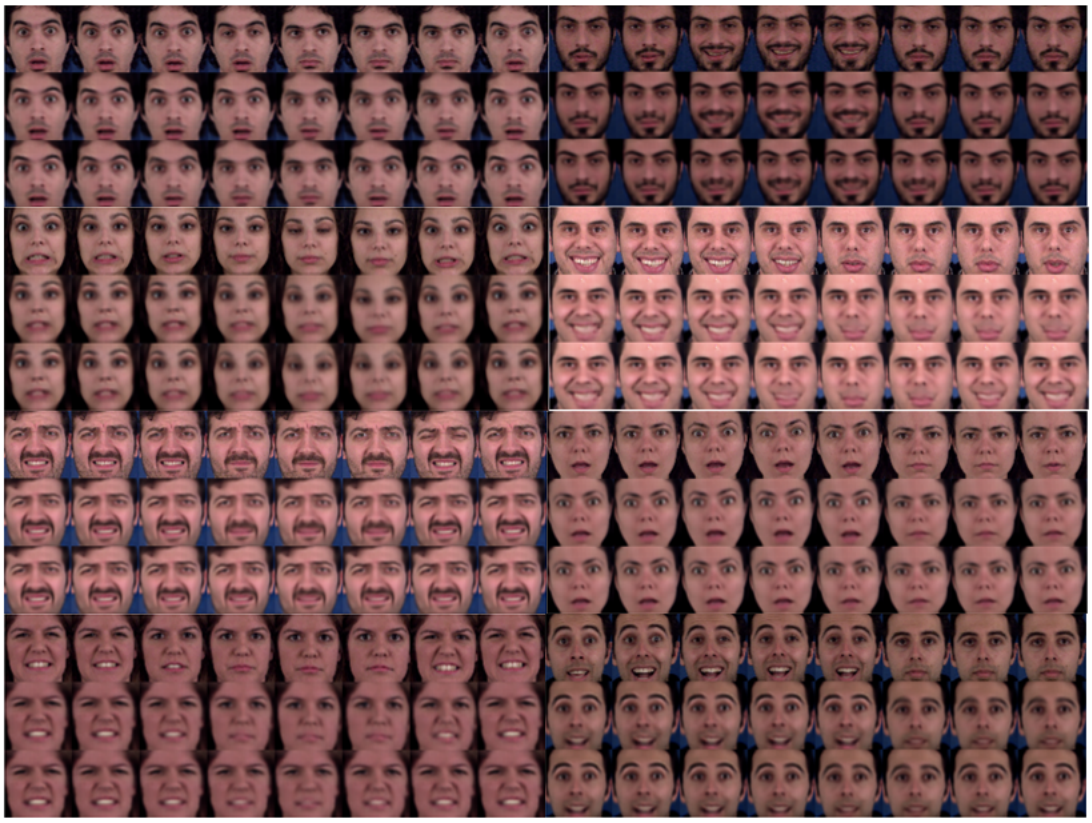


(b) RNN Model



(c) *HALO*

Figure 3.13: Results on Image to sequence generation. On the left is the starting frame, and on the right are different motions generated by the dynamical models.



(a) Conditional Sequence Generation. The first row is the original sequence, the second row is a reconstructed sequence, and the third is generated by an action of a dynamical model on the first time frame

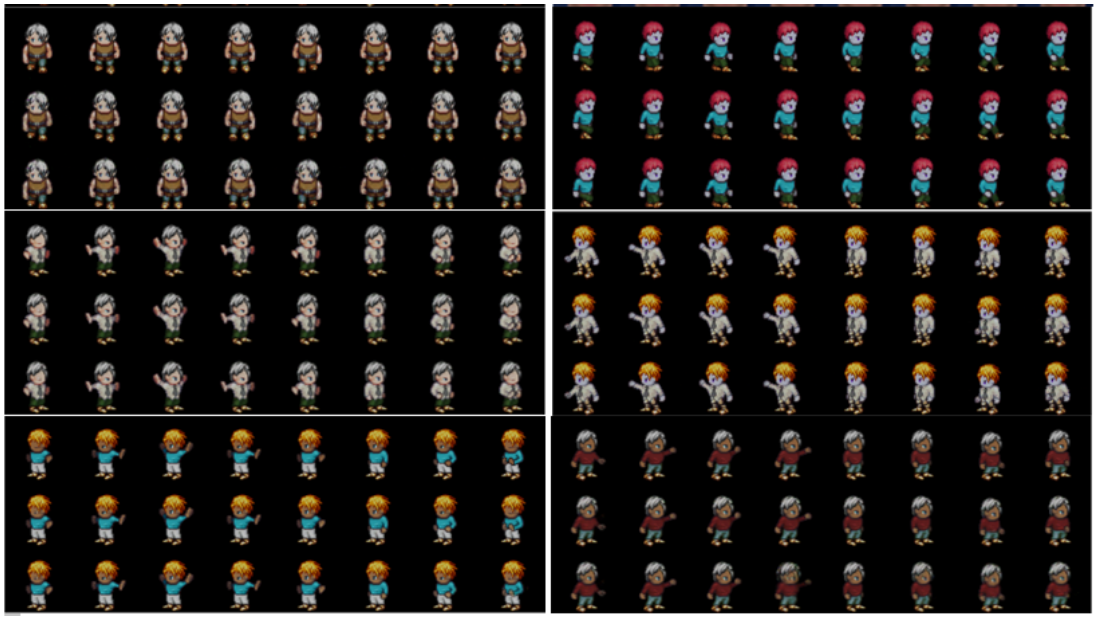


Figure 3.14: Conditional Sequence Generation. The first row is the original sequence, the second row is a reconstructed sequence, and the third is generated by an action of a dynamical model on the first time frame

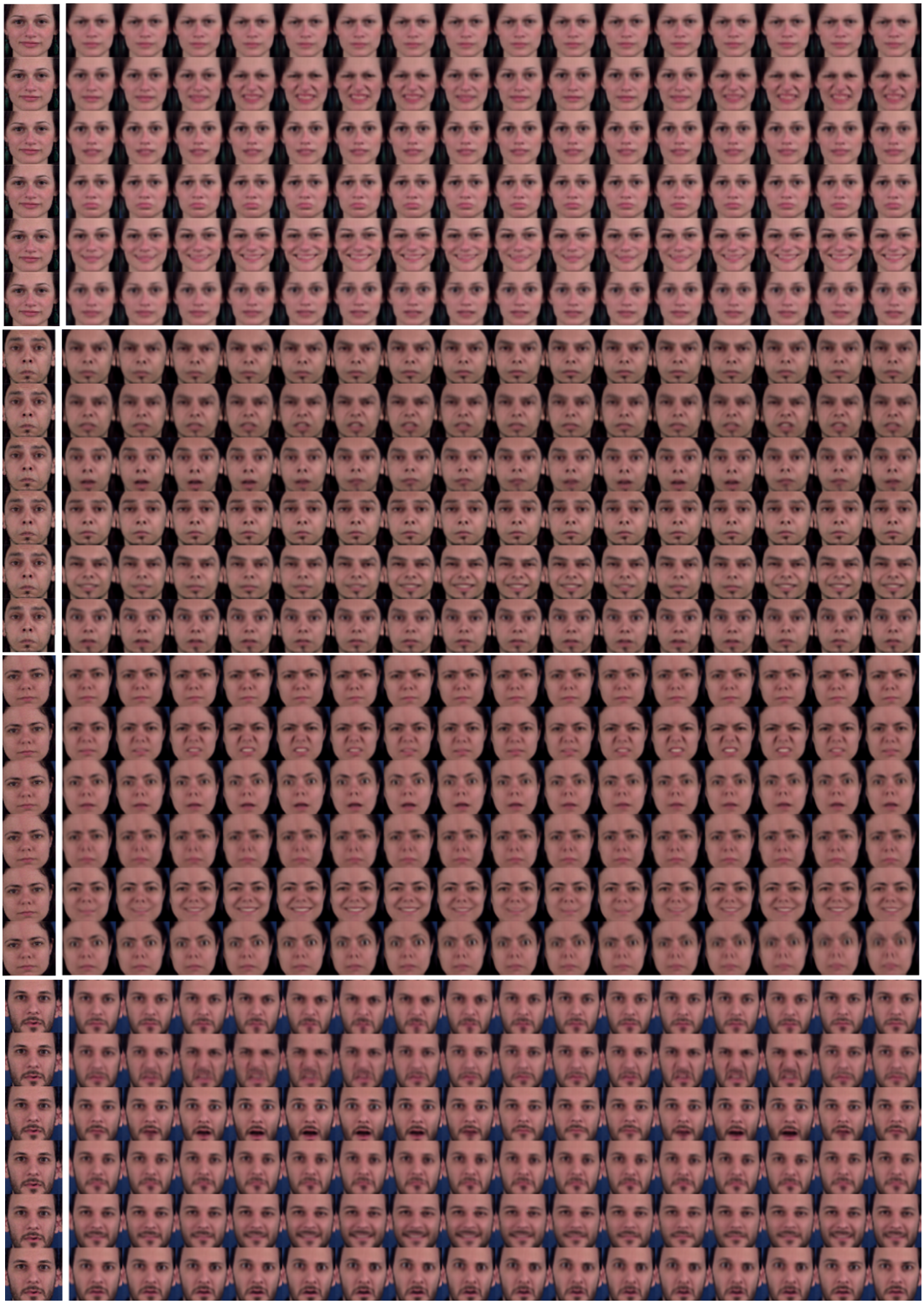


Figure 3.15: Image to Sequence generation. We generate dynamics of different actions from a given image. Each row is a unique action generated by the operator associated with that action.

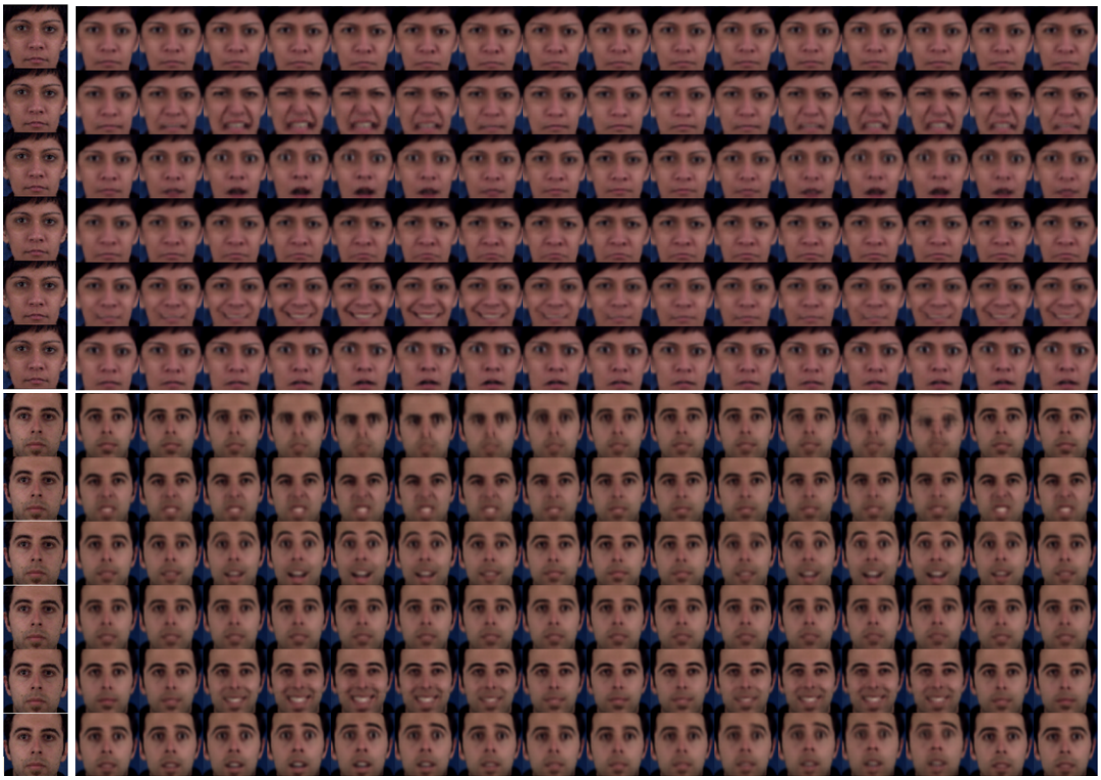


Figure 3.16: Image to Sequence generation. We generate dynamics of different actions from a given image. Each row is a unique action generated by the operator associated with that action.

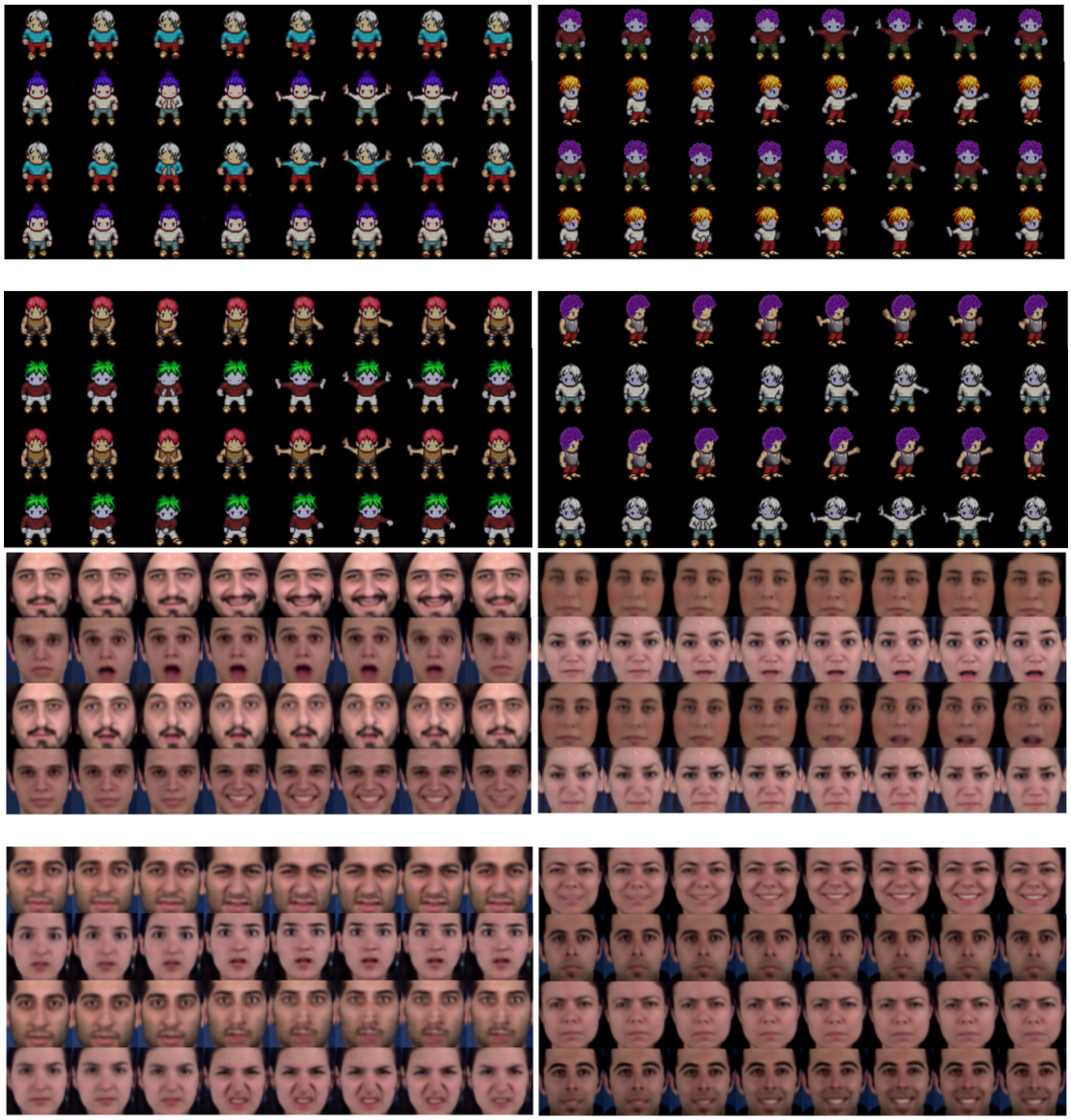


Figure 3.17: Motion Swapping. In each patch, the first two rows are the original sequence, and the next two rows are obtained by swapping the motion variables of two sequences.

Chapter 4

Adversarial Robustness of VAEs through the Lens of Local Geometry

Variational autoencoders (VAEs) are a class of deep latent variable models that map samples from easy-to-sample latent distribution to complex data distributions using a generator neural network while simultaneously learning a meaningful latent representation of data using an inference network. This attractive property of VAE makes them a popular choice in various domains, including image synthesis, natural language processing, and healthcare (Vahdat and Kautz, 2020; Gómez-Bombarelli et al., 2018). While VAEs have shown promising results, the learned latent space suffers from topological and geometrical issues. Such problems are attributed to the failure of the inference network to cover the latent space because of the mismatch between the aggregated variational posterior and the prior distribution. Subsequently, regions of low/zero density in the latent space showcase a failure to incorporate *smoothness* property, leaving VAEs vulnerable to noise perturbations in input data. Thus, an adversary can manipulate the input data such that its latent representation deviates from the learned latent manifold, generating samples with undesired changes.

The previous chapter emphasised the importance of incorporating geometry to introduce properties like disentanglement and temporal coherence. The choice of the Hamiltonian operator resulted in a structured latent space where motions are associated with an energy term that encourages the smoothness of the latent space. In classical deep latent variable models, such properties are not explicitly encoded. Here, we utilise differential geometry to investigate the smoothness of the latent space of a classical VAE with a Gaussian prior. We show how a lack of smoothness is associated with high-curvature regions that leave VAEs vulnerable to adversarial attacks.

Geometry-based metrics such as the pullback metric tensor help understand the connection between the infinitesimal regions of the input space mapped to regions in the latent space through a neural network. Thus, they are valuable tools for understanding the smoothness of

latent space and its sensitivity to perturbations in data samples.

Analysing the impact of adversarial perturbations on the latent space can identify vulnerable regions and provide insights into developing defences to enhance the robustness of VAEs. The critical finding of this chapter are four-folds,

- The distortions in latent space can be viewed through the lens of Riemannian geometry.
- An optimal adversary can attack VAEs by exploiting the anisotropic nature of the pullback metric tensor induced by the inference network.
- The higher values of β parameter in $\beta-$VAE formulation reduce distortions in the latent space, thereby improving robustness.
- By introducing a constraint to fill the empty regions in the latent space implemented as a *mixup* loss is a promising technique for improving robustness to adversarial attacks.

The chapter is structured as follows. We first motivate the problem in the introduction Section 4.1; next, we provide a survey of related work in Section 4.2. Section 4.3 establishes the connection between an optimal adversary and a stochastic pullback metric tensor of an inference network. We also introduce the scores for evaluating robustness using pullback geometry. Next, Section 4.4 presents empirical results, followed by a detailed discussion in Section 4.4. Finally, we conclude with the key takeaways in Section 4.6. The relevant technical definitions are presented in a background Chapter 2.1.

4.1 Introduction

Variational autoencoders (VAEs) belong to a class of deep generative models that utilise a stochastic encoder-decoder network (Kingma and Welling, 2014). The encoder parameterises the variational distribution over latent variables conditioned on input data samples, while the decoder estimates the data distribution by sampling from the latent distribution. Thus, VAEs serve a dual purpose of estimating data density and providing a rich representation space with uncertainty quantification. In recent years, several works have shown the application of VAEs to various domains, including high-fidelity image generation (Vahdat and Kautz, 2020), music generation (Roberts et al., 2017), video generation (Wu et al., 2021), and many others.

Despite their success, VAEs, like other machine learning models, are also not immune to adversarial attacks, as demonstrated in recent works (Tabacof et al., 2016; Gondim-Ribeiro et al., 2018; Willetts et al., 2021; Kos et al., 2017; Kuzina et al., 2021). The attacks on VAEs have raised concerns regarding the robustness and reliability of these models, such as generating misleading or incorrect outputs, compromising data privacy, etc. Therefore, it is crucial to study the vulnerability of VAEs to adversarial attacks and develop robust defences against them.

In a typical adversarial attack scenario, an adversary aims to perturb an input sample so that it results in a significant change in its latent encoding, leading to unexpected output. This mechanism involves solving the optimisation problem involving maximisation of a particular loss function (introduced later in Equation 4.2), which is generally solved using stochastic gradient methods (Chakraborty et al., 2021; Szegedy et al., 2014). For a more comprehensive overview of attacks on VAEs and other generative models, we refer readers to (Sun et al., 2023).

The primary reason for the vulnerability of VAEs to attacks is the distortion in the latent space resulting from the mismatch between approximated posterior and latent space prior, also known as the *posterior-prior gap* (Mathieu et al., 2019). As a result, the learned latent space becomes non-smooth, and similar inputs tend to be mapped to distant regions in the latent space under the Euclidean metric. Several methods have emphasised the importance of reducing the *posterior-prior gap* for improving the robustness of VAEs and encouraging disentanglement of latent factors (Mathieu et al., 2019; Willetts et al., 2021).

The $\beta-$ VAE formulation introduced by Higgins et al. (2017) incorporates a parameter β to control the gap directly. Other approaches such as Chen et al. (2018b); Kim and Mnih (2018); Esmaeili et al. (2019) utilise the total correlation (TC) term to disentangle the latent coordinates of VAEs and promote smoothness in the latent space, thereby improving the robustness. However, a central limitation of these approaches is that they cause over-smoothing of the reconstructed samples and require a careful training mechanism to balance the regularisation term.

Willetts et al. (2021) address the aforementioned problem by introducing a TC term in hierarchical VAEs, offering robustness and sharp reconstruction capabilities. Despite the improvement, the existing approaches do not provide meaningful insight into the robustness property, mainly due to the use of regularisation terms, which are optimised using gradient methods with a limited understanding of the approximated posterior learned by the inference network. Specifically, the notion of a small change in the input to a large change in latent space is not well established. Consequently, comparing these schemes relies on the visual inspection of distorted images at varying magnitudes of adversarial loss, lacking a rigorous quantitative assessment.

Recently, Camuto et al. (2021) proposed a theoretical framework that considers the uncertainty of the encoder for studying the robustness of VAEs. However, attacks in the input space do not consider the effect of geometry induced by an encoder or decoder mapping. In another recent work, Kuzina et al. (2021) introduced an asymmetric KL term to capture the difference between the latent representation of input and its perturbation. They determine the magnitude of perturbation using the Jacobian matrix of the latent code evaluated at the input perturbation. Nevertheless, their optimisation objective does not take a geometrical perspective. Moreover, they do not consider the contribution of the standard deviation term when computing the Jacobian matrix. Thus not accounting for the uncertainty in the representation space.

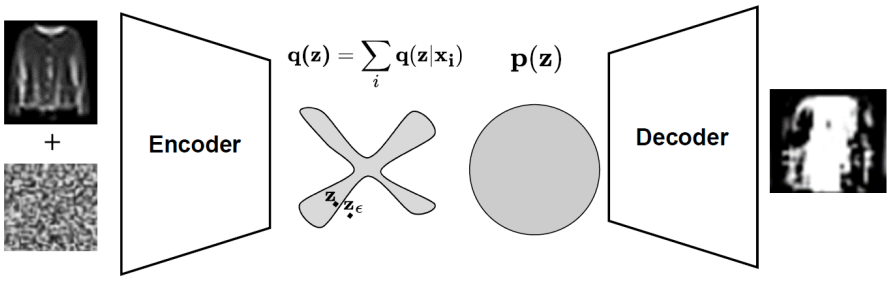


Figure 4.1: Illustration that adversarial examples find non-smooth change in the latent encodings. A small perturbation in the input sample exploits a direction that maximally changes latent encoding by moving from high density to low/zero density region in the latent space. In this work, we show an optimal perturbation can be found by moving along the eigendirection of the local pullback metric tensor of a data point.

In this chapter, we propose a novel perspective on the adversarial attack problem using the manifold geometry induced by the inference network. Unlike existing approaches treating the input space as Euclidean, we utilise the stochastic pullback-metric tensor induced by the encoder map to measure the distance in the input space. We demonstrate that the distortion in the latent space leads to a directional bias appearing in the form of an anisotropic metric tensor. We show that an optimal strategy for an adversary is to move along the dominant eigendirection of the induced metric tensor. Moreover, we propose robustness scores based on the eigenspectrum of the pullback metric tensor. We hypothesise that methods aiming to reduce *posterior-prior gap* for improving robustness directly influence the induced metric tensors. To this end, we demonstrate that the proposed scores correlate with the β parameter of $\beta-$VAE, commonly used to control robustness. To the best of our knowledge, this geometric perspective on the robustness of VAEs has not been previously investigated.

$\beta-$VAE generates over-smooth samples when a high value of β is used. To address this challenge, we introduce a constraint regularisation term to the evidence lower bound objective that builds on a mixup training scheme (Zhang et al., 2018) to improve the robustness without much effect on reconstruction quality. Specifically, the regularisation loss term encourages the encoder to fill empty regions with encodings of the linear interpolation in the data space. This mechanism ensures the decoder can generate the respective interpolations, thereby avoiding the issue of an unconstrained generation. Through extensive empirical evaluation, we demonstrate that such a training scheme improves the robustness of VAEs as measured by the proposed scores.

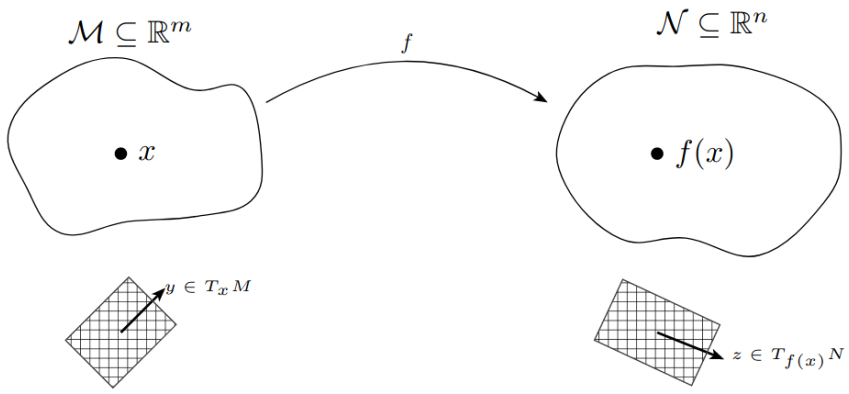


Figure 4.2: A smooth mapping f from the data manifold M to the latent manifold N induces pullback metrics on M. The Jacobian $\mathbf{J}_{f(x)} = \frac{\partial f}{\partial x}$ is a linear map from a tangent vector $y \in T_x\mathcal{M}$ to a tangent vector $z \in T_{f(x)}\mathcal{N}$ that induces a Riemannian pullback metric tensor $\mathbf{G}_x = \mathbf{J}_{f(x)}^T \mathbf{J}_{f(x)}$. The determinant of metric tensor $\mathbf{G}_x$ represents the change in infinitesimal volume element when projected to the latent space.

4.2 Background

4.2.1 $\beta-$Variational autoencoder

$\beta-$VAE (Higgins et al., 2017) is a probabilistic encoder-decoder framework that simultaneously parameterises the latent distribution and emission distribution using deep neural networks. Consider a sample $\mathbf{x} \in \mathbf{X} = \mathbb{R}^N$ drawn from unknown data distribution $p(\mathbf{x})$, VAE learns an approximate posterior distribution $\mathrm{q}_\phi(\mathbf{z}|\mathbf{x})$ over latent variables $\mathbf{z} \in \mathbf{Z} = \mathbb{R}^{d_z}$ using a stochastic encoder network, and an emission distribution $p_\theta(\mathbf{x}|\mathbf{z})$ using a stochastic decoder network. The parameters θ of an encoder network and ϕ of a decoder network are learned by maximising the evidence lower bound (ELBO),

$$\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta KL[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] \tag{4.1}$$

where KL stands for Kullback-Leibler divergence (Kullback and Leibler, 1951), and a parameter β controls the smoothness of latent distribution, setting $\beta = 1$ is equivalent to a standard VAE (Kingma and Welling, 2014).

4.2.2 Adversarial attacks on VAEs

The adversarial attacks on VAEs assume access to a pretrained encoder-decoder network. The adversary aims to exploit the capacity of VAE by finding small perturbations in an input sample that lead to a large change in its latent encoding or reconstruction. In a supervised scenario, the adversary starts with input and finds a minimal change that can match the reconstruction to the known target. In an unsupervised setting, the aim is to maximise the distance in latent

codes, which will corrupt the reconstruction. Several recent developments have proposed mechanisms for designing an adversary and evaluating the robustness of existing deep generative models (Tabacof et al., 2016; Willetts et al., 2021).

This chapter takes a geometrical viewpoint of an unsupervised attack by analysing the pullback metric tensor induced by the stochastic encoder network. We first introduce the unsupervised variational attack problem and present our contributions in Section 4.3.

For an encoder neural network $f_\theta : \mathbf{X} \to \mathbf{Z}$ the unsupervised latent space attack optimises the objective (Gondim-Ribeiro et al., 2018),

$$\max_{\eta:||\eta||_2 \leq \eta_0} \quad d(f_\theta(\mathbf{x}), f_\theta(\mathbf{x}+\eta)). \tag{4.2}$$

where η_0 is a small constant that decides the severity of the attack, and $d(.,.)$ is a distance function that measures the proximity in the latent space. A common approach to finding a corruption η is to use stochastic gradient methods (Sun et al., 2023; Willetts et al., 2021).

4.2.3 Latent Space Distortion

To understand the latent distortion, we use an alternative derivation of ELBO (Makhzani et al., 2016),

$$-RE[\log p_\theta(\mathbf{x}|\mathbf{z})] - CE[q_\phi(\mathbf{z})||p(\mathbf{z})] + H[q_\phi(\mathbf{z}|\mathbf{x})] \tag{4.3}$$

where RE is the reconstruction error, CE is the cross entropy, and H is the entropy. For a Gaussian encoder, the H term maximises the variance of latent encoding; the RE term forces the distribution to be peakier (low variance), and the CE term forces the approximated posterior to match with the prior distribution. Optimizing the full objective means the encoder has to match a fixed prior and simultaneously have a peaky distribution. Due to this tradeoff, the encoder fails to match the fixed prior, resulting in the *posterior-prior gap*. As a result, latent space is distorted with regions of low or zero density for which the decoder is unconstrained. For the derivation of Equation 4.3 with a more comprehensive discussion, we refer readers to (Makhzani et al., 2016; Tomczak, 2022).

The *posterior-prior gap* implies small changes in the input sample can result in significant changes in the latent encoding, leading to abrupt changes in the reconstruction. The β parameter reduces *posterior-prior gap*, but the higher value of β reduces the quality of reconstruction/generation using a decoder network. The low value of *posterior-prior gap* is desirable for learning representations robust to perturbation. Several other works argue to reduce this gap without compromising the performance of the decoder (Rezende and Viola, 2018; Willetts et al., 2021). In this work, we particularly look at this gap from the lens of pullback geometry and show how it results in a high curvature region in latent space that is useful for studying its smoothness and robustness to adversarial perturbation. Figure 4.1 demonstrates an example of a vulnerability of VAEs.

4.2.4 Related Work

The distortions in the latent space of VAEs have been previously shown to result in limited generalisation capacity of VAEs (Rezende and Viola, 2018; Chen et al., 2020c). Rezende and Viola (2018) further proposed a constrained optimisation to control the model performance. In our work, we take a geometric view of such distortion and use it to investigate the robustness of VAEs.

The use of encoder Jacobian matrices has previously appeared in approximating a tangent space of a data manifold, which captures the data points' sensitivity to its latent encoding (Rifai et al., 2011c,a). However, they don't consider the stochasticity of the encoder and decoder mappings. Several recent works treat the decoder mapping of VAEs as a smooth immersion and use the pullback as an induced metric in the latent space. The computation of such metrics has been helpful in various applications such as drawing on manifold samples, latent space interpolation, clustering, motion planning and many more (Arvanitidis et al., 2018; Yang et al., 2018; Hauberg, 2018; Chen et al., 2018a; Shao et al., 2017; Arvanitidis et al., 2020; Mohammadi et al., 2022).

Chen et al. (2020c) propose to flatten the pullback metric tensor in the latent space induced by the decoder, which allows them to use standard Euclidean distance as a metric in latent space. The computation of a pullback in Chen et al. (2018a); Yang et al. (2018); Chen et al. (2020c) does not consider the contribution of the uncertainty in the decoder mapping. It is, therefore, limited in its ability to capture the topological properties of the manifold. Arvanitidis et al. (2018) and (Hauberg, 2018) proposed to consider the uncertainty of the decoder by treating the Gaussian decoder as a random projection of a deterministic manifold. This viewpoint allows them to treat the reconstruction space as a random manifold and the pullback metric tensors as stochastic, proving helpful in handling topological holes and low-density regions.

We want to remark that many existing methods focus on improving the sampling in the latent space that enhances the generation performance. Here, we show that an adversary can exploit distortions in latent space and discuss the importance of local geometry for evaluating and training robust VAEs. Previously Zhao et al. (2019), Sun et al. (2019) and Martin and Elster (2020) studied the spectrum of Fisher information (pullback from probability simplex to the input space) of a classifier to investigate the robustness of adversarial perturbations. To our knowledge, there is no such study for generative models. Also, the metric tensor constructed in our work considers the effect of uncertainty in the latent space, which shows the presence of curvature in the latent space, which is vital for understanding latent distortions.

4.3 Adversarial Attack Exploit Local Geometry

Let us consider an encoder function, denoted as f_ϕ, which can be viewed as a smooth mapping from a data manifold to a latent manifold. We can then construct the pullback metric tensor induced by f_ϕ to express the infinitesimal distance in the input space using the local metric tensor of the latent space. Thus, unlike the existing methods that rely on Euclidean distance in the input space, we use the pullback metric induced by the inference network to measure the infinitesimal distance. Figure 4.2 illustrates this concept using the Jacobian of the mapping f_ϕ to estimate the local metric tensor.

In our work, we begin by formulating the adversarial optimisation problem in terms of the pullback metric tensor. Next, we show that the adversary can exploit the directional bias of a metric tensor to design an optimal attack.

Remark 1. *The infinitesimal distance between the representation of any data point* $\mathbf{x}$ *and its perturbation* $\mathbf{x}_\eta = \mathbf{x}+\eta$ *under parametric encoder* f_θ *for* η *small in* l_2 *is approximated as,*
i) $d(f_\theta(\mathbf{x}), f_\theta(\mathbf{x}+\eta)) = \eta^T \mathbf{G}_\mathbf{x} \eta$, *where* $\mathbf{G}_\mathbf{x} = \mathbf{J}^T_{f_\theta(\mathbf{x})} \mathbf{J}_{f_\theta(\mathbf{x})}$ *for locally flat latent manifold,*
ii) $d(f_\theta(\mathbf{x}), f_\theta(\mathbf{x}+\eta)) = \eta^T \mathbf{G}_\mathbf{x} \eta$, *where* $\mathbf{G}_\mathbf{x} = \mathbf{J}^T_{f_\theta(\mathbf{x})} \mathbf{G}_\mathbf{z} \mathbf{J}_{f_\theta(\mathbf{x})}$ *for a latent Riemannian manifold equipped with metric tensor* $\mathbf{G}_\mathbf{z}$. *Here* $\mathbf{G}_\mathbf{z} = \mathbf{J}^T_{g_\phi(\mathbf{z})} \mathbf{J}_{g_\phi(\mathbf{z})}$ *is a pullback under the parametric decoder mapping* g_ϕ.

where $\mathbf{J}_{f_\theta(\mathbf{x})} \in \mathbb{R}^{d_z \times N}$ is a Jacobian matrix, d_z is the dimensionality of $\mathbf{Z}$ and N is the dimensionality of $\mathbf{X}$. The matrix $\mathbf{G}_\mathbf{x}$ is a symmetric, positive definite matrix known as a pullback metric tensor under the mapping f_θ. We can use it to measure the local inner product for every $\mathbf{x}$ in the input space $\mathbf{x} \in \mathbf{X}$.

4.3.1 Stochastic Pullback Metric Tensor

In prior work,Eklund and Hauberg (2019); Arvanitidis et al. (2018) proposed a concept of an expected pullback metric tensor for measuring distances in the latent space of deep generative models. Building upon a similar idea, we construct the pullback of an encoder mapping to evaluate the robustness of representations. Here, we introduce the technical details in the context of this work. For a reference to definitions and introduction to the Riemannian manifold, we refer readers to background Chapter 2.1.1.

Let us consider a stochastic encoder mapping, denoted as f_θ, which can be expressed as a combination of mean $\mu_\theta(\mathbf{x})$ and standard deviation $\sigma_\theta(\mathbf{x})$ parameterisations: $f_\theta(\mathbf{x}) = \mu_\theta(\mathbf{x}) + \varepsilon \odot \sigma_\theta(\mathbf{x})$, where $\varepsilon \sim \mathcal{N}(0, \mathbf{I}_d)$, then a Jacobian of f_θ with respect to input $\mathbf{x}$ can be expressed as $\mathbf{J}_{f_\theta} = \mathbf{J}_{\mu(\mathbf{x})} + \varepsilon \odot \mathbf{J}_{\sigma(\mathbf{x})}$. We then define the pullback matrix $\mathbf{G}_\mathbf{x}$ as,

$$\begin{aligned}\mathbf{G}_\mathbf{x} &= (\mathbf{J}_{\mu(\mathbf{x})} + \varepsilon \odot \mathbf{J}_{\sigma(\mathbf{x})})^T (\mathbf{J}_{\mu(\mathbf{x})} + \varepsilon \odot \mathbf{J}_{\sigma(\mathbf{x})}) \\ &= \mathbf{J}^T_{\mu(\mathbf{x})} \mathbf{J}_{\mu(\mathbf{x})} + \mathbf{J}^T_{\mu(\mathbf{x})} \varepsilon \mathbf{J}_{\mu(\mathbf{x})} + \mathbf{J}^T_{\sigma(\mathbf{x})} \varepsilon \mathbf{J}_{\mu(\mathbf{x})} + \mathbf{J}^T_{\sigma(\mathbf{x})} \varepsilon^2 \mathbf{J}_{\sigma(\mathbf{x})}.\end{aligned}$$

We can view the latent space as a random projection of a deterministic manifold and a metric tensor as a stochastic matrix. Under the assumption the sample paths from the stochastic encoder mapping f_θ are smooth, we can estimate the metric as an expectation $\hat{\mathbf{G}}_\mathbf{x} = \mathbb{E}_{\varepsilon\sim p(\varepsilon)}[\mathbf{G}_\mathbf{x}]$, where ε follows the Gaussian distribution with zero mean and unit covariance the $\mathbb{E}[\varepsilon] = 0$ and $\mathbb{E}[\varepsilon^2] = 1$. With this, the final expected metric tensor is expressed as,

$$\hat{\mathbf{G}}_\mathbf{x} = \mathbf{J}^T_{\mu(\mathbf{x})}\mathbf{J}_{\mu(\mathbf{x})} + \mathbf{J}^T_{\sigma(\mathbf{x})}\mathbf{J}_{\sigma(\mathbf{x})} \tag{4.4}$$

Theorem 4.3.1. *Given a stochastic encoder mapping f_θ, for an arbitrary data point $\mathbf{x} \in \mathbf{X}$ the adversarial perturbation $\mathbf{x}_\eta$ under l_2 norm is optimal when moving along the eigendirection of a stochastic pullback metric tensor induced by f_θ at $\mathbf{x}$.*

Proof. By using the stochastic pull back metric tensor given in Equation 4.4, we reformulate the adversarial attack optimisation of Equation 4.2 as,

$$\begin{aligned} \max_\eta \quad & \eta^T \hat{\mathbf{G}}_\mathbf{x}\eta \\ & \text{subject to} \quad ||\eta||_2 = \eta_0. \end{aligned} \tag{4.5}$$

Next, to solve the problem, we combine the constraints by introducing Lagrange multiplier λ,

$$\max_\eta \quad \eta^T \hat{\mathbf{G}}_\mathbf{x}\eta + \lambda(\eta_0 - ||\eta||_2^2) \tag{4.6}$$

The closed-form solution of the above optimisation takes the form:

$$\hat{\mathbf{G}}_\mathbf{x}\eta = \lambda\eta \tag{4.7}$$

where a pair (λ, η) represents the eigenvalue and eigenvector of the stochastic pullback metric tensor. The eigenvector with the largest eigenvalue corresponds to the direction of maximal change. □

Hence, given an input $\mathbf{x}$, an adversary can design an optimal attack by stepping along the eigendirection of the expected metric tensor $\hat{\mathbf{G}}_\mathbf{x}$. The perturbed input can be obtained as $\mathbf{x}_c = \mathbf{x} + \delta\lambda\eta$, where δ is a step size. By adjusting the step size δ, the adversary can compromise the reconstruction by aiming for $||\mathbf{x} - \hat{\mathbf{x}}_c||_2 > ||\mathbf{x} - \hat{\mathbf{x}}||_2$, where $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}_c$ denotes the reconstruction of original input and its corrupted version, respectively.

4.3.2 Robustness Evaluation

To mitigate the vulnerability to the attack above, a robustness scheme should suppress the maximum eigenvalue of the pullback metric tensor and eliminate the directional bias resulting from the anisotropic distribution of eigenvalues. To quantify these effects, we introduce the following two scores,

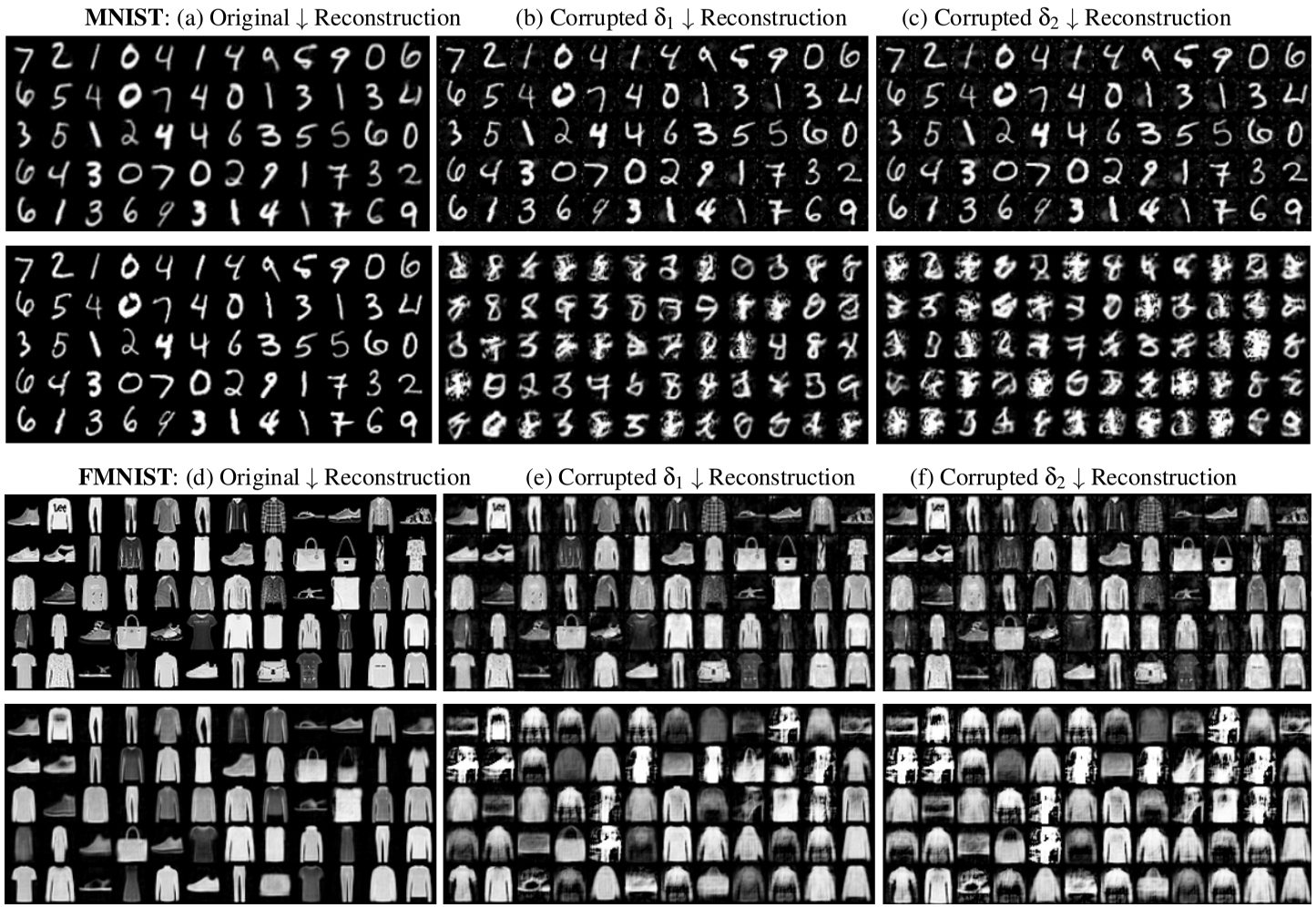


Figure 4.3: Illustration of adversarial attack along the dominant eigenvector of a stochastic pullback metric tensor. The first two rows are the results of MNIST data, and the bottom two are on the FashionMNIST dataset. We evaluate the reconstruction for original images and its two corrupted versions with different step sizes $\delta_1 = 0.5233$ and $\delta_2 = 0.7443$. Moving along eigendirection doesn't affect the input image but significantly changes its reconstruction.

(a) $\beta = 0.01$ (b) $\beta = 1.0$ (c) $\beta = 7.5$

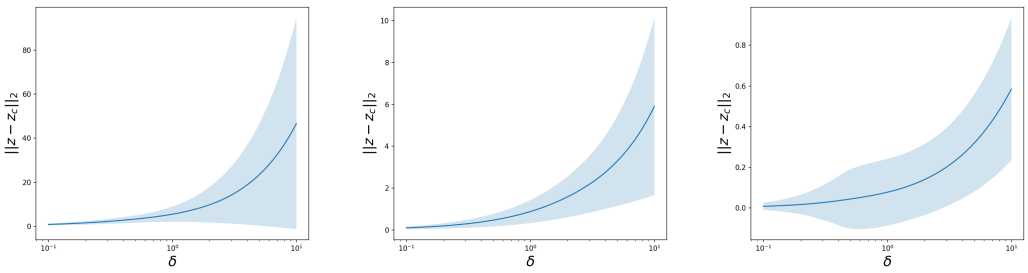


Figure 4.4: The plot shows the change in the latent encoding of $\beta-$VAE (in terms of Euclidean distance) for different values of β when moving along the dominant eigendirection of a pullback metric tensor $\hat{\mathbf{G}}_{\mathbf{x}}$ with different step size δ. We can see for small β, the changes are of much higher magnitude compared to larger β, demonstrating that increasing the β makes the latent space more smooth.

(a) Original $\downarrow$ Reconstruction (b) Corrupted $\delta_1 \downarrow$ Reconstruction (c) Corrupted $\delta_2 \downarrow$ Reconstruction

Figure 4.5: Illustration of adversarial attack along the dominant eigenvector of a stochastic pullback metric tensor on CelebA dataset. We evaluate the reconstruction for original images and its two corrupted versions with different step sizes $\delta_1 = 0.5233$ and $\delta_2 = 0.7443$.

Spectral Radius for a matrix $\mathbf{G}$ is defined as,

$$\rho(\mathbf{G}) = \max\{|\lambda|, \lambda \text{ is an eigenvalue of } \mathbf{G}\}. \tag{4.8}$$

A robust model is expected to have a smaller spectral radius.

Von Neumann Entropy (Bengtsson et al., 2008) $\mathcal{S}(\mathbf{G})$ of a metric tensor $\mathbf{G}$ is given by the Shannon entropy of its eigenvalues,

$$\mathcal{S}(\mathbf{G}) = -\sum_k \lambda_k \log \lambda_k. \tag{4.9}$$

A higher value would imply the metric tensor is anisotropic, resulting in a directional bias. Thus, a robust model will have a low value of $\mathcal{S}(\mathbf{G})$.

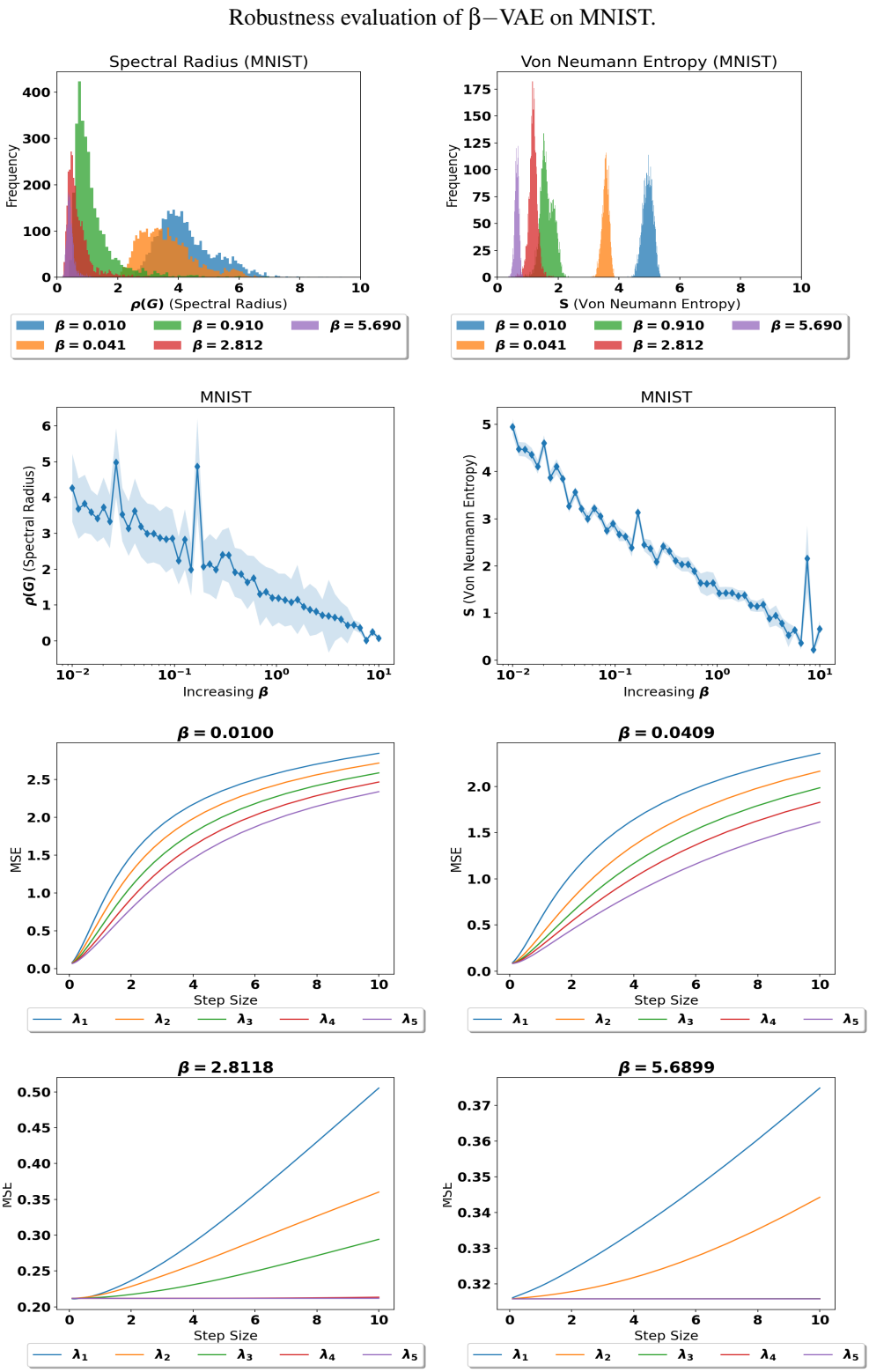


Figure 4.6: In the first row, we report the histogram of spectral radius and Von Neumann entropy (on test samples) for different values of β in β-VAE. In the second row, we report the average of two scores across test samples for an increasing value of β. We observe that increasing the value of β suppresses the metric tensor's maximum eigenvalue, and the eigenspectrum distribution gets more isotropic. In the third and fourth rows, we corrupt the test images along the top five eigendirections (denoted by $\lambda_1, \lambda_2, \lambda_3, \lambda_4,$ and λ_5) with an increasing step size for different values of β. The plots describe the average MSE across test samples. We observe that the average step size increases for a higher value of β. Increasing the value of β reduces the *posterior-prior gap*, minimising distortion in the latent space.

(b) Robustness evaluation of β−VAE on FashionMNIST.

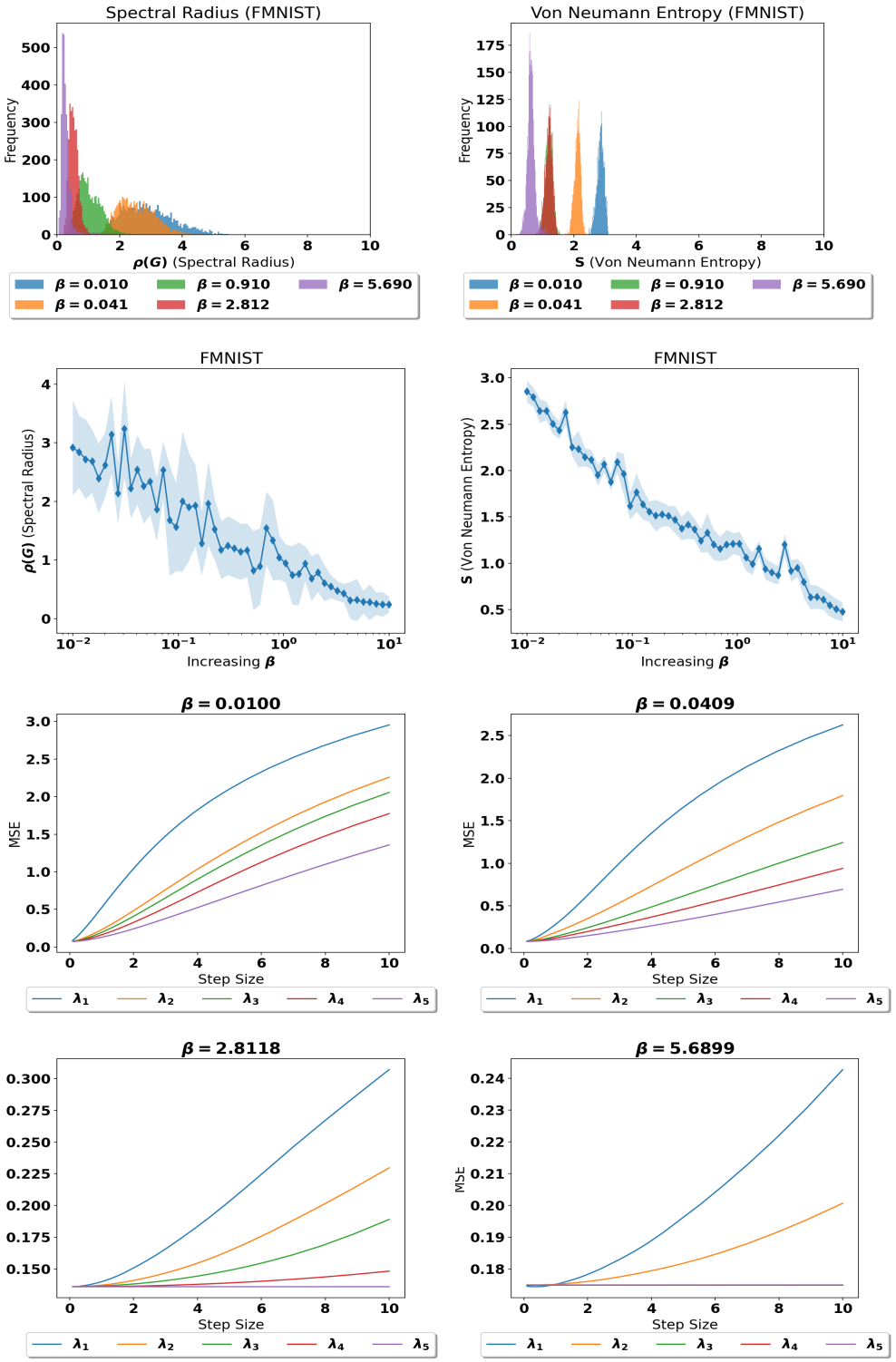


Figure 4.7: In the first row, we report the histogram of spectral radius and Von Neumann entropy (on test samples) for different values of β in β-VAE. In the second row, we report the average of two scores across test samples for an increasing value of β. We observe that increasing the value of β suppresses the metric tensor's maximum eigenvalue, and the eigenspectrum distribution gets more isotropic. In the third and fourth rows, we corrupt the test images along the top five eigendirections (denoted by $\lambda_1, \lambda_2, \lambda_3, \lambda_4,$ and λ_5) with an increasing step size for different values of β. The plots describe the average MSE across test samples. We observe that the average step size increases for a higher value of β. Increasing the value of β reduces the *posterior-prior gap*, minimising distortion in the latent space.

4.4 Results and Discussion

Here, we investigate two scenarios for the latent space geometry: (i) a locally flat space and (ii) a Riemannian manifold where a metric tensor is a pullback under the stochastic decoder mapping, as outlined in Remark (3.1). In the latter case, the stochastic pullback metric tensor in the latent space can be computed as $\mathbf{G}_{\mathbf{z}} = \mathbf{J}^T_{g_\omega(\mathbf{z})}\mathbf{J}_{g_\omega(\mathbf{z})}$ where g_θ represents a stochastic decoder.

While the main discussion of the work is centred on (i) where $\mathbf{G}_{\mathbf{z}} = \mathbf{I}$, we also analyse the case (ii) with results discussed later in Section 4.5.

In Section 4.4.1, we provide implementation details, including the training procedure. We then demonstrate the vulnerability of VAEs through empirical experiments, where we design a one-step attack by perturbing the input along the dominant eigendirection of the pullback metric tensor. Subsequently, we investigate the robustness of $\beta-$VAE formulation and propose an alternative strategy using *mixup* training. This approach aims to flatten the latent space and fill it with interpolation of latent codes from different data points, ensuring that the decoder generates valid samples.

4.4.1 Implementation details

Our implementation uses PyTorch (Paszke et al., 2019), and all models are trained on a single 11GB Nvidia RTX 2080 GPU. The code is publicly available on GitHub[1].

For the MNIST and FashionMNIST datasets, we use the same architecture across all the experiments. The encoder network is a four-layer multi-layered perceptron (MLP) with hidden units of size 256, 256, 512 and 32, respectively. The latent space follows a multivariate Gaussian distribution with mean and standard deviation parameterised by two linear mappings of size 32×32. We use the standard zero mean and unit covariance as a prior for the latent space. The decoder network is the inverse of an encoder, with hidden units of size 32, 512, 256 and 256, respectively. We use `tanh` as an activation function and batch-normalisation (Ioffe and Szegedy, 2015) before each activation.

For the CelebA dataset, we utilised a convolutional neural network (CNN) for the encoder. It consists of four convolution layers with an increasing number of filters 32, 64, 128, 256, and 512, followed by a dense layer that maps to the latent space. The latent distribution is a multivariate Gaussian, where mean and standard deviation are given by linear mappings of size 128×128. The decoder architecture is an inverse of the encoder with a dense layer utilising 1024 hidden units, followed by transpose convolutional layers with decreasing numbers of filters: 512, 256, 128, 64 and 32. The activation function `tanh` and batch-normalisation layer are applied after every convolutional layer.

All models are trained using Adam (Kingma and Ba, 2015) optimiser with a learning rate of

[1] https://github.com/MdAsifKhan/RobustnessVAE/

0.003.

4.4.2 Adversarial Attack

Figure 4.3 showcases two instances of corruption along the dominant eigenvector of $\beta = 1$ VAE on the MNIST (Lecun et al., 1998) and FashionMNIST (Xiao et al., 2017) datasets. For each dataset, the three columns in the first row are a set of original images and their corrupted version with a step size of $\delta = 0.5223$ and $\delta = 0.7443$. In the second row, we report their corresponding reconstructions. The results demonstrate that with a relatively small step size of $\delta = 0.5223$, the reconstructions significantly deviate from the original images. With a more significant step size of $\delta = 0.7443$, the deviation becomes more severe, exposing the vulnerability of the VAE. These observations empirically establish that an attacker can exploit the directional bias of the metric tensor to design a one-shot attack. Figure 4.5 further demonstrates a similar finding on the CelebA (Liu et al., 2015) dataset.

Furthermore, to analyse the latent distortions, we generate perturbations of increasing magnitude for each sample $\mathbf{x}$ by increasing step size δ along the dominant eigendirection of the pullback metric tensor. Figure 4.4 displays the distance between the latent encoding of the original input and its perturbations averaged across data samples.

4.4.3 Robustness Evaluation

This section delves into the connection between our proposed scores and the robustness of $\beta-$VAE. We also introduce an alternative approach to enhance robustness using a *mixup* training loss.

4.4.3.1 $\beta-$Variational Autoencoder

The β parameter in $\beta-$VAE controls the gap between the approximated posterior and a prior distribution. By increasing the value of β, the gap is reduced, eliminating the latent distortions which an adversary can exploit (Higgins et al., 2017; Willetts et al., 2021; Rezende and Viola, 2018). To study the effect of different values of β on the two scores, we sample 50 logarithmically spaced values of β in a range $[0.01, 10]$. For each sampled value, we trained an encoder-decoder model and computed the two scores $\rho(\hat{\mathbf{G}})$ and $\mathcal{S}(\mathbf{G})$ for every data point. Given the high computational cost of training 50 different models per dataset, we limit our experiments to MNIST and FashionMNIST datasets.

Figure 4.6 first row illustrates the histogram of the scores for four different values of β for MNIST and the first row of Figure 4.7 for the FashionMNIST. We observe that the higher value of β results in the suppression of the spectral radius and a decrease in the Von Neumann entropy, demonstrating that the local directions get isotropic. Importantly, this observation implies that it

Robustness evaluation of *mixup* on MNIST.

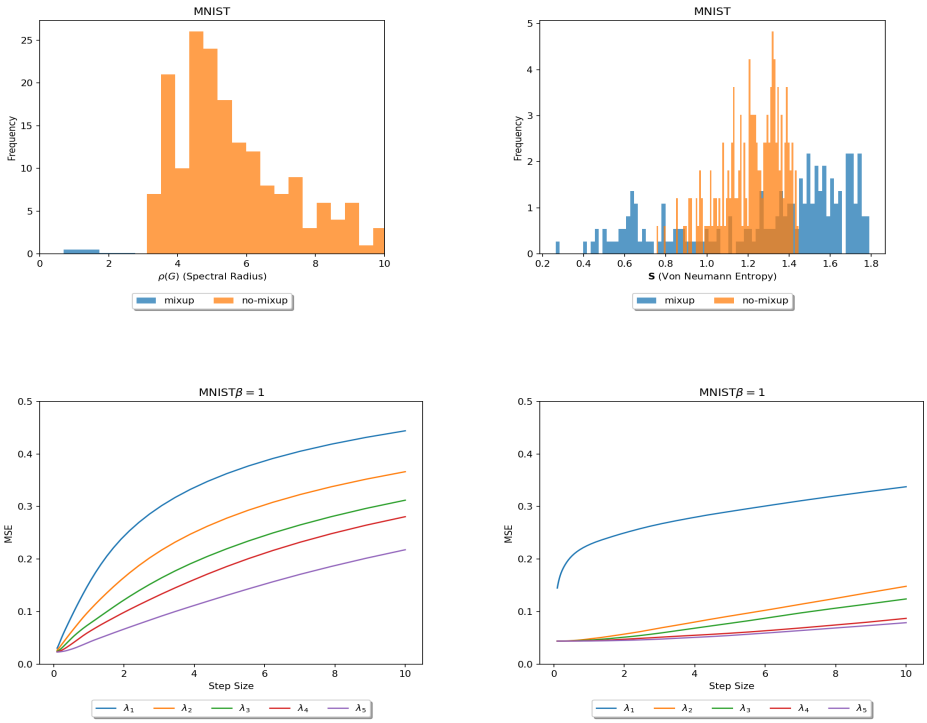


Figure 4.8: The first row reports histograms of spectral radius and Von Neumann entropy (on test samples) with and without mixup regularisation. Second row, we corrupt the test images along the top five eigendirections (denoted by $\lambda_1, \lambda_2, \lambda_3, \lambda_4,$ and λ_5) with an increasing step size for different values of β. The plots describe the average MSE across test samples. We observe that mixup suppresses the spectral radius, and the eigenspectrum distribution becomes more isotropic.

gets challenging for an adversary to attack for high values of β with η small in the norm. In the second row of Figure 4.6, we report the mean and standard deviation of the scores computed across fifty models on MNIST. Similarly, for FashionMNIST in the second Figure 4.7. The results demonstrate that by reducing the KL gap, the parameter β prevents distortion in the latent space and eliminates the directional bias exploited by an adversary.

Next, we examine the relationship between the step size δ and the strength of attacks under different β values. We measure the mean squared error (MSE) between an original image and its reconstruction for varying corruption rates along five dominant eigendirections. We generated 40 logarithmically spacing steps in the $[0.01, 10]$ range. The third and fourth row of Figure 4.6 (MNIST) and Figure 4.7 (FashionMNIST) reports the MSE versus step size averaged across all test samples for four different values of β, respectively. It can be seen for the small β values that all five directions tend to have high MSEs, while larger β values require a larger step size to achieve a noticeable change in MSE. Increasing the value of β enhances the incorporation of the

Robustness evaluation of *mixup* on FashionMNIST.

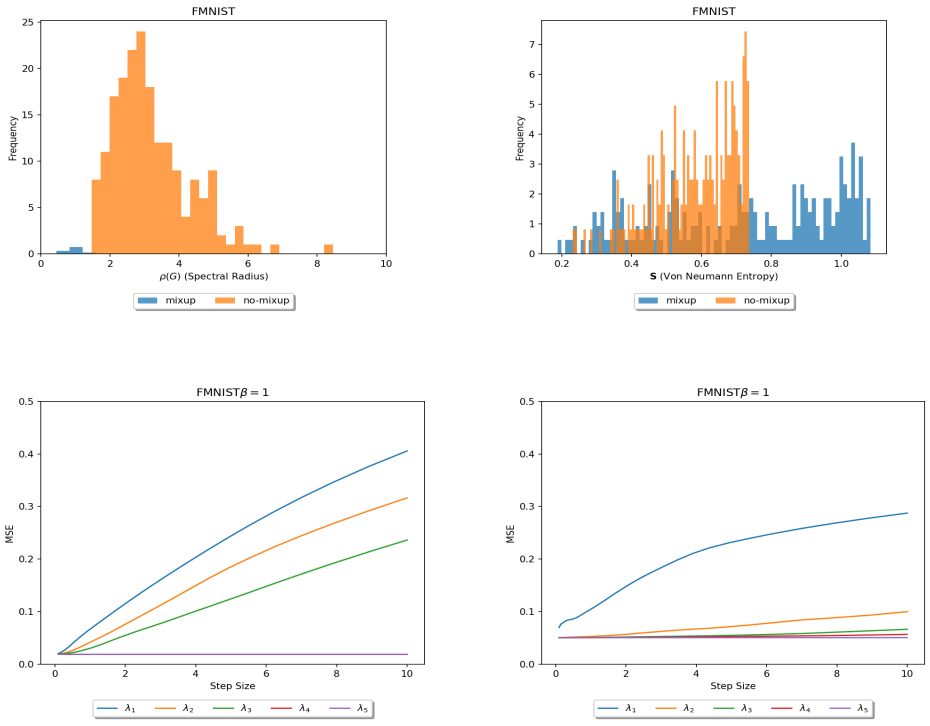


Figure 4.9: The first row reports histograms of spectral radius and Von Neumann entropy (on test samples) with and without mixup regularisation. Second row, we corrupt the test images along the top five eigendirections (denoted by $\lambda_1, \lambda_2, \lambda_3, \lambda_4,$ and λ_5) with an increasing step size for different values of β. The plots describe the average MSE across test samples. We observe that mixup suppresses the spectral radius, and the eigenspectrum distribution becomes more isotropic.

uncertainty term, indicating that the probabilistic encoder-decoder model is more robust than its deterministic counterpart. The stochastic metric tensor consists of two components arising from the variational distribution of the latent space. When β has small values, the encoder ignores the contribution of the uncertainty term and fails to align with the prior distribution. That is to say, the encoder is not restricted to a space covered by a prior distribution and can map to an arbitrary location in the latent space. As an outcome, the latent space is distorted with empty and low-density regions, as reflected by the eigendirections of the metric tensor. Conversely, as β increases, these distortions are mitigated. The encoder is penalised for not mapping to the regions in latent space covered by the prior. The second term in the metric tensor becomes more influential and effectively captures the uncertainty in the latent space, preventing the spectral radius of the metric tensor from becoming excessively large. In β-VAE, increasing the value of β bridges the gap between the posterior and the prior distributions, but it also leads to a degradation of reconstruction quality. As an alternative approach to address the issue of latent

distortion, we propose to fill the low or zero-density regions in the latent space. To this end, we utilise a *mixup* training strategy.

4.4.3.2 Latent *mixup*

The *mixup* is a powerful data augmentation technique that has been shown to improve the robustness and generalisation of classification models (Zhang et al., 2018; Lamb et al., 2019; Verma et al., 2019).

To apply *mixup*, we first generate augmented samples by performing linear interpolation between a pair of distinct data points $\mathbf{x}_i$ and $\mathbf{x}_j$. Additionally, we interpolate between their respective encodings $\mathbf{z}_i$ and $\mathbf{z}_j$ as,

$$\mathbf{z}_m = \upsilon \mathbf{z}_i + (1-\upsilon)\mathbf{z}_j, \quad \mathbf{x}_m = \upsilon \mathbf{x}_i + (1-\upsilon)\mathbf{x}_j. \tag{4.10}$$

where υ is sampled from a Beta distribution $\mathcal{B}(a,b)$ with shape parameters a and b set to 0.5. To mitigate the latent space distortions, we introduce a regularization penalty term to ELBO that fills the low-density or empty regions:

$$C = ||\mathbf{z}_m - f_\phi(\mathbf{x}_m)||_2 + ||g_\theta(\mathbf{z}_m) - \mathbf{x}_m||_2. \tag{4.11}$$

here, the first term encourages the encoder to match mixing in the input space, and the second term encourages the decoder to match mixing in the latent space. By combining the two loss terms, the encoder learns to fill the empty region of latent space, ensuring that linear interpolation in the latent space corresponds to linear interpolation in the input space. This regularisation reduces the *posterior-prior-gap* by filling the low/zero density regions in latent space by mixup augmentations. This mechanism ensures the encoder covers the space specified by the prior distribution, resulting in a smooth latent space which prevents the decoder from generating unconstrained output.

Next, we present the robustness scores of the VAE trained using a constrained loss term for increasing step sizes δ. The comparison of these scores, along with *mixup* training, is depicted in Figure 4.8 for both the MNIST and FashionMNIST datasets. Notably, *mixup* training effectively reduces the spectral radius, enhancing the robustness of the models. Moreover, Figure 4.10 further illustrates the qualitative performance of the VAE trained with the *mixup* loss, further highlighting its effectiveness.

4.5 Locally Curved Latent Space

We also conducted experiments considering the scenario where the latent space exhibits local curvature, denoted by $\mathbf{G}_\mathbf{z} = \mathbf{J}^T_{g_\phi(\mathbf{z})}\mathbf{J}_{g_\phi(\mathbf{z})}$, where g_ϕ represents a decoder network with parameters ϕ. In the case of a stochastic decoder mapping $g_\phi = \mu(\mathbf{z}) + \varepsilon \odot \sigma_\phi(\mathbf{z})$, the metric tensor $\mathbf{G}_\mathbf{z}$

FashionMNIST: (a) Original (b) Reconstruction **MNIST**: (c) Original (d) Reconstruction

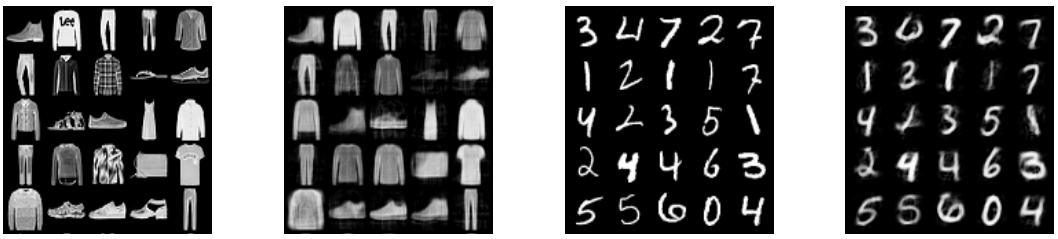


Figure 4.10: Here, we report a qualitative evaluation of training VAE with a mixup loss. We report the input samples and their respective reconstructions. On the left are the results of the FashionMNIST and the rights of MNIST.

can be further expressed as $\mathbf{G}_{\mathbf{z}} = \mathbf{J}^{T}_{\mu_\phi(\mathbf{z})}\mathbf{J}_{\mu_\phi(\mathbf{z})} + \mathbf{J}^{T}_{\sigma_\phi(\mathbf{z})}\mathbf{J}_{\sigma_\phi(\mathbf{z})}$. We utilise this expression of the pullback metric tensor in the latent space to measure infinitesimal distance, which leads to the following expression of the combined pullback metric tensor,

$$\hat{\mathbf{G}}_{\mathbf{x}} = \mathbf{J}^{T}_{\mu_\theta(\mathbf{x})}\mathbf{G}_{\mathbf{z}}\mathbf{J}_{\mu_\theta(\mathbf{x})} + \mathbf{J}^{T}_{\sigma_\theta(\mathbf{x})}\mathbf{G}_{\mathbf{z}}\mathbf{J}_{\sigma_\theta(\mathbf{x})} \tag{4.12}$$

Figures 4.12-4.14 present the empirical results obtained on the MNIST and FashionMNIST datasets. Additionally, Figure 4.11 and 4.13 provide a qualitative analysis of the robustness scores. Notably, we observe a higher value of the spectral radius when incorporating the geometry of the decoder, indicating the presence of local curvature in the latent space.

4.6 Conclusion

This chapter provided a geometrical perspective on the robustness of VAEs to adversarial attacks. Our findings demonstrate that the sensitivity of the encoder to a given input is influenced by the eigendirections of the stochastic pullback metric tensor, which an adversary can exploit to design attacks. We have proposed evaluation scores based on the spectral radius and Von Neumann's entropy of the pullback metric tensor. The large value of the scores means the latent space is non-smooth, and small changes in input can quickly move its encoding to an empty or low-density region in the latent space. Additionally, we have shown a correlation between these scores and the parameter β in β-VAE, offering insights into the smoothness introduced by increasing the β parameter improves the robustness.

However, a caveat of β-VAE is that increasing β leads to a tradeoff between representation capacity and reconstruction quality, resulting in overly smooth reconstructions for large values of β. To address this issue, we have employed a *mixup* training scheme that fills the empty regions of the latent space and improves robustness as measured by the proposed scores. It is important to note that *mixup* does not guarantee to fill all empty or low-density regions in

(a) Robustness evaluation of β−VAE on MNIST.

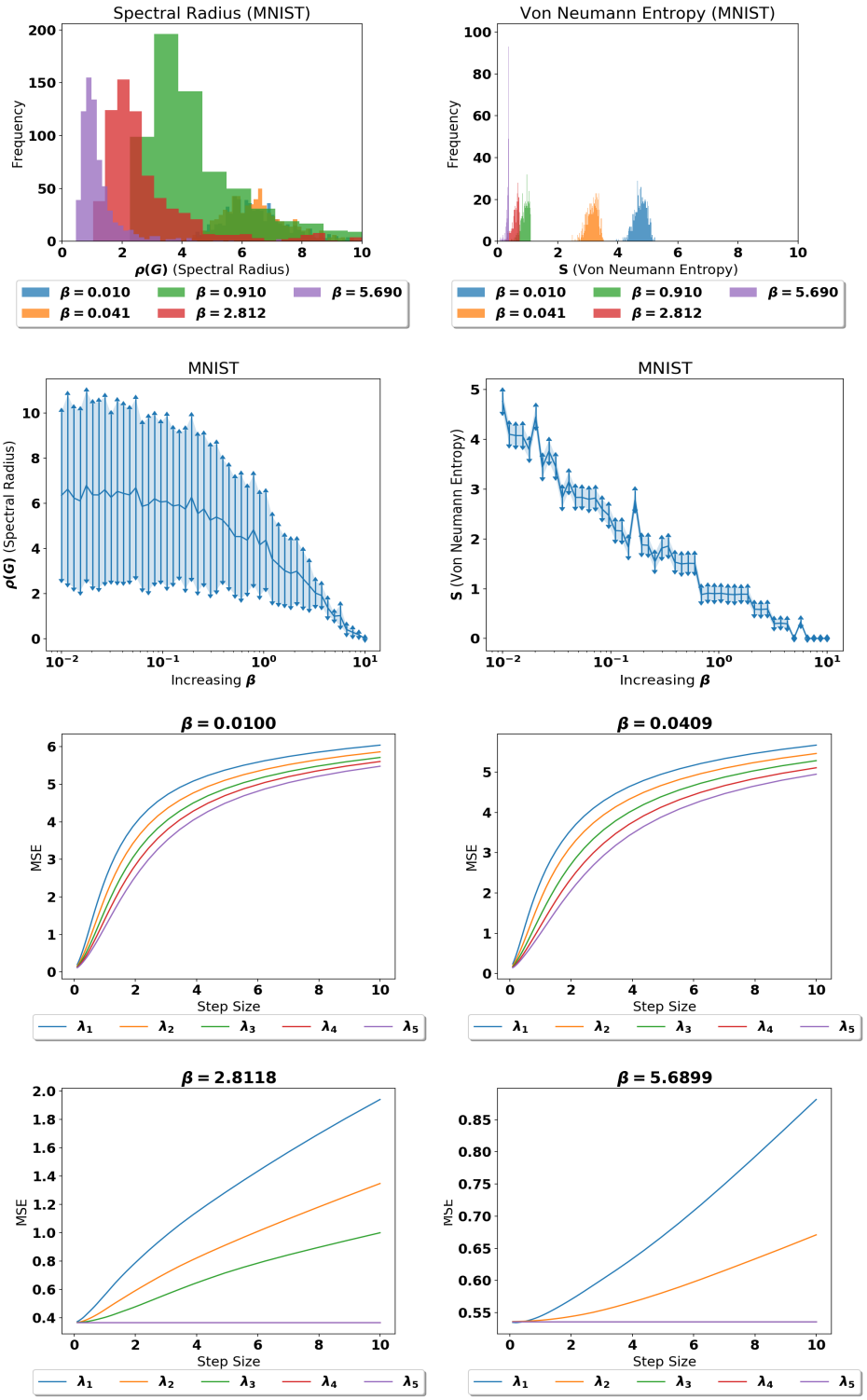


Figure 4.11: Evaluation on MNIST dataset using stochastic pullback metric tensor given by Equation 4.12. In the first row, we report the histogram of spectral radius and Von Neumann entropy (on test samples) for different values of β in β-VAE. In the second row, we report the average of two scores across test samples for an increasing value of β. Increasing the value of β suppresses the metric tensor's maximum eigenvalue, and the eigenspectrum distribution gets more isotropic. In the third and fourth rows, we corrupt the test images along the top five eigendirections (denoted by $\lambda_1, \lambda_2, \lambda_3, \lambda_4,$ and λ_5) with an increasing step size for different values of β. The plots describe the average MSE across test samples. We observe that the average step size increases for a higher value of β, the average step size increases. Increasing the value of β reduces the *posterior-prior gap*, minimising distortion in the latent space.

MNIST: (a) Original ↓ Reconstruction (b) Corrupted δ_1 ↓ Reconstruction (c) Corrupted δ_2 ↓ Reconstruction

Figure 4.12: Illustration of adversarial attack along the dominant eigenvector of a stochastic pullback metric tensor given by Equation 4.12. We evaluate the reconstruction for original images and its two corrupted versions with different step sizes $\delta_1 = 0.5233$ and $\delta_2 = 0.7443$.

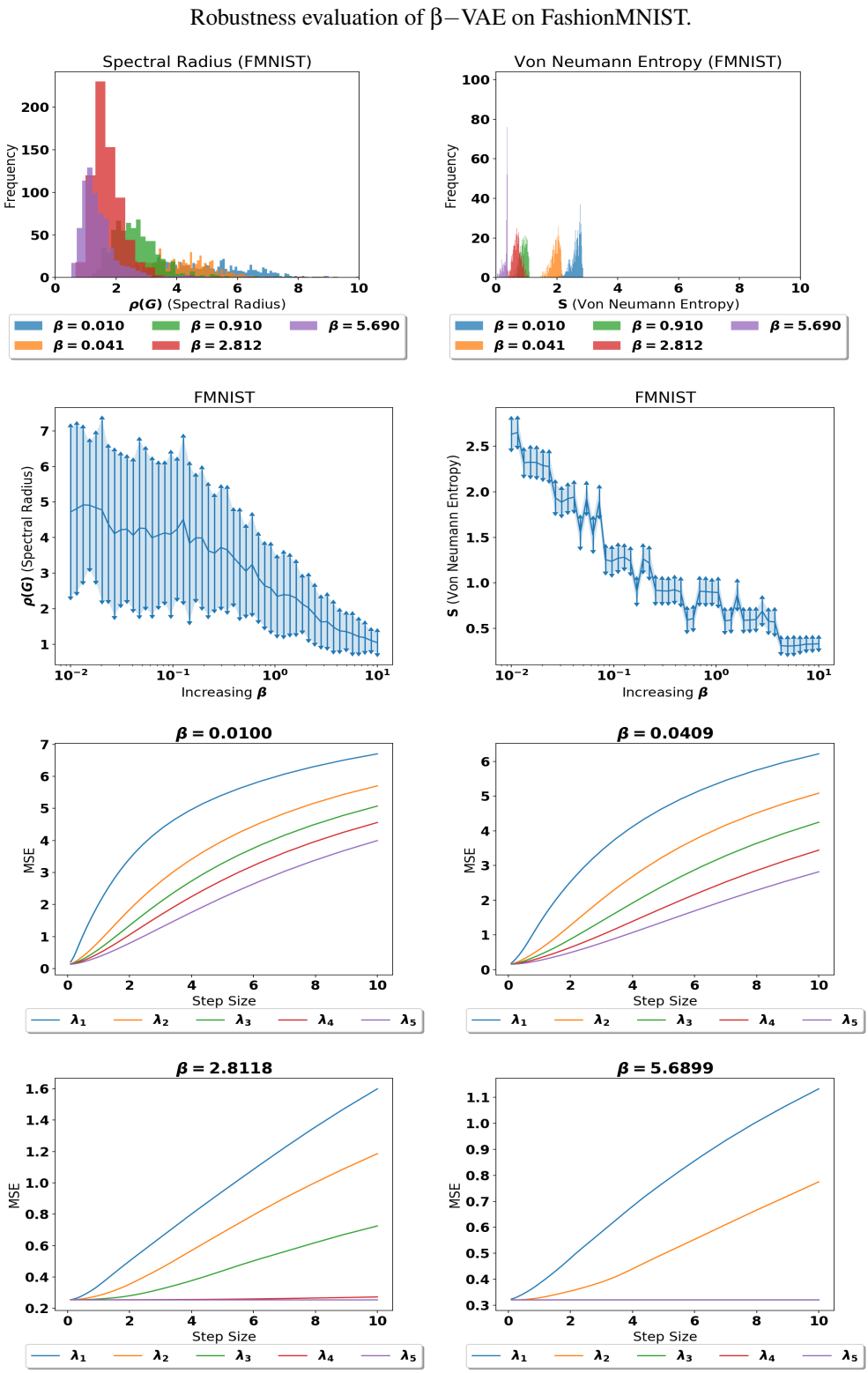


Figure 4.13: Evaluation on FashionMNIST dataset using stochastic pullback metric tensor given by Equation 4.12. In the first row, we report the histogram of spectral radius and Von Neumann entropy (on test samples) for different values of β in β-VAE. In the second row, we report the average of two scores across test samples for an increasing value of β. Increasing the value of β suppresses the metric tensor's maximum eigenvalue, and the eigenspectrum distribution gets more isotropic. In the third and fourth rows, we corrupt the test images along the top five eigendirections (denoted by $\lambda_1, \lambda_2, \lambda_3, \lambda_4,$ and λ_5) with an increasing step size for different values of β. The plots describe the average MSE across test samples. We observe that the average step size increases for a higher value of β. Increasing the value of β reduces the *posterior-prior gap*, minimising distortion in the latent space.

FMNIST: (a) Original ↓ Reconstruction (b) Corrupted δ_1 ↓ Reconstruction (c) Corrupted δ_2 ↓ Reconstruction

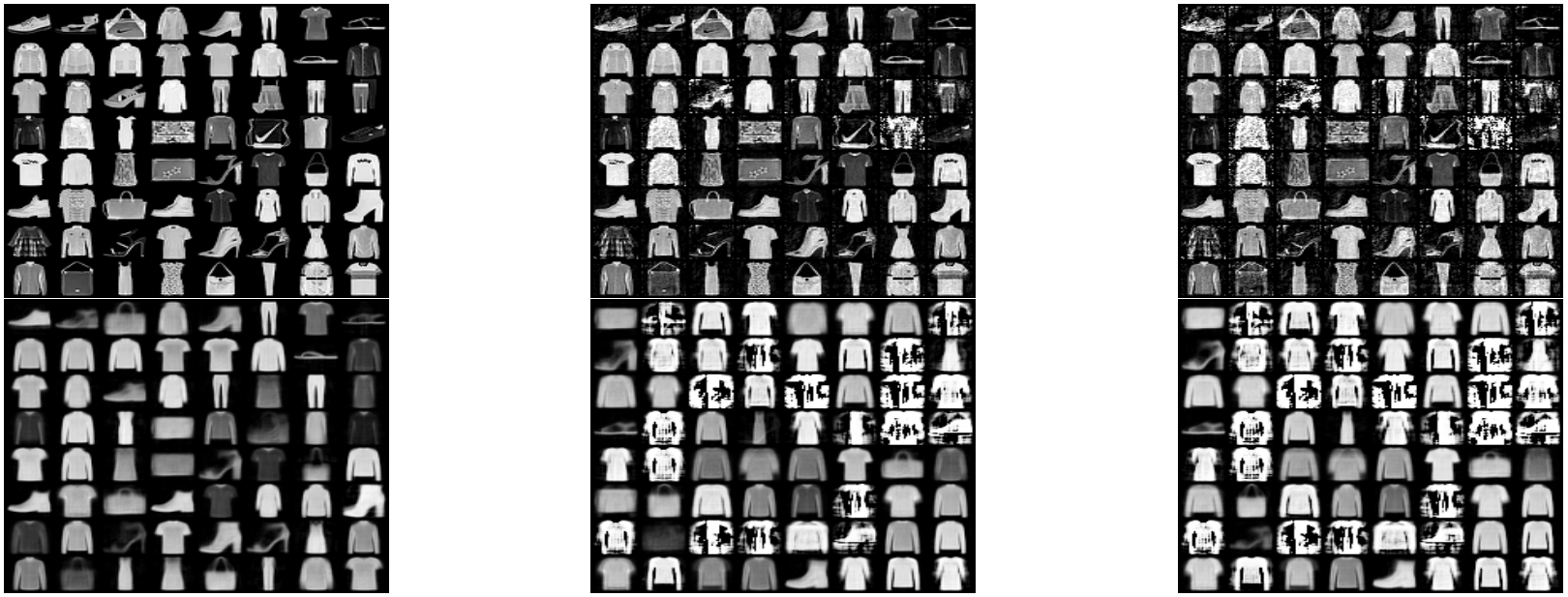


Figure 4.14: Illustration of adversarial attack along the dominant eigenvector of a stochastic pullback metric tensor given by Equation 4.12. We evaluate the reconstruction for original images and its two corrupted versions with different step sizes $\delta_1 = 0.5233$ and $\delta_2 = 0.7443$.

the latent space, as coverage depends on the regularisation hyperparameter and the size of the training dataset. Therefore, it is still possible for regions of zero or low density to exist.

A limitation of our current study is that we only consider unsupervised attacks when the target sample is unknown. Moreover, there can be different forms of attacks by replacing the l_2-norm with more general p-norms. We consider these as important future directions that we further discuss in Chapter 5.

The discussed geometrical perspective on the robustness of VAEs provides inspiration for developing practical approaches to detect and mitigate adversarial attacks, thereby improving the reliability and interpretability of machine learning models in real-world applications. Furthermore, these insights can enhance our understanding of the generalisation capabilities of VAEs, especially in analysing the behaviour of the latent space for unseen or out-of-distribution data. This work was presented at the AISTATS conference in April 2023, underscoring its significance in the field.

The failure of VAEs to match a prior distribution has led to alternative approaches such as deterministic autoencoders (Ghosh et al., 2020). The authors of the deterministic autoencoder show the stochasticity of the encoder paramount to noise injection before passing the latent code through the decoder. This training scheme eliminates the optimisation issues associated with matching the prior, and the coverage of latent space is controlled by a regularisation loss defined over latent variables. The pullback geometry is agnostic of the choice of representation framework. Our work can further be applied to deterministic counterparts of VAEs to investigate the effect of various regularisation terms on the smoothness and robustness of latent space.

So far in this thesis, we have demonstrated the importance of geometry principles in accounting for some important properties in the representations, including disentanglement, temporal coherence and robustness. In the next chapter, we look at the problem of learning representation of graph-structured data. Due to the complex nature of graph data, there are short-term, long-term, and hierarchical dependencies among nodes. A good representation of nodes should capture such relationships to perform well on downstream tasks. We will utilise diffusion geometry to introduce a data augmentation technique that will allow a representation learning framework to account for hierarchical dependencies.

Chapter 5

Representation Learning for Heterophilic Graphs

The introductory Chapter 1 discussed the importance of geometry for incorporating properties in deep representation learning that will lead to good representations. So far, we have explored the role of geometry in a sequential variational autoencoder (VAE) for modelling image sequences; this facilitated the learning of disentangled factors of variations. We then saw how it could be used as a lens for inspecting the smoothness of the latent space of VAEs, thereby allowing us to turn them into more robust representation learners.

While the previous chapters have primarily focused on image data where an underlying domain is a regular grid graph, the emergence of data on general graph domains has led to deep-learning applications in these scenarios. In the context of graph representation learning, the objective is to embed the nodes of a graph into an Euclidean space, where the embeddings capture the hierarchical dependencies within the graph. Additionally, some cases involve the availability of attribute vectors for each node. For instance, in a social network graph, nodes represent individuals, edges signify connections or friendships, and attributes can denote individual activities. By learning node representations, downstream tasks such as link prediction (for friend recommendations), node classification (for grouping based on demographics) or graph classification (for predicting the properties of molecules represented as graphs) can be effectively performed. The growing field of deep learning approaches for solving problems on graph-structured data (Bronstein et al., 2017) offers diverse applications ranging from social networks to discovering novel gene-disease associations.

A key challenge to deep learning in data on a graph domain is the availability of limited labelled data. This has sparked interest in the self-supervised learning (SSL) paradigm due to its success with image data. The intuition behind SSL is to design a "pretext" task that could encourage necessary invariances in representations. For instance, in image data, a task can be to predict the correct order of patches in an image or rotate an image along an axis and

predict an angle of rotation. A widely popular approach within SSL is known as contrastive learning, which first constructs a set of augmentations for a given data point and then learns an embedding invariant to such transformations. SSL has emerged as a popular paradigm for graph representation learning, leveraging the data's inherent structure to generate supervisory signals for training. For example, dropping nodes or edge attributes in a graph and a pretext task can be predicting missing nodes or inferring edge attributes. This learning mechanism allows SSL to reason about the graph structure and learn meaningful representations, which can be used for a broad range of downstream applications, including node clustering or graph classification.

Despite the remarkable success of SSL for graphs, they still exhibit limitations in capturing the multi-scale structural dependencies relevant for *heterophilic* graphs. Heterophily is a graph property where nodes are labelled based on their local structural correspondence, meaning nodes that share similar connectivity patterns may have the same label even if they are not immediate neighbours. For instance, in a social network, let us say our task is to detect fraudsters or scammers. Fraudsters may not directly connect with one another; however, they share similar behaviour, making it possible for two distant nodes to be labelled as "fraud". Learning representations that encode this multilevel dependence information is crucial for addressing challenges in graphs where node labels are determined by their intricate connectivity patterns.

In many real-world scenarios, such as social networks with users of different demographics or biological networks with different types of proteins, heterophilic graphs are prevalent. Considering the multi-scale structure of heterophilic graph data in the learning process is critical to overcoming the above limitations. This chapter introduces a new training scheme for SSL that ensures the learned representations encode for hierarchical dependencies in the graph supported by empirical results on a range of heterophilic graph tasks. Specifically, we propose an algorithm for generating augmentations of a graph using a diffusion process and use these augmentations with a multi-view contrasting setup for learning the representation of nodes.

Diffusion processes operate by iteratively propagating information across the graph, simulating the diffusion of information through the edges. At each iteration, the information from neighbouring nodes is aggregated and weighted by the edge weights. This process theoretically captures the graph's local and global dependencies and enables the representations to encode rich relational information.

In this chapter, we combine graph diffusion with SSL to propose a framework that encourages representation space to capture multi-scale relationships between nodes. Specifically, we suggest constructing a set of diffusion filters that capture interdependencies among nodes at varying time intervals, thereby explicitly expressing intricate patterns and interactions prevalent in heterophilic graphs. We utilise these filters as augmentations of graphs for training multi-view contrastive learning framework. The learned representations enable more accurate predictions

and better clustering on heterophilic graphs, as demonstrated on synthetic and real datasets.

We begin with an introduction in Section 5.1, providing an overview of the problem and its significance. Next, in Section 5.2, we review the existing literature to establish the context and highlight the gaps in current approaches. We then present our proposed learning framework in Section 5.3, outlining the key components and methodologies employed. Moving forward, Section 5.4 focuses on the experimental evaluation, presenting the results and engaging in a detailed discussion. Finally, in Section 5.5, we conclude the chapter by summarising the key findings.

5.1 Introduction

Graphs have emerged as prominent data structures for capturing complex relational interactions among entities in various domains, including social networks, molecular systems, and more (Wang et al., 2022). The availability of diverse graph datasets (Bojchevski and Günnemann, 2018; Zitnik and Leskovec, 2017; Rozemberczki et al., 2021) and the need to address tasks such as molecular property prediction (Wu et al., 2018), protein-protein interaction (Zitnik and Leskovec, 2017), drug-disease associations (Lin et al., 2020), and inferring a new connection between entities (Zhang and Chen, 2018), have fueled the development of machine learning (ML) approaches tailored for graph-structured data.

Traditionally, supervised (Cangea et al., 2018; Shervashidze et al., 2011; Dai et al., 2016; Ruiz et al., 2020), semi-supervised (Belkin et al., 2006; Weston et al., 2008; Kipf and Welling, 2017; Bui et al., 2018), and unsupervised methods (Kipf and Welling, 2016; Belkin and Niyogi, 2003; Cai et al., 2018) have been employed in ML for graphs for addressing various problems. Supervised approaches rely on class labels to guide representation learning, but their reliance on annotated data poses challenges due to limited availability and high labelling costs. This limitation has motivated the exploration of self-supervised and unsupervised learning methods that can leverage the inherent structure of the data to learn meaningful representations without the need for explicit labels. Such approaches have gained significant attention due to their ability to efficiently learn representations in a data-driven manner while mitigating the annotation bottleneck.

Graph representation learning methods aim to capture the geometry of graph data by embedding nodes into a lower-dimensional space. The objective is to ensure that nodes with a particular notion of similarity in the graph are mapped to nearby locations in the embedding space. These learned embeddings can be readily used for various downstream tasks such as node classification, link prediction, and clustering (Hamilton, 2020; Chen et al., 2020b; Wang et al., 2017b). However, it is crucial to incorporate appropriate assumptions during the training process to achieve better generalisation to downstream tasks. For instance, in a homophilic graph, where

nodes with similar attributes tend to be connected, the representation space should prioritise the similarity of encodings between proximal nodes compared to non-proximal ones. Conversely, in a heterophilic graph, where structurally similar nodes are usually distantly connected, it is expected the embedding of these nodes in a representation space is to nearby locations. By tailoring the learning process to the specific characteristics of the graph, we can enhance the quality and applicability of the learned representations.

The contrastive approaches in the SSL paradigm utilise an objective to encourage similar samples (positives) to be nearby in the embedding space while dissimilar samples (negatives) are pushed farther apart (Chen et al., 2017, 2020b; Jaiswal et al., 2021). The positives are the alternative views of data obtained using augmentation transformations, and the negatives are sampled using a stochastic corruption strategy. SSL has been extended in graph-structured data to capture the inherent graph properties(Velickovic et al., 2019; You et al., 2020; Hassani and Khasahmadi, 2020). Graph contrastive learning (GraphCL; You et al. 2020) introduces task-specific augmentations to generate multiple graph views, enabling the learning of view-consistent node representations. Deep graph infomax (DGI; Velickovic et al. 2019) maximises the mutual information between a node embedding and a global graph embedding obtained through a readout function. However, these methods have limitations in capturing global structural information due to their reliance on local graph networks represented by normalised adjacency matrices. To address this limitation, multi-view graph representation learning (MVGRL; (Hassani and Khasahmadi, 2020)) utilises a random walk diffusion matrix as a global structural view of the graph. However, the dominance of low-frequency components in the random walk diffusion limits its ability to express higher-order hierarchical information crucial for capturing structural equivalence (Gao et al., 2019).

The hierarchical nature of graphs comes with the higher-order relational information embedded within different levels of resolution (Coifman and Maggioni, 2006).In this chapter, we argue that capturing such a hierarchy is vital for learning good representations, especially relevant for structural equivalence tasks. We propose a mechanism that utilises information from multi-resolution diffusion filters on graphs for representation learning. We achieve this by constructing a cascade of diffusion filters on the input graph, starting with a lazy diffusion operator and constructing coarser graphs using dilation filters. Each filter represents a different resolution of the diffusion operator applied to the graph. By training an encoder to map these coarse views to a feature space using a contrastive objective, we enable the encoder to learn representations invariant to multi-resolution views. This invariance accounts for the higher-order information necessary for identifying structural equivalence. Our empirical results demonstrate that our approach outperforms other SSL methods on various synthetic and real-world heterophilic graphs. By leveraging the principles of multi-view coding, our simple yet effective method holds promise for advancing representation learning on graph-structured data.

The choice of dilated filters offers several advantages. Each coarse view acts as a band-pass filter, capturing interdependency among nodes at varying resolutions. This approach provides augmentations of the input graph that explicitly presents the encoder network with a hierarchy of spatially localised information, enabling the learning of latent space that captures long-range dependencies, thereby accounting for structural equivalence. Furthermore, diffusion filters are stable, remaining robust to perturbations and deformations in the input node features of a graph. This stability is vital to the robustness of the learned representations, ensuring their reliability under real-world applications.

5.2 Related Work

Representation learning for graphs encompasses various approaches, including supervised and unsupervised methods. Several works have built on the success of contrastive learning in image data to propose algorithms for graph-structured data. In the context of this chapter, we refer to the following works as particularly relevant contributions to representation learning for graphs.

5.2.1 Random Walk Methods

Embedding models have gained significant popularity in natural language processing, as they enable the mapping of words to dense feature representations, where semantically similar words tend to be located close to each other in the embedding space. A notable example is Word2Vec, introduced byMikolov et al. (2013), which utilises the Skip-Gram model to maximise the log probability of a word given its context window. In doing so, Word2Vec effectively learns to map words with high co-occurrence within a context window to representations that are nearby under Euclidean distance. In the context of graph data, various embedding methods have employed random walk sequences to train models inspired by Word2Vec, such as DeepWalk (Perozzi et al., 2014), node2vec(Grover and Leskovec, 2016), subgraph2vec(Narayanan et al., 2016), and many others. However, random walk-based methods have limitations in capturing long-term dependencies and often perform poorly on tasks requiring structural similarity.

To address the above challenges, Narayanan et al. (2016) proposed an algorithm that learns node embeddings, such that the embeddings of nodes with a similar local structure are more similar. However, their algorithm relies on a predefined notion of local structure and may struggle with graphs exhibiting more complex structural equivalence. Several recent approaches (Yanardag and Vishwanathan, 2015; Al-Rfou et al., 2019; Borgwardt et al., 2020) have tackled the issue of capturing structural similarity by decomposing graphs into sub-structures and employing graph kernels to measure node similarity. Nonetheless, a fundamental limitation is that determining the appropriate sub-structures requires domain knowledge, which may not be readily available for a wide range of graph data.

5.2.2 Spectral Methods

Methods in this category rely on the spectral properties of adjacency or the Laplacian matrix of the graph. The eigenvectors of the Laplacian matrix are interpreted as frequencies in the Fourier space encoding information about graph properties, such as the size of cuts. Classical methods like Laplacian eigenmaps (Belkin and Niyogi, 2003) use the eigenvectors corresponding to top-k eigenvalues of Laplacian as feature representations. However, top-k eigenvalues of Laplacian correspond to low-frequency components; they fail to capture higher frequencies important for long-term dependencies.

Alternatively, methods using the diffusion process are based on the heat kernel of a graph, which is the solution of the heat equation associated with the Laplacian operator of the graph. The heat kernel can be interpreted as a similarity matrix where each entry is an expected distance value across all paths between pairs of nodes. The embeddings of the nodes are constructed by taking top-k eigenfunctions of the heat kernel matrix. Other approaches like Tsitsulin et al. (2018) stack the trace of the diffusion matrix at different scales to get its feature representation, which in turn is used for supervised classification. Still, under a large diffusion time, the contributions from higher frequency are suppressed, leading to ineffective representations for expressing higher-order structural information. Several other approaches directly utilise spectral methods to learn filters that capture structural relationships, such as graph convolution (GCN) (Kipf and Welling, 2017) ChebNet (Defferrard et al., 2016), CayleyNets (Levie et al., 2019), and so on. These methods learn suitable filters based on the spectral properties of the graph to better capture structural relationships.

While Laplacian-based methods and diffusion approaches have their merits, they are still limited in capturing multi-scale information crucial for complex structural relationships.

5.2.3 Contrastive Methods

Contrastive approaches have emerged as a popular choice for representation learning. The idea is to learn representations that share information expressed in different data augmentations, a.k.a views. The methods hold application to a comprehensive set of problems encountered in computer-vision (Chuang et al., 2020; Chen et al., 2020d; He et al., 2020), reinforcement learning (Laskin et al., 2020), etc. The Skip-Gram training of DeepWalk is also a form of contrastive learning; it utilises positive and negative pairs of nodes based on a distance in a random walk and trains a discriminator to increase the score of positive samples and decrease the scores of negatives. More recent contrastive methods focus on different forms of discriminator function as well as ways for generating good negative samples (Chen et al., 2017, 2020d). Velickovic et al. (2019) introduced deep graph infomax (DGI) for node representation learning that contrasts the local-global representations from the negative pairs. Another work,

Infograph (Sun et al., 2020), used infomax to learn graph-level representations.

A recent article closely related to our work is multi-view graph representation learning (MVGRL; (Hassani and Khasahmadi, 2020)). MVGRL utilised multiple views of graphs obtained by varying time scales of the diffusion process and trained a neural network with the DGI objective. However, as noted by the authors, the results do not benefit from additional views. Our work differs in a critical aspect; we do not use a single global diffusion view; instead, we construct a cascade of filters that provide multi-resolution views of the structural information.

5.2.4 Autoencoder Methods

The encoder-decoder model and its probabilistic counterpart are widely used for representation learning. Kipf and Welling (2016) extended variational autoencoder (VAE) (Kingma and Welling, 2014) to graph-structured data by employing layers of GCN as an encoder. They used a Gaussian prior on node representations and a dot product decoder to reconstruct the graph structure. Subsequent works proposed a more expressive GNN architecture for the encoder (Hamilton et al., 2017a; Grover et al., 2019; Pan et al., 2018; Yang et al., 2021) or the decoder (Wang et al., 2017a; Park et al., 2019; Shi et al., 2020). However, most of the above methods focused only on local neighbourhood structures during reconstruction, resulting in poor performance on structural equivalence tasks. To address this limitation Tang et al. (2022) proposed an approach called neighbourhood Wasserstein reconstruction (NWR). NWR introduces an objective function that increases the capacity of representation in capturing both proximity and structure information. Instead of using a probabilistic encoder, NWR training utilises a deterministic one. Our approach is not directly comparable to NWR as our method belongs to a self-supervised class that does not rely on a decoder neural network.

5.2.5 Diffusion Wavelets

Diffusion wavelets, introduced by Coifman and Maggioni (2006), offer a powerful yet simple tool for multi-resolution analysis of signals. The central idea is to start with a lazy diffusion operator and examine the difference in heat diffusion across varying timescales. Building on the work of Mallat (2012), Gama et al. (2019) utilised diffusion wavelets to construct multi-resolution filter banks on graphs. They demonstrated the stability of such filters in the face of signal deformation and perturbation, enabling the training of deep and stable graph neural networks. More recently, Gao et al. (2019) showcased that diffusion wavelets could be used to learn a universal representation of graphs. While Tong et al. (2021) proposed a method to learn the scale parameter of filters using stochastic softmax tricks (Paulus et al., 2020). In contrast to these approaches, our work employs multi-resolution filters as alternate graph views that can learn resolution-invariant representations useful for structural equivalence in graphs when combined with a contrastive objective.

Another wavelet-based method, GraphWave (Donnat et al., 2018), utilises wavelet diffusion to learn multi-resolution structural embedding. Our work differs from GraphWave in two aspects. Firstly, while GraphWave solely operates on structural information, our approach can leverage both structure and node attributes. Secondly, GraphWave learns embeddings by characterising wavelet filters in the spectral domain using the characteristic function of the Laplacian eigenvalues. In contrast, we construct dyadic scale views of the graph and utilise them as augmentations for the feature representation network. These augmentations encourage the encoder to learn node embeddings that capture hierarchical information shared across multi-resolution graphs.

5.2.5.1 Wavelet Filter Bank

Let us consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $N = |\mathcal{V}|$ nodes, $|\mathcal{E}|$ edges, and $\mathbf{X} \in \mathbb{R}^{N \times d_i}$ as the node attribute matrix. We define $\mathbf{A} \in \mathbb{R}^{N \times N}$ as the weighted adjacency matrix, $\mathbf{D} = \text{diag}(d_1, \ldots, d_N)$ as the degree matrix where each diagonal entry is the degree of the respective node, $\tilde{\mathbf{A}}_1 = \mathbf{A}\mathbf{D}^{-1}$ as the column-normalised adjacency matrix, and $\mathbf{T} = \alpha \mathbf{I}_n + (1-\alpha)\tilde{\mathbf{A}}_1$ as the normalised lazy random walk diffusion matrix with a restart probability $\alpha \in [0.5, 1]$.

We utilise the lazy diffusion operator to introduce multi-resolution filters on the graph, following the approach of Coifman and Maggioni (2006) and Gama et al. (2019). To put it formally,

$$\mathbf{\Phi}_1 = \mathbf{I} - \mathbf{T}, \quad \mathbf{\Phi}_j = \mathbf{T}^{2^{j-1}}(\mathbf{I} - \mathbf{T}^{2^{j-1}}), \quad j > 0. \tag{5.1}$$

Here, $\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_{K-1}$ represents a bank of filters on a dyadic scale where each filter emphasises the higher-order connectivity information of a graph at different granularity. The restart probability α controls the influence of the local structure and the rate of diffusion propagation in the filters. This sequence of filters forms a diffusion wavelet filter bank on the graph, which we later utilise for training the encoder neural network.

5.3 Multi-Resolution Graph Contrastive Learning

This section presents our approach to constructing graph augmentations using a diffusion filter bank defined in the previous section.

Graph tasks such as node classification assume that nodes with the same labels have similar features. This assumption serves as an inductive bias for training an encoder that learns a lower-dimensional latent space, where similar nodes are mapped to nearby locations. However, the notion of similarity can vary across different graphs. In homophilic graphs, nodes in close proximity tend to have similar labels, whereas in heterophilic graphs, the similarity is determined

by the structural equivalence of the nodes. Our objective is to learn an encoder that captures various levels of structural similarity between pairs of nodes in a self-supervised manner.

Many existing methods attempt to capture structural information by relying on predefined notions of similarity, which may fall short when dealing with complex structural equivalence (Yanardag and Vishwanathan, 2015; Ribeiro et al., 2017). Diffusion methods quantify similarity at different time scales using the notion of diffusion distance. This metric measures the heat diffused between nodes i and j within a given time t and vice versa. However, as t increases, the diffusion process is dominated by low-frequency components, thereby neglecting higher-order structural information. Alternatively, diffusion wavelet filters (Coifman and Maggioni, 2006) capture a band-pass response of a signal on a graph; that is, each filter reflects the diffusion distance in a time interval. The filters are constructed on a dyadic scale ordered as fine to coarse-grained. Thus explicitly highlighting the higher-order structural information encoded in a graph.

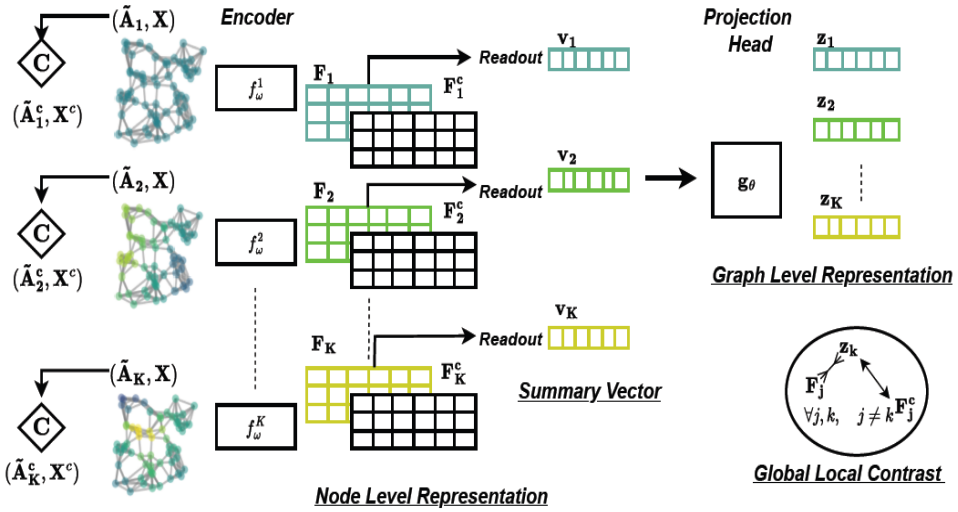


Figure 5.1: An illustration of our multi-resolution self-supervised learning pipeline. We first use diffusion wavelets to construct multi-resolution views $\{G_1, \cdots, G_K\}$ and combine them with attribute information to learn node level representation $\mathbf{F}_k$ using an encoder network (we use GCN with two layers). For each resolution, we also generate a corrupted view using a stochastic corruption function $\mathbf{C}$ and map them to node-level representation $\mathbf{F}_k^c$. Next, for each of the k views, we pass the node representation matrix to a *Readout* operation followed by an MLP to get the graph level representation $\mathbf{z}_k$. A contrastive loss is trained to maximise consistency between node and graph representation of intra-views. The final node level representations are obtained by pooling across views as $\mathbf{F} = \frac{\sum_k \mathbf{F}_k}{K}$

.

Our work leverages diffusion wavelet filters to construct augmentations of the graph and utilise it with a multi-view contrastive objective for learning resolution invariant representations.

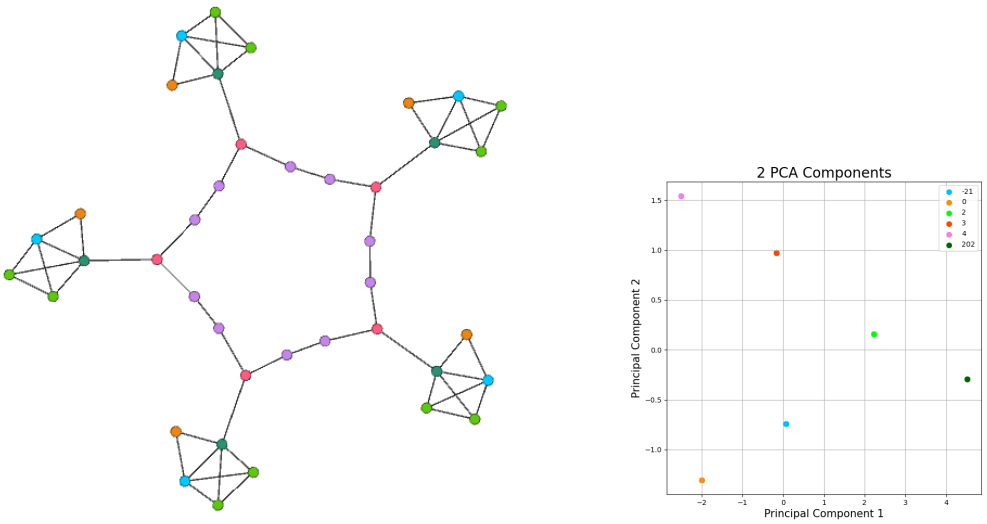


Figure 5.2: An illustration of structural equivalence. The left panel depicts a toy graph, where nodes are assigned different colours to denote their roles within the graph structure. On the right panel, we present the first two principal components of the node embeddings obtained through our multi-resolution contrastive approach. These principal components are further colour-coded based on the node labels. The numbers next to the legend in the plot are random labels associated with each type of label depicted by colour in a graph. Notably, the embeddings effectively capture the notion of structural similarity, as nodes with similar structures are assigned similar representations, as indicated by the consistent colouring of nodes with similar roles in the graph. This figure is a toy demonstration of a graph with a heterophilic property. Later, in Table 5.2, we compare different state-of-the-art methods.

Our experimental results on structural equivalence tasks demonstrate that the learned encodings can capture a varying level of hierarchical relationship within graphs. Figure 5.2 illustrates our approach using a synthetic graph, where nodes are coloured based on their structural roles, and the right side depicts the first two principal components of the learned representations. We observe that structurally equivalent nodes are grouped together in the representation space, showcasing the benefits of employing multi-resolution views. The overall pipeline of our approach is presented in Figure 5.1. Next, we introduce diffusion wavelets and provide details of the training setup employed in this work.

5.3.1 Multi-resolution graph augmentations

We utilise the filter bank $\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_{K-1}$ defined in Section 5.2.5.1 to construct a set of graph augmentations that we utilise for representation learning. To begin with, we sparsify the filters using a threshold ε that sets the entries below ε to zero, resulting in $\mathbf{A}_k = \mathbf{\Phi}_k[\mathbf{\Phi}_k < \varepsilon]$, where

$[\mathbf{\Phi}_k < \varepsilon]$ represents element-wise comparison. We then normalise $\mathbf{A}_k$ to obtain $\tilde{\mathbf{A}}_k$. The ε and the restart probability α are treated as hyperparameters.

5.3.2 Feature Representation Network

Given a set $\tilde{\mathbf{A}} = \{\tilde{\mathbf{A}}_1, \cdots, \tilde{\mathbf{A}}_K\}$ of K augmentation views of the graph. We combine each view with the attribute matrix and utilise an encoder network $f_\omega^k : \mathbb{R}^{N\times d_i} \times \mathbb{R}^{N\times N} \to \mathbb{R}^d$ to map them to their respective node feature representation matrix $\mathbf{F}_k \in \mathbb{R}^{N\times d}$. The individual node representations from each view are then aggregated using a readout operation to obtain K summary representations. These summary representations are further projected to obtain a graph level representation $\mathbf{z}_k$ using a shared projection network $g_\theta : \mathbb{R}^{N\times d} \to \mathbb{R}^{d_z}$. Formally,

$$\mathbf{F}_k = f_\omega(\mathbf{X}, \tilde{\mathbf{A}}_k), \tag{5.2}$$

$$\mathbf{v}_k = \text{ReadOut}(\mathbf{F}_k), \tag{5.3}$$

$$\mathbf{z}_k = g_\theta(\mathbf{v}_k), \quad \forall k \geq 1. \tag{5.4}$$

The encoder f_ω^k is implemented as a two-layer GCN with parameters ω and a PReLU non-linearity (He et al., 2015). The projection network g_θ is implemented as a two-layer MLP with parameters θ, also utilising a PReLU non-linearity.

5.3.3 Contrastive Objective

We train the encoder neural network using a local-global contrastive strategy, which involves contrasting the node representation of one view with the graph representation of other views, so, for K different views, we have $\binom{K}{2}$ pairs. The negative samples for each view of the graph are generated using a stochastic corruption function from Velickovic et al. (2019) given as $(\mathbf{X}^c, \tilde{\mathbf{A}}^c) = C(\mathbf{X}, \tilde{\mathbf{A}})$, where $\mathbf{H}_k^c$ represents the node-level representation of the corrupted nodes for the respective views.

The final optimisation objective is a binary cross-entropy loss with a noise contrastive objective, given by

$$\mathcal{L} = \frac{1}{N+K} \sum_k \sum_{j, j\neq k} \Big(\sum_i \mathbb{E}_{(\mathbf{X}, \tilde{\mathbf{A}}_k)} \Big[\log D(\mathbf{h}_i^j, \mathbf{z}_k) \Big] + \mathbb{E}_{(\mathbf{X}^c, \tilde{\mathbf{A}}_j^c)} \left[\log(1 - D(\mathbf{h}_i^c, \mathbf{z}_k)) \right] \Big). \tag{5.5}$$

where D represents a *discriminator* that maps a pair of a local node representation and a global graph representation to a real-valued score. This scoring mechanism acts as a proxy for mutual information and aims to assign a higher score to node-graph pairs from the same view and a lower score to pairs from different views. In line with existing works (Velickovic et al., 2019; Hassani and Khasahmadi, 2020), we implement $D(\mathbf{w}_i, \mathbf{w}_j) = \langle \mathbf{w}_i, \mathbf{w}_j \rangle$ as a dot product.

	Proximity			*Structural*			*Mixed*		
	Cora	**Citeseer**	**PubMed**	**Cornell**	**Texas**	**Wisconsin**	**Squirrel**	**Chameleon**	**Actor**
Nodes	3327	5429	19717	183	183	251	5201	2277	7600
Edges	5429	4732	44338	295	309	499	217073	36101	33544
Attributes	1433	3703	500	1703	1703	1703	2089	2325	931
Classes	7	6	3	5	5	5	5	5	5
Homophily	0.83	0.71	0.79	0.11	0.06	0.16	0.22	0.25	0.24

Table 5.1: Summary statistics of different graphs used in our experiments. We also report the homophily score in the last row to outline the difference between various datasets. A higher value means graphs have a homophily property, and a lower value implies heterophily.

5.4 Experiments and Results

5.4.1 Implementation Details and Datasets

Our codebase[1] is implemented in Python, utilising the torch geometric (Fey and Lenssen, 2019) and PyGCL (Zhu et al., 2021) (open-source library built on top of PyTorch (Paszke et al., 2019)). All experiments were conducted on Nvidia GeForce RTX 2080 GPUs with 11 GB memory. For training purpose following (Velickovic et al., 2019; Hassani and Khasahmadi, 2020; Tang et al., 2022), we use the Adam (Kingma and Ba, 2015) optimiser with a learning rate of 0.001. The restart probability α was set to 0.2 across all experiments and ε to $10e-4$. The hyperparameter values here were chosen based on the experiments on the synthetic dataset. The number of augmentations K was set to 3 based on the best-performing configuration on a structural dataset. Later, in Section 5.4.3.3, we conduct an ablation study to investigate the role of hyperparameters.

We want to remind the readers that our main objective is to leverage higher-order structural information for node representation learning, which is particularly important for heterophily graphs. To investigate our approach, we initially conducted experiments on **synthetic structural graphs** following the methodology of Donnat et al. (2018). These synthetic datasets begin with simple equivalence structures such as *House*, *Fan*, and *Star*, which are subsequently interconnected along a cycle of predefined length to form an entire graph. The nodes are labelled according to their structural roles, and the node degree is used as the node attribute.

Next, we considered **real-world structural graphs** from the WebKB (Ghani, 2001) database. WebKB comprises datasets of webpage connectivity within computer science departments across various universities. The nodes represent web pages, and the edges denote hyperlinks connecting them. The labels correspond to five structural roles: student, project, course, staff, and faculty. The node attributes are derived from the bag-of-words representation of the webpage content. We

[1] https://github.com/MdAsifKhan/graphNCE.git

utilised the Cornell, Texas, and Wisconsin graphs from the WebKB dataset for our experiments.

The multi-resolution views provide a lens to look at a graph on varying scales. By simply reducing the number of filters, we can encourage the encoder to represent the local information content without losing generality. This property makes our approach more universal and suitable for both proximal and structural graphs. Hence, we also conducted experiments on commonly used **proximal graphs**: Cora, Citeseer, and Pubmed citation networks (Sen et al., 2008; Narayanan et al., 2016), as well as **mixed graphs**: Chameleon and Squirrel from the Wikidatabase (Rozemberczki et al., 2021), and the Actor graph introduced in Donnat et al. (2018). In the proximal graphs, the nodes represent publications, and the edges represent citations. The bag-of-words representation of the publication content gives the node attributes. In the mixed graph, the nodes correspond to Wikipedia pages, the edges represent links between pages, and the attributes are bag-of-words representations of the nouns on each page. In the actor graph, the nodes represent actors, and the edges are based on the co-occurrence of their web pages.

The choice of a dataset is based on their prevalence across the existing research on heterophilic graphs (Rozemberczki et al., 2021; Donnat et al., 2018). The dataset statistics are summarised in Table 5.1. Additionally, we report an edge homophily score, which estimates the level of proximal or structural information within each dataset. For a graph $\mathcal{G} = (V, E)$, the score is defined as,

$$\textbf{Homophily} = \frac{\sum_{(v_i, v_j) \in E} \mathbb{I}(v_i = v_j)}{|E|}. \tag{5.6}$$

where $\mathbb{I}(.)$ is an indicator function that returns 1 if the condition is true, otherwise 0. An overall score close to 1 indicates a more proximal graph, while a score close to 0 suggests a more structural graph.

5.4.2 Baselines and Evaluation Setup

In this work, we compare our approach against various unsupervised representation learning methods, including, DeepWalk (Perozzi et al., 2014), node2vec (Grover and Leskovec, 2016), RolX (Henderson et al., 2012), struc2vec (Ribeiro et al., 2017), GraphWave (Donnat et al., 2018), GAE and its probabilistic counterpart VGAE (Kipf and Welling, 2016), ARGVA (Pan et al., 2018), DGI (Velickovic et al., 2019), GraphCL (You et al., 2020), MVGRL (Hassani and Khasahmadi, 2020) and NWR-GAE (Tang et al., 2022).

To ensure a fair comparison, similar to previous works (Velickovic et al., 2019; Hassani and Khasahmadi, 2020), we set the node representation size to 512 and employ the same architecture for our encoder network. Specifically, an encoder is a two-layer GCN that combines graph attribute matrix and normalised adjacency matrix to learn 512-dimensional embedding per node. This representation is followed by a two-layer MLP with a hidden size of 512 and ReLU as a non-linearity.

For our multi-resolution views, we conduct two sets of experiments. In the first set, we utilise a dedicated GCN for each view, while in the second set, we employ a shared GCN across all views. We only use dedicated GCN on synthetic datasets, and in real-world experiments, we report results for both dedicated and shared GCN in the encoder network.

We utilise clustering metrics similar to existing approaches (Donnat et al., 2018; Tang et al., 2022) to evaluate the performance on synthetic datasets. The definitions of metrics are introduced in the background Chapter 2.1.3. For real datasets, we assess the performance of downstream tasks of node classification. For classification purposes, we use a logistic regression model. To ensure consistency, we create 10 random splits, with 60% of the data allocated for training, 20% for validation, and the remaining portion for testing. The performance measure is reported as the mean and standard deviation of accuracy. Table 5.3 summarises the results for all datasets.

5.4.3 Results and Discussion

5.4.3.1 Synthetic graphs

We adopt the clustering procedure outlined by Donnat et al. (2018), and Tang et al. (2022) for this particular task. We employ agglomerative clustering with a single linkage to cluster the node representations and report three evaluation metrics: *Homogeneity* (which measures the conditional entropy of the ground-truth labels given the predicted clusters), *Completeness* (which calculates the ratio of samples with the same ground-truth label assigned to the same group), and *Silhouette score* (which compares the intra-cluster distance to the inter-cluster distance). We refer to the background in Chapter 2 for the formulae of these scores.

In this set of experiments, we set the number of filters to three ($K = 3$). The results for all synthetic graphs are presented in detail in Table 5.2. On the *House* graph, our approach yields comparable performance to RolX, GraphWave, GAE, DG, GraphCL, MVGRL, and NWR-GAE. This outcome can be attributed to the simplicity of the graphs, resulting in high scores across most methods. However, the performance of the baseline methods drops significantly on other tasks, except for NWR-GAE. In a majority of cases, our approach outperforms NWR-GAE. On the *Varied* graph, we achieve comparable Homogeneity and Completeness scores while outperforming the Silhouette score, indicating denser clusters in our case. These results provide compelling evidence that our approach best suits structural tasks best.

We observed that increasing the number of views did not lead to improvement, which could be due to the toy nature of the datasets.

Dataset	Metrics	DeepWalk	node2vec	RolX	struc2vec	GraphWave	GAE	VGAE	ARGVA	DGI	GraphCL	MVGRL	NWR-GAE	Ours
House	Homogeneity	0.01	0.01	**1.0**	0.99	**1.0**	**1.0**	0.25	0.28	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	Completeness	0.01	0.01	**1.0**	0.99	**1.0**	**1.0**	0.27	0.28	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	Silhouette	0.29	0.33	**0.99**	0.45	**0.99**	**0.99**	0.21	0.19	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
House Perturbed	Homogeneity	0.06	0.03	0.65	0.21	0.54	0.36	0.29	0.24	0.24	0.41	0.68	0.60	**0.78**
	Completeness	0.06	0.03	0.69	0.24	0.56	0.37	0.29	0.24	0.25	0.42	0.69	0.62	**0.79**
	Silhouette	0.25	0.28	0.51	0.18	0.37	0.69	0.21	0.19	0.44	0.43	0.57	0.49	**0.76**
Varied	Homogeneity	0.26	0.23	0.91	0.63	0.83	0.65	0.50	0.66	0.36	0.93	**0.93**	**0.93**	**0.93**
	Completeness	0.23	0.22	0.93	0.58	0.85	0.69	0.36	0.57	0.37	0.89	0.89	**0.94**	**0.94**
	Silhouette	0.35	0.40	0.82	0.24	0.81	0.83	0.21	0.23	0.94	0.93	0.90	0.95	**0.97**
Varied Perturbed	Homogeneity	0.30	0.30	0.74	0.46	0.69	0.44	0.42	0.57	0.36	0.70	0.73	0.78	**0.83**
	Completeness	0.27	0.27	0.72	0.43	0.68	0.45	0.43	0.49	0.36	0.63	0.67	0.81	**0.86**
	Silhouette	0.33	0.36	0.61	0.29	0.51	0.52	0.21	0.20	0.45	0.69	0.61	0.84	**0.89**

Table 5.2: Here, we compare our approach with other baselines on the task of structural role identification for synthetic dataset introduced by Donnat et al. (2018). Our approach consistently outperforms the baselines with three filter views.

Method	*Proximity*			*Structural*			*Mixed*		
	Cora	**Citeseer**	**PubMed**	**Cornell**	**Texas**	**Wisconsin**	**Chameleon**	**Squirrel**	**Actor**
DeepWalk (Perozzi et al., 2014)	82.97 ± 1.67	68.99 ± 0.95	82.39 ± 4.88	41.21 ± 3.40	41.89 ± 7.81	43.62 ± 2.46	68.03 ± 2.13	59.22 ± 2.35	23.84 ± 2.14
node2vec (Grover and Leskovec, 2016)	81.93 ± 1.43	64.56 ± 1.65	81.02 ± 1.48	40.54 ± 1.62	48.64 ± 2.92	36.27 ± 2.08	65.67 ± 2.31	48.29 ± 1.67	24.14 ± 1.02
RolX (Henderson et al., 2012)	29.70 ± 2.89	20.90 ± 0.72	39.85 ± 2.33	25.67 ± 11.78	42.56 ± 7.13	24.92 ± 13.43	22.75 ± 2.12	20.50 ± 1.18	25.42 ± 0.55
struc2vec (Ribeiro et al., 2017)	41.46 ± 1.49	51.70 ± 0.67	81.49 ± 0.33	23.72 ± 13.69	47.29 ± 7.21	24.59 ± 12.14	60.63 ± 2.90	52.59 ± 0.69	25.13 ± 0.79
GraphWave (Donnat et al., 2018)	28.83 ± 2.39	20.79 ± 1.59	20.96 ± 2.35	45.96 ± 2.20	37.45 ± 7.09	39.24 ± 5.16	17.59 ± 3.42	25.69 ± 0.53	27.29 ± 3.09
GAE (Kipf and Welling, 2016)	72.06 ± 2.54	57.10 ± 1.62	73.24 ± 0.88	45.40 ± 9.99	58.78 ± 3.41	34.11 ± 8.06	22.03 ± 1.09	29.34 ± 1.12	28.63 ± 1.05
VGAE (Kipf and Welling, 2016)	72.87 ± 1.48	60.78 ± 1.92	81.34 ± 0.79	49.32 ± 9.19	39.18 ± 8.96	38.27 ± 6.12	20.17 ± 1.30	19.57 ± 1.63	26.41 ± 1.07
ARGVA (Pan et al., 2018)	72.88 ± 3.83	63.36 ± 2.08	75.32 ± 0.63	41.08 ± 4.85	43.24 ± 5.38	41.17 ± 5.20	21.17 ± 0.78	20.61 ± 0.73	28.97 ± 1.17
DGI (Velickovic et al., 2019)	84.76 ± 1.39	71.68 ± 1.54	84.29 ± 1.07	46.48 ± 7.97	52.97 ± 5.64	55.68 ± 2.97	25.89 ± 1.49	25.89 ± 1.62	20.45 ± 1.32
GraphCL	84.23 ± 1.51	73.51 ± 1.73	82.59 ± 0.71	44.86 ± 3.73	46.48 ± 5.85	53.72 ± 1.07	26.27 ± 1.53	21.32 ± 1.66	28.64 ± 1.28
MVGRL	86.23 ± 2.71	73.81 ± 1.53	83.94 ± 0.75	53.51 ± 3.26	56.75 ± 5.97	57.25 ± 5.94	58.73 ± 2.03	40.64 ± 1.15	31.07 ± 0.29
NWR-GAE	83.62 ± 1.61	71.45 ± 2.41	83.44 ± 0.92	58.64 ± 5.61	69.62 ± 6.66	68.23 ± 6.11	$\mathbf{72.04 \pm 2.59}$	$\mathbf{64.81 \pm 1.83}$	30.17 ± 0.17
Ours (dedicated)	$\mathbf{87.67 \pm 1.49}$	$\mathbf{75.41 \pm 1.11}$	87.3 ± 0.30	66.84 ± 7.01	$\mathbf{72.89 \pm 6.60}$	$\mathbf{76.27 \pm 4.06}$	64.54 ± 2.25	43.17 ± 1.11	$\mathbf{36.01 \pm 0.68}$
Ours (shared)	85.96 ± 1.75	75.22 ± 0.99	$\mathbf{87.65 \pm 0.33}$	$\mathbf{74.70 \pm 7.56}$	72.36 ± 3.76	72.35 ± 4.75	61.49 ± 2.14	42.87 ± 1.03	35.60 ± 0.80

Table 5.3: We compare the performance of our approach with various baselines on node classification. The first task is the proximal graphs, where nodes in the local neighbourhood share the same label. Next are the structural graphs, where nodes with structural similarity share the same label, and lastly, mixed graphs with both types of node labels. Our setup works best across all three graph categories and significantly outperforms baselines on structural graphs. We note that the performance of NWR-GAE is best on the squirrel.

5.4.3.2 Structural, Mixed and Proximal graphs

The comparison of node classification accuracy with baseline methods across three types of graph data is presented in Table 5.3. We utilise (K=4) multi-resolution views for structural and mixed graphs, while for proximal graphs, we employ (K=2) views in addition to the local graph adjacency. Interestingly, we observed that the performance did not improve when more filters were included.

On structural graphs, our approach outperforms the baselines significantly, achieving an improvement of 16.06% on Cornell (using shared GCN), 3.27% on Texas (using dedicated GCN), and 8.04% on Wisconsin compared to the best baseline method NWR-GAE. Furthermore, our approach surpasses all baselines on proximal graphs, with a 3.7% improvement on the PubMed graph, which is notably significant in size. These results demonstrate the universality of our approach.

On mixed data, our approach outperforms NWR-GAE by 5.84% on the actor dataset. However, NWR-GAE achieves better performance on the chameleon and squirrel datasets. We hypothesise that this can be attributed to NWR-GAE using a more expressive decoder and additional loss terms based on degree and neighbourhood size.

Regarding our two encoder variants, we observe that the GCN *shared* variant achieves higher performance on the Cornell and PubMed tasks. Additionally, it performs competitively on other tasks. This result can be attributed to the explicit frequency resolution provided by the multi-resolution views, which helps prevent representation smoothing often observed with GCN networks (Chen et al., 2020a). On the other hand, the GCN *dedicated* variant, which employs a separate encoder for each view, emerges as the overall best method.

The importance of representations is primarily determined by their applications to downstream tasks. By allowing the feature representation network to learn the appropriate scale invariance, we can tailor the number of views accordingly. For example, we can reduce the number of filter views in graphs where local connectivity is more crucial. Conversely, introducing more filters can improve performance in graphs where structural information is more relevant. Overall, the multi-resolution views are a simple and efficient strategy for training self-supervised representation learners.

Next, we investigate the effect of including higher-order resolution views on the performance of structural datasets.

5.4.3.3 Ablation Study

We conducted ablation experiments to examine the impact of increasing the number of resolutions on the downstream performance. We only use structural data for this purpose since we expect higher-order filters to be more beneficial for learning structural information. Figure 5.3 illustrates the results on structural data.

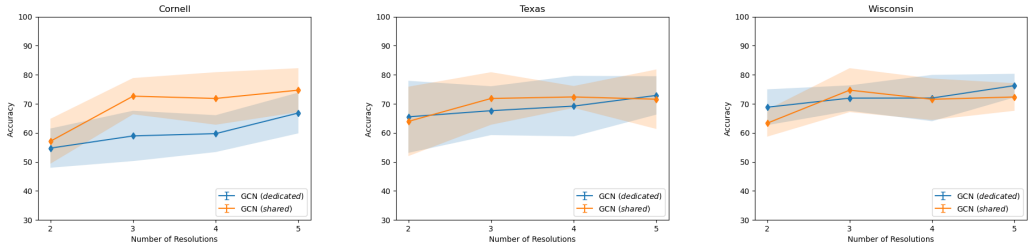


Figure 5.3: Ablation Study. Here, we investigate the effect of including higher-resolution graph views on top of the local graph structure. We report node classification accuracy averaged across ten random splits on structure graphs with increasing views.

We observed that increasing the resolution leads to improved performance. However, beyond a certain level, the performance gain becomes marginal. We hypothesise that this phenomenon arises because the same node attributes are used for message passing across different scales. Due to a dyadic scale, the coarse-grained filters get sparse, and the message passing may smooth out and not update further. We consider this a limitation that can be a potential future direction.

5.5 Conclusion

This chapter introduced wavelet diffusion filters as a means to incorporate multi-resolution graph views in self-supervised learning for heterophilic graphs. These filters have desirable properties, including stability to signal deformation and perturbation and the ability to capture higher-order structural regularities within graphs. Utilising these filters as graph augmentations in the contrasting training objective introduces a resolution invariance in the learned representations, which proves advantageous for addressing structural equivalence problems. We demonstrated the effectiveness of our approach through empirical experiments on both synthetic and real-world structural datasets. Importantly, our method also outperformed baseline approaches on proximal graphs, highlighting its general applicability for learning representations of nodes on a broad range of graphs.

The research work in this chapter was carried out in mid-2022, and the paper is available on an Arxiv platform. The work is currently under preparation for submission to a conference or a journal venue.

Most existing approaches for representation learning of graph-structured data either work on homophilic or heterophilic graphs. Our work presents a strategy under which the same framework can be used for the two types of graphs by simply incorporating suitable invariance by using diffusion filters on graphs as augmentations.

Chapter 6

Conclusion

Deep representation learning methods use neural networks to encode high-dimensional data into a lower-dimensional latent space. The learned representations are expected to capture relevant information that characterises them as “good"–for example, properties like the smoothness of latent space, which is vital for robustness against noise. This thesis has demonstrated the importance of geometric principles to inspect and account for necessary properties within the framework of deep representation learning by addressing three challenging problems: the disentanglement of latent factors of variation, the assessment of the robustness of deep latent variable models, and the learning of representations from graph-structured data.

Each chapter in the thesis introduced a specific research problem and a comprehensive discussion of the geometric principles employed for representation learning. To summarise, we outline the main research questions and contributions made through three projects.

6.1 Key Contributions

The first project of this thesis focused on the research question of learning interpretable disentangled latent representations from image sequence data. Sequential variational autoencoders (VAEs) are commonly employed for learning an embedding space that decomposes latent factors into *content* and *motion* components. However, these models rely on a latent dynamical model to capture the underlying 1-dimensional motion manifold. Moreover, the dynamical model’s non-linear nature hinders the interpretability of the latent factors. This project proposed a novel approach incorporating a physics-inspired dynamical model inducing a symplectic geometry in the latent motion subspace. By leveraging the symplectic structure, the latent factors are naturally organised based on the concept of energy conservation. This formulation introduces a meaningful notion of proximity in the latent motion space, where representations that might be nearby in space may not be reachable in time due to the infeasibility of the underlying motion.

Extensive experiments were conducted to demonstrate the efficacy of this approach on

various datasets, including a toy physics problem involving rotating balls, sprite videos, and real-world human expression datasets. The results showcased the benefits of the proposed framework across several tasks, such as the control generation of dynamic scenes and motion transfer. Moreover, the incorporation of geometry enhanced the interpretability of the learned representations, as the dynamics were linear and characterised by constant bilinear energy terms.

The impact of this work extends to various applications, such as enabling more accurate and controllable modelling in practical applications such as designing robot manipulation or control in the latent space, physics-engine for game design, transfer-learning from simulation to real-world scenarios, etc. Furthermore, the notion of energy in latent space provides a new perspective for designing interpretable and reliable latent dynamical models that might be relevant to a broad community.

The second project of this thesis focused on studying the robustness of VAEs using Riemannian geometry. VAEs are widely employed for representation learning in various real-world applications, ranging from robotics (Asenov et al., 2019) to molecule design (Gómez-Bombarelli et al., 2018) and genomics analysis (Choi et al., 2023). Understanding their robustness against adversarial perturbations is crucial to ensure their reliable deployment in practical scenarios.

The primary contribution of this project was an analysis of the geometry of the latent space and its impact on the robustness of VAEs. The findings revealed that the learned latent space exhibits distortions in the form of high curvature, along with regions of low or zero density. Adversaries can exploit these distortions to compromise the representation capacity of VAEs. To quantify robustness, novel scores were introduced that utilise the pullback-metric tensor. Furthermore, the project investigated the robustness induced by alternative optimisation techniques used in β-VAE and demonstrated that increasing the value of the β parameter reduces latent space distortions, thereby improving robustness. Additionally, a simple yet effective training scheme was proposed, leveraging the concept of *mix-up* to fill regions of low or zero density. This scheme proved to enhance the resilience of VAEs against adversarial attacks.

By leveraging geometry to understand and enhance the robustness of VAEs, this work provides valuable tools and insights for developing more reliable and secure deep-learning models. The exploration of geometric properties offers a novel perspective for future research in designing robust architectures and defences against adversarial attacks, ultimately advancing the field of secure deep learning.

The final project tackled the challenge of learning node representations in heterophilic graphs, where non-local nodes exhibit similar properties based on their local connectivity. Such graphs are typical in applications such as fraudster detection. Therefore, the ability to group nodes with various complex interdependencies in an embedding space is crucial. This project proposed a novel approach that leverages diffusion geometry to generate graph augmentations. These augmentations resemble bandpass filters capturing hierarchical dependencies of several

orders. The augmented graphs are then used to train an encoder neural network with a contrastive objective for representation learning. By incorporating multiscale information in the form of graph augmentation, the contrastive framework encourages the encoder to group nodes with heterophilic properties. Experimental evaluations on widely used benchmarks demonstrate the superiority of our proposed method compared to existing approaches.

The impact of this research extends to several real-world applications, including anomaly in router networks and community detection.

6.2 Broader Impact

A general motivation of deep representation learning methods is to employ complex neural network mappings to embed diverse data domains into a rich, meaningful representation space. These representations must encode necessary information and satisfy specific properties to be relevant for solving downstream tasks. Bengio et al. (2013a) summarised some properties, including robustness, disentanglement, etc. However, developing models that account for these is non-trivial, and much of the existing work, as discussed in individual chapters, introduces schemes that either achieve these approximately or are too convoluted to understand intuitively. This work has shown the importance of geometry-based principles in tackling the above challenges in building relevant properties in the learned representations. To deploy these representation learning frameworks in real-world scenarios such as robot control in the latent space (Asenov et al., 2019), it is vital to account for robustness that is attributed to properties like the smoothness of latent space. This thesis has demonstrated the use of geometry in investigating and establishing such properties of deep learning approaches, as exemplified by the three projects.

Geometry is ubiquitous in physics, especially for its role in studying various symmetry groups and manifolds. By integrating geometry-based ideas in deep representation learning approaches, we can utilise principled theoretical advancements for developing effective and reliable representation learning methods that capture the true structure of the real world.

6.3 Limitations and Future Work

By means of this thesis, we have demonstrated the importance of geometry-based inductive biases in enforcing suitable properties in the latent space. In scenarios with limited training data, using such inductive biases helps improve sample efficiency and generalisation within a specific range of tasks. However, in applications where we do not have apriori knowledge of structure in a dataset, the assumptions could limit the adaptability of a model. For instance, in Chapter 2, we use Hamiltonian energy, which is an explicit, rigid constraint imposed in the motion component

of latent space. Suppose the underlying motion does not preserve or follow a constant of motion. In that case, it will fail to encode suitable information in a representation space, thereby limiting its generalisation in downstream tasks. Alternatively, one could introduce soft constraints, which slack a model to learn more versatile representations. In practice, such flexibility comes at the cost of sample efficiency and, therefore, can require more data for training purposes. Finding a balance between soft and hard inductive biases is an open problem requiring a model to capture domain-specific knowledge and generalisable features effectively.

Each of the studies presented in this thesis opens up promising avenues for future research. Here, we outline specific limitations and a few key directions that can be pursued:

Chapter 3 proposed disentanglement method is a bit rigid in its form since we explicitly restrict every motion to a unique subspace that requires prior knowledge of the action label of a sequence. A potential direction can be modelling complex dynamical action as a composition of primitive motion components using a prior mixing distribution. For instance, consider the following two scenarios: in one, an individual standing at a fixed location is hand waving, and in another, walking in a circular path. We can view a complex motion as a combination of two where one is simultaneously walking and hand waving. Here, we can assume walking and hand-waving are primitive actions.

Another potential direction is handling irregularly sampled sequences, changes in tempo, or reversals. We hypothesise this can be achieved by allowing a more flexible prior on the spacing between time steps.

An interesting real-world extension can be employing *HALO* framework for molecular trajectories. The trajectory of a molecule evolves under constant Hamiltonian energy, which results from forces due to bonds or folding of molecules. Considering its dynamic nature becomes important when predicting properties such as stability or binding with other molecules. A model can learn more enhanced representations that capture relevant properties by learning a latent space where conformational dependent properties can be expressed in a motion space and independent one in a content space. Moreover, unrolling trajectories in latent space can reduce the computation cost of trajectory simulation.

In **Chapter 4**, we only considered the unsupervised attack where there is no prespecified target and the goal of the adversary is mainly to deviate away from the latent manifold. A more interesting scenario would be to inspect a targeted attack where an adversary finds a perturbation that can change the encoding to an encoding of a desired target. Moreover, there can be different forms of attack by replacing the l_2-norm with more general $p-$norms. A geometric analysis of such scenarios could provide valuable insights into understanding the robustness and generalisation of deep generative models.

Recently, a few mechanisms have been proposed to reduce the distortion in latent space and improve the robustness of VAEs (Willetts et al., 2021; Kuzina et al., 2021). A potential

future direction can be establishing the geometrical understanding of these robustness schemes by analysing their pullback metric tensors using proposed evaluation scores.

The geometry induced in the latent space depends on the choice of architectures of encoder-decoder neural networks. Future research can explore alternative architectures and training strategies for robustness by studying the geometric properties of latent manifolds. Thus, geometric principles are explicitly incorporated in designing network architectures.

Understanding the impact of geometry on transfer learning is another important avenue for future research. Investigating how the geometry of latent manifold influences the transferability of learned representations across domains, tasks, or modalities can lead to more effective and efficient transfer learning algorithms.

Chapter 5 utilises diffusion geometry to construct multiple augmentations of graphs and use a contrastive objective to train an encoder neural network. For K augmentations, there are $\binom{K}{2}$ pairs which increase the computational cost of contrasting loss, especially with scalability associated with large-scale graph datasets. An avenue for future work can be reducing this cost using contrastive multiview coding (Tian et al., 2020). In the current work, we consider a number of views (K) as a hyperparameter that is required apriori. A potential extension can be introducing an optimisation loop to learn it adaptively for a given task.

Since the inception of this work, some recent approaches have targeted learning representations of heterophilic graphs. Bodnar et al. (2022) proposed sheaf diffusion that can address the over-smoothing of GCNs and learn a hierarchy of information that proves helpful for complex node classification tasks encountered in heterophilic graphs. A key difference to our work is that we provide multiscale information explicitly as augmentations to the learning framework. In contrast, sheaf diffusion shows that generalising GCN to sheaves can allow the network to infer it in a data-driven way. Sheaf diffusion is also trained in a supervised setting. A potential future scope can be an extensive comparison with recent developments. Additionally, we can investigate the applicability of sheaf diffusion GCN in a self-supervised setting and whether our augmentations combined with architecture that builds on sheaf diffusion layers can lead to improvement.

6.4 Summary

The three research projects presented in this thesis underscore the importance of geometry in developing sample-efficient deep representation learning methods. Notable improvements have been made in disentanglement, robustness analysis, and representation learning for heterophilic graphs by integrating geometry into respective tasks. These contributions have broader implications, offering insights, methodologies, and techniques to improve deep representation learning models' interpretability, generalisability, and reliability.

Bibliography

Achille, A. and Soatto, S. (2018). Emergence of Invariance and Disentanglement in Deep Representations . *Journal of Machine Learning Research*, 19(50):1–34.

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer.

Aifanti, N., Papachristou, C., and Delopoulos, A. (2010). The MUG Facial Expression Database. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, pages 1–4.

Al-Rfou, R., Perozzi, B., and Zelle, D. (2019). Ddgk: Learning graph representations for deep divergence graph kernels. In *The World Wide Web Conference*, WWW '19, page 37–48, New York, NY, USA. Association for Computing Machinery.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein Generative Adversarial Networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR.

Arvanitidis, G., Hansen, L. K., and Hauberg, S. (2018). Latent Space Oddity: on the Curvature of Deep Generative Models. In *International Conference on Learning Representations*.

Arvanitidis, G., Hauberg, S., and Schölkopf, B. (2020). Geometrically Enriched Latent Spaces. In *International Conference on Artificial Intelligence and Statistics*.

Asenov, M., Burke, M., Angelov, D., Davchev, T., Subr, K., and Ramamoorthy, S. (2019). Vid2param: Modeling of Dynamics Parameters From Video. *IEEE Robotics and Automation Letters*, 5(2):414–421.

Bader, P., Blanes, S., and Casas, F. (2019). Computing the matrix exponential with an optimized Taylor polynomial approximation. *Mathematics*, 7(12):1174.

Balasubramanian, M., Schwartz, E. L., Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2002). The isomap algorithm and topological stability. *Science*, 295(5552):7–7.

Barratt, S. and Sharma, R. (2018). A note on the inception score. *arXiv preprint arXiv:1801.01973*.

Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396.

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research*, 7(85):2399–2434.

Bengio, Y., Courville, A., and Vincent, P. (2013a). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Bengio, Y., Courville, A., and Vincent, P. (2013b). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Bengtsson, I., Zyczkowski, K., and Milburn, G. J. (2008). Geometry of quantum states: an introduction to quantum entanglement by ingemar bengtsson and karol zyczkowski. *Quantum Inf. Comput.*, 8(8):860.

Bird, A., Williams, C. K. I., and Hawthorne, C. (2022). Multi-Task Dynamical Systems. *Journal of Machine Learning Research*, 23(230):1–52.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

Bodnar, C., Di Giovanni, F., Chamberlain, B., Lió, P., and Bronstein, M. (2022). Neural Sheaf Diffusion: A Topological Perspective on Heterophily and Oversmoothing in GNNs. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 18527–18541. Curran Associates, Inc.

Bojchevski, A. and Günnemann, S. (2018). Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *International Conference on Learning Representations*.

Bondesan, R. and Lamacraft, A. (2019). Learning Symmetries of Classical Integrable Systems. In *ICML 2019 Workshop on Theoretical Physics for Deep Learning*.

Borgwardt, K., Ghisu, E., Llinares-López, F., O'Bray, L., and Rieck, B. (2020). Graph kernels: State-of-the-art and future challenges. *Found. Trends Mach. Learn.*, 13(5–6):531–712.

Botev, A., Jaegle, A., Wirnsberger, P., Hennes, D., and Higgins, I. (2021). Which priors matter? Benchmarking models for learning latent dynamics. In Vanschoren, J. and Yeung, S., editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran.

Brand, M. (2003). Charting a manifold. In *Advances in neural information processing systems*, pages 985–992.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. (2021). *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. arXiv.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2014). Spectral networks and locally connected networks on graphs. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Bui, T. D., Ravi, S., and Ramavajjala, V. (2018). Neural Graph Learning: Training Neural Networks Using Graphs. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 64–71, New York, NY, USA. Association for Computing Machinery.

Cai, H., Zheng, V. W., and Chang, K. C.-C. (2018). A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.

Camuto, A., Willetts, M., Roberts, S. J., Holmes, C. C., and Rainforth, T. (2021). Towards a Theoretical Understanding of the Robustness of Variational Autoencoders. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021,*, volume 130 of *Proceedings of Machine Learning Research*, pages 3565–3573. PMLR.

Cangea, C., Veličković, P., Jovanović, N., Kipf, T., and Liò, P. (2018). Towards Sparse Hierarchical Graph Classifiers. *arXiv preprint arXiv:1811.01287*.

Casale, F. P., Dalca, A., Saglietti, L., Listgarten, J., and Fusi, N. (2018). Gaussian Process Prior Variational Autoencoders. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Caselles-Dupré, H., Garcia Ortiz, M., and Filliat, D. (2019). Symmetry-based disentangled representation learning requires interaction with environments. In Wallach, H., Larochelle,

H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Cayton, L. and Dasgupta, S. (2006). Robust euclidean embedding. In *Proceedings of the 23rd international conference on machine learning*, pages 169–176.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., and Mukhopadhyay, D. (2021). A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45.

Chamberlain, B. P., Clough, J., and Deisenroth, M. P. (2017). Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*.

Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., and Sun, X. (2020a). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3438–3445.

Chen, F., Wang, Y.-C., Wang, B., and Kuo, C.-C. J. (2020b). Graph Representation Learning: A Survey. *APSIPA Transactions on Signal and Information Processing*, 9.

Chen, L. and Buja, A. (2009). Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219.

Chen, N., Klushyn, A., Ferroni, F., Bayer, J., and Van Der Smagt, P. (2020c). Learning Flat Latent Manifolds with VAEs. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., and Smagt, P. (2018a). Metrics for Deep Generative Models. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1550. PMLR.

Chen, R. T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. (2018b). Isolating Sources of Disentanglement in Variational Autoencoders. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018c). Neural Ordinary Differential Equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020d). A Simple Framework for Contrastive Learning of Visual Representations. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Chen, T., Sun, Y., Shi, Y., and Hong, L. (2017). On Sampling Strategies for Neural Network-Based Collaborative Filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 767–776, New York, NY, USA. Association for Computing Machinery.

Chen, Y. and Wu, L. (2022). Graph neural networks: Graph structure learning. In Wu, L., Cui, P., Pei, J., and Zhao, L., editors, *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 297–321. Springer Singapore, Singapore.

Choi, Y., Li, R., and Quon, G. (2023). sivae: interpretable deep generative models for single-cell transcriptomes. *Genome Biology*, 24(1):29.

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. (2020). Debiased Contrastive Learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8765–8775. Curran Associates, Inc.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. (2015). A Recurrent Latent Variable Model for Sequential Data. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Cohen, T. and Welling, M. (2014). Learning the Irreducible Representations of Commutative Lie Groups. In *International Conference on Machine Learning*, pages 1755–1763. PMLR.

Coifman, R. R. and Maggioni, M. (2006). Diffusion Wavelets. *Applied and computational harmonic analysis*, 21(1):53–94.

Connor, M. and Rozell, C. (2020). Representing Closed Transformation Paths in Encoded Network Latent Space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3666–3675.

Cox, M. A. and Cox, T. F. (2008). Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer.

Cox, T. F. and Cox, M. A. (1991). Multidimensional scaling on a sphere. *Communications in Statistics-Theory and Methods*, 20(9):2943–2953.

Cranmer, M., Sanchez Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., and Ho, S. (2020). Discovering Symbolic Models from Deep Learning with Inductive Biases. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17429–17442. Curran Associates, Inc.

Cremers, D. and Yuille, A. (2003). A Generative Model Based Approach to Motion Segmentation. In *Joint Pattern Recognition Symposium*, pages 313–320. Springer.

Culpepper, B. and Olshausen, B. (2009). Learning transport operators for image manifolds. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

Dai, H., Dai, B., and Song, L. (2016). Discriminative Embeddings of Latent Variable Models for Structured Data. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2702–2711, New York, New York, USA. PMLR.

Dai, S., Gan, Z., Cheng, Y., Tao, C., Carin, L., and Liu, J. (2021). Apo-vae: Text generation in hyperbolic space. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 416–431. Association for Computational Linguistics.

Dandi, Y., Das, A., Singhal, S., Namboodiri, V., and Rai, P. (2020). Jointly Trained Image and Video Generation using Residual Vectors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3028–3042.

Davidson, T., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. (2018). Hyperspherical Variational Auto-Encoders. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 856–865.

de Ocáriz Borde, H. S., Kazi, A., Barbero, F., and Lio, P. (2023). Latent Graph Inference using Product Manifolds. In *The Eleventh International Conference on Learning Representations*.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Dehmamy, N., Barabási, A.-L., and Yu, R. (2019). Understanding the representation power of graph neural networks in learning graph topology. *Advances in Neural Information Processing Systems*, 32.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Denton, E. L. and Birodkar, v. (2017). Unsupervised Learning of Disentangled Representations from Video. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Dhingra, B., Shallue, C. J., Norouzi, M., Dai, A. M., and Dahl, G. E. (2018). Embedding Text in Hyperbolic Spaces. In Glavas, G., Somasundaran, S., Riedl, M., and Hovy, E. H., editors, *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing, TextGraphs@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 6, 2018*, pages 59–69. Association for Computational Linguistics.

Donnat, C., Zitnik, M., Hallac, D., and Leskovec, J. (2018). Learning Structural Node Embeddings via Diffusion Wavelets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, page 1320–1329, New York, NY, USA. Association for Computing Machinery.

Dupont, E., Martin, M. B., Colburn, A., Sankar, A., Susskind, J., and Shan, Q. (2020). Equivariant Neural Rendering. In *International Conference on Machine Learning*, pages 2761–2770. PMLR.

Easton, R. W. (1993). Introduction to Hamiltonian dynamical systems and the N-body problem (KR Meyer and GR Hall). *SIAM Review*, 35(4):659–659.

Eastwood, C. and Williams, C. K. I. (2018). A Framework for the Quantitative Evaluation of Disentangled Representations. In *International Conference on Learning Representations*.

Eklund, D. and Hauberg, S. (2019). Expected path length on random manifolds. *ArXiv*, abs/1908.07377.

Elliott, R. J. and Krishnamurthy, V. (1999). New Finite-Dimensional Filters for Parameter Estimation of Discrete-Time Linear Gaussian Models. *IEEE Transactions on Automatic Control*, 44(5):938–951.

Eraslan, G., Avsec, Ž., Gagneur, J., and Theis, F. J. (2019). Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, 20(7):389–403.

Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J., and van de Meent, J.-W. (2019). Structured Disentangled Representations. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference*

on Artificial Intelligence and Statistics, volume 89 of *Proceedings of Machine Learning Research*, pages 2525–2534. PMLR.

Falorsi, L., de Haan, P., Davidson, T. R., and Forré, P. (2019). Reparameterizing distributions on lie groups. In Chaudhuri, K. and Sugiyama, M., editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 3244–3253. PMLR.

Fey, M. and Lenssen, J. E. (2019). Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Fraccaro, M., Kamronn, S., Paquet, U., and Winther, O. (2017). A disentangled recognition and nonlinear dynamics model for unsupervised learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Franceschi, J.-Y., Delasalles, E., Chen, M., Lamprier, S., and Gallinari, P. (2020). Stochastic Latent Residual Video Prediction. In *International Conference on Machine Learning*, pages 3233–3246. PMLR.

Friedman, J. H. (1997). On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77.

Gama, F., Ribeiro, A., and Bruna, J. (2019). Diffusion Scattering Transforms on Graphs. In *International Conference on Learning Representations*.

Gao, F., Wolf, G., and Hirn, M. (2019). Geometric Scattering for Graph Data Analysis. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2122–2131. PMLR.

Ghani, R. (2001). CMU World Wide Knowledge Base (WebKB) Project. *Online Referencing: http://www. cs. cmu. edu/˜ webkb/(Access Date: 22 June 2018)*.

Ghosh, P., Sajjadi, M. S. M., Vergari, A., Black, M. J., and Schölkopf, B. (2020). From Variational to Deterministic Autoencoders. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Gibson, W. (1960). Nonlinear factors in two dimensions. *Psychometrika*, 25(4):381–392.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS central science*, 4(2):268–276.

Gondim-Ribeiro, G., Tabacof, P., and Valle, E. (2018). Adversarial Attacks on Variational Autoencoders. *arXiv preprint arXiv:1806.04646*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian Neural Networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Grover, A. and Leskovec, J. (2016). Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 855–864, New York, NY, USA. Association for Computing Machinery.

Grover, A., Zweig, A., and Ermon, S. (2019). Graphite: Iterative Generative Modeling of Graphs. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2434–2444. PMLR.

Hamilton, W., Ying, Z., and Leskovec, J. (2017a). Inductive Representation Learning on Large Graphs. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hamilton, W. L. (2020). Graph Representation Learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017b). Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74.

Hammond, D. K., Vandergheynst, P., and Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150.

Hassani, K. and Khasahmadi, A. H. (2020). Contrastive Multi-View Representation Learning on Graphs. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4116–4126. PMLR.

Hauberg, S. (2018). Only Bayes should learn a manifold (on the estimation of differential geometric structure from data). *ArXiv*, abs/1806.04994.

He, J., Lehrmann, A., Marino, J., Mori, G., and Sigal, L. (2018). Probabilistic Video Generation Using Holistic Attribute Control. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Henderson, K., Gallagher, B., Eliassi-Rad, T., Tong, H., Basu, S., Akoglu, L., Koutra, D., Faloutsos, C., and Li, L. (2012). RolX: Structural Role Extraction and Mining in Large Graphs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 1231–1239, New York, NY, USA. Association for Computing Machinery.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. (2018). Towards a Definition of Disentangled Representations. *arXiv preprint arXiv:1812.02230*.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. (2017). beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural computation*, 9(8):1735–1780.

Hogg, D. (1983). Model-Based Vision: A Program to See a Walking Person. *Image and Vision Computing*, 1(1):5–20.

Hsieh, J.-T., Liu, B., Huang, D.-A., Fei-Fei, L. F., and Niebles, J. C. (2018). Learning to Decompose and Disentangle Representations for Video Prediction. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Hsu, W.-N., Zhang, Y., and Glass, J. (2017). Unsupervised Learning of Disentangled and Interpretable Representations from Sequential Data. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hurri, J. and Hyvärinen, A. (2003). Temporal and Spatiotemporal Coherence in Simple-cell Responses: a Generative Model of Natural Image Sequences. *Network: Computation in Neural Systems*, 14(3):527–551. PMID: 12938770.

Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., and Makedon, F. (2021). A Survey on Contrastive Self-Supervised Learning. *Technologies*, 9(1).

Jin, W., Barzilay, R., and Jaakkola, T. S. (2018). Junction Tree Variational Autoencoder for Molecular Graph Generation. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2328–2337. PMLR.

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45.

Kannan, A., Jojic, N., and Frey, B. (2005). Generative Model for Layers of Appearance and Deformation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pages 166–173. PMLR.

Karl, M., Soelch, M., Bayer, J., and van der Smagt, P. (2017). Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data. In *International Conference on Learning Representations*.

Kazi, A., Cosmo, L., Ahmadi, S.-A., Navab, N., and Bronstein, M. M. (2022). Differentiable graph module (dgm) for graph convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1606–1617.

Khrulkov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., and Lempitsky, V. (2020). Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428.

Kim, H. and Mnih, A. (2018). Disentangling by Factorising. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2649–2658. PMLR.

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Kipf, T. N. and Welling, M. (2016). Variational Graph Auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Klein, F. (1872). *Vergleichende betrachtungen über neuere geometrische forsuchungen*. A. Deichert.

Kos, J., Fischer, I., and Song, D. X. (2017). Adversarial Examples for Generative Models. *2018 IEEE Security and Privacy Workshops (SPW)*, pages 36–42.

Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.

Kuzina, A., Welling, M., and Tomczak, J. M. (2021). Diagnosing Vulnerability of Variational Auto-Encoders to Adversarial Attacks. *RobustML Workshop@ICLR 2021*.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253.

Lamb, A., Verma, V., Kannala, J., and Bengio, Y. (2019). Interpolated Adversarial Training: Achieving Robust Neural Networks Without Sacrificing Too Much Accuracy. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, page 95–103, New York, NY, USA. Association for Computing Machinery.

Laskin, M., Srinivas, A., and Abbeel, P. (2020). CURL: Contrastive Unsupervised Representations for Reinforcement Learning. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5639–5650. PMLR.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of machine learning research*, 6(Nov):1783–1816.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, J. M. (2006). *Riemannian manifolds: An Introduction to Curvature*, volume 176. Springer Science & Business Media.

Leimeister, M. and Wilson, B. J. (2018). Skip-gram word embeddings in hyperbolic space. *arXiv preprint arXiv:1809.01498*.

Leskovec, J. and Krevl, A. (2014). Snap datasets: Stanford large network dataset collection.

Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. (2019). CayleyNets: Graph Convolutional Neural Networks With Complex Rational Spectral Filters. *IEEE Transactions on Signal Processing*, 67(1):97–109.

Lin, X., Quan, Z., Wang, Z.-J., Ma, T., and Zeng, X. (2020). KGNN: Knowledge Graph Neural Network for Drug-Drug Interaction Prediction. In *IJCAI*, volume 380, pages 2739–2745.

Liu, M., Wang, Z., and Ji, S. (2021). Non-Local Graph Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, M.-Y., Breuel, T., and Kautz, J. (2017). Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep Learning Face Attributes in the Wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I. (2016). Adversarial Autoencoders. In *International Conference on Learning Representations*.

Mallat, S. (2012). Group Invariant Scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398.

Martin, J. and Elster, C. (2020). Inspecting adversarial examples using the fisher information. *Neurocomputing*, 382:80–86.

Mathieu, E., Rainforth, T., Siddharth, N., and Teh, Y. W. (2019). Disentangling Disentanglement in Variational Autoencoders. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4402–4412. PMLR.

McDonald, R. P. (1962). A general approach to nonlinear factor analysis. *Psychometrika*, 27(4):397–415.

Memisevic, R. (2012). On Multi-View Feature Learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, page 1067–1074. Omnipress.

Mika, S., Schölkopf, B., Smola, A. J., Müller, K.-R., Scholz, M., and Rätsch, G. (1999). Kernel pca and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Miladinović, Đ., Gondal, W., Schölkopf, B., Buhmann, J. M., and Bauer, S. (2019). Disentangled State Space Models: Unsupervised Learning of Dynamics Across Heterogeneous Environments.

Minderer, M., Sun, C., Villegas, R., Cole, F., Murphy, K. P., and Lee, H. (2019). Unsupervised Learning of Object Structure and Dynamics from Videos. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mohammadi, H. B., Hauberg, S., Arvanitidis, G., Neumann, G., and Rozo, L. D. (2022). Reactive Motion Generation on Learned Riemannian Manifolds. *arxiv*, abs/2203.07761.

Narayanan, A., Chandramohan, M., Chen, L., Liu, Y., and Saminathan, S. (2016). subgraph2vec: Learning Distributed Representations of Rooted Sub-graphs from Large Graphs. *arXiv preprint arXiv:1606.08928*.

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347.

Nickel, M. and Kiela, D. (2018). Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3776–3785. PMLR.

Noether, E. (1918). Invariant Variation Problems, gott.

Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., and Zhang, C. (2018). Adversarially Regularized Graph Autoencoder for Graph Embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2609–2615.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Park, J., Lee, M., Chang, H. J., Lee, K., and Choi, J. Y. (2019). Symmetric Graph Convolutional Autoencoder for Unsupervised Graph Representation Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Paulus, M., Choi, D., Tarlow, D., Krause, A., and Maddison, C. J. (2020). Gradient Estimation with Stochastic Softmax Tricks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5691–5704. Curran Associates, Inc.

Pavlovic, V., Rehg, J. M., and MacCormick, J. (2000). Learning Switching Linear Models of Human Motion. In *NIPS*, volume 2, page 4.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710. Association for Computing Machinery.

Pfau, D., Higgins, I., Botev, A., and Racanière, S. (2020). Disentangling by Subspace Diffusion. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17403–17415. Curran Associates, Inc.

Pfau, D., Petersen, S., Agarwal, A., Barrett, D. G. T., and Stachenfeld, K. L. (2019). Spectral Inference Networks: Unifying Deep and Spectral Learning. In *International Conference on Learning Representations*.

Pu, Y., Gan, Z., Henao, R., Li, C., Han, S., and Carin, L. (2017). Vae learning via stein variational gradient descent. In *Advances in Neural Information Processing Systems*, pages 4236–4245.

Quessard, R., Barrett, T., and Clements, W. (2020). Learning disentangled representations and group structure of dynamical environments. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19727–19737. Curran Associates, Inc.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Rao, R. and Ruderman, D. (1998). Learning lie groups for invariant visual perception. In Kearns, M., Solla, S., and Cohn, D., editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press.

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Rezende, D. J. and Viola, F. (2018). Taming VAEs. *arXiv preprint arXiv:1810.00597*.

Ribeiro, L. F., Saverese, P. H., and Figueiredo, D. R. (2017). Struc2vec: Learning Node Representations from Structural Identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 385–394, New York, NY, USA. Association for Computing Machinery.

Rifai, S., Dauphin, Y. N., Vincent, P., Bengio, Y., and Muller, X. (2011a). The Manifold Tangent Classifier. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. (2011b). Contractive auto-encoders: Explicit invariance during feature extraction. In *Icml*.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. (2011c). Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 833–840, Madison, WI, USA. Omnipress.

Roberts, A., Engel, J., and Eck, D. (2017). Hierarchical Variational Autoencoders for Music. In *Workshop on Machine Learning for Creativity and Design, NIPS*.

Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., and Bronstein, M. (2020). Temporal graph networks for deep learning on dynamic graphs. In *ICML 2020 Workshop on Graph Representation Learning*.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Roweis, S. and Ghahramani, Z. (1999). A Unifying Review of Linear Gaussian Models. *Neural computation*, 11(2):305–345.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

Rozemberczki, B., Allen, C., and Sarkar, R. (2021). Multi-Scale Attributed Node Embedding. *Journal of Complex Networks*, 9(2). cnab014.

Ruiz, L., Gama, F., and Ribeiro, A. (2020). Gated Graph Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 68:6303–6318.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Chen, X. (2016). Improved Techniques for Training GANs. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Sanchez-Gonzalez, A., Bapst, V., Cranmer, K., and Battaglia, P. (2019). Hamiltonian Graph Networks with ODE integrators. *arXiv preprint arXiv:1909.12790*.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. (2020). Learning to Simulate Complex Physics with Graph Networks. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8459–8468. PMLR.

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. (2018). Modeling Relational Data with Graph Convolutional Networks. In Gangemi, A., Navigli, R., Vidal, M.-E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., and Alam, M., editors, *The Semantic Web*, pages 593–607. Springer International Publishing.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, 29(3):93.

Shao, H., Kumar, A., and Fletcher, P. T. (2017). The Riemannian Geometry of Deep Generative Models. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 428–4288.

Shervashidze, N., Schweitzer, P., Van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 12(9).

Shi, H., Fan, H., and Kwok, J. T. (2020). Effective decoding in graph auto-encoder using triadic closure. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):906–913.

Simonovsky, M. and Komodakis, N. (2018). GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders. In Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., and Maglogiannis, I., editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 412–422, Cham. Springer International Publishing.

Skopek, O., Ganea, O., and Bécigneul, G. (2020). Mixed-curvature variational autoencoders. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sohl-Dickstein, J., Wang, C. M., and Olshausen, B. A. (2010). An Unsupervised Algorithm for Learning Lie Group Transformations. *arXiv preprint arXiv:1001.1027*.

Srivastava, N., Mansimov, E., and Salakhudinov, R. (2015). Unsupervised Learning of Video Representations using LSTMs. In *International conference on machine learning*, pages 843–852. PMLR.

Starner, T. and Pentland, A. (1997). Real-Time American Sign Language Recognition from Video using Hidden Markov Models. In *Motion-based recognition*, pages 227–243. Springer.

Storkey, A. and Williams, C. K. I. (2003). Image Modeling with Position-Encoding Dynamic Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):859–871.

Sun, F.-Y., Hoffman, J., Verma, V., and Tang, J. (2020). InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *International Conference on Learning Representations*.

Sun, H., Zhu, T., Zhang, Z., Jin, D., Xiong, P., and Zhou, W. (2023). Adversarial Attacks Against Deep Generative Models on Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3367–3388.

Sun, K., Koniusz, P., and Wang, Z. (2019). Fisher-Bures Adversary Graph Convolutional Networks. In *Uncertainty in Artificial Intelligence*, page 161.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014). Intriguing properties of neural networks. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Tabacof, P., Tavares, J., and Valle, E. (2016). Adversarial Images for Variational Autoencoders. *Adversarial Training Workshop, NIPS*.

Tang, M., Li, P., and Yang, C. (2022). Graph Auto-Encoder via Neighborhood Wasserstein Reconstruction. In *International Conference on Learning Representations*.

Teh, Y. W. and Roweis, S. T. (2003). Automatic alignment of local representations. In *Advances in neural information processing systems*, pages 865–872.

Tian, Y., Krishnan, D., and Isola, P. (2020). Contrastive Multiview Coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer.

Tomczak, J. M. (2022). *Deep Generative Modeling*. Springer.

Tong, A., Wenkel, F., MacDonald, K., Krishnaswamy, S., and Wolf, G. (2021). Data-Driven Learning of Geometric Scattering Networks. In *IEEE MLSP*.

Toth, P., Rezende, D. J., Jaegle, A., Racanière, S., Botev, A., and Higgins, I. (2020). Hamiltonian Generative Networks. In *International Conference on Learning Representations*.

Tsitsulin, A., Mottin, D., Karras, P., and Müller, E. (2018). Verse: Versatile graph embeddings from similarity measures. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 539–548, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Tu, L. W. (2011). *An introduction to manifolds*. Springer.

Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. (2018). MoCoGAN: Decomposing Motion and Content for Video Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vahdat, A. and Kautz, J. (2020). NVAE: A Deep Hierarchical Variational Autoencoder. In *Neural Information Processing Systems (NeurIPS)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is All you Need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. (2019). Deep Graph Infomax. *ICLR (Poster)*, 2(3):4.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. (2019). Manifold mixup: Better representations by interpolating hidden states. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR.

Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. (2017). Decomposing Motion and Content for Natural Video Sequence Prediction. In *International Conference on Learning Representations*.

Wang, C., Pan, S., Long, G., Zhu, X., and Jiang, J. (2017a). MGAE: Marginalized Graph Autoencoder for Graph Clustering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 889–898, New York, NY, USA. Association for Computing Machinery.

Wang, J., Yi, X., Guo, R., Jin, H., Xu, P., Li, S., Wang, X., Guo, X., Li, C., Xu, X., et al. (2021). Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2614–2627.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019). Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.*, 38(5).

Wang, Y., Wang, J., Cao, Z., and Farimani, A. B. (2022). Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.*, 4(3):279–287.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Wang, Z., Chen, C., and Li, W. (2017b). Predictive Network Representation Learning for Link Prediction. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 969–972, New York, NY, USA. Association for Computing Machinery.

Weston, J., Ratle, F., and Collobert, R. (2008). Deep Learning via Semi-Supervised Embedding. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1168–1175, New York, NY, USA. Association for Computing Machinery.

Willetts, M. J., Camuto, A., Rainforth, T., Roberts, S., and Holmes, C. C. (2021). Improving {vae}s' Robustness to Adversarial Attack. In *International Conference on Learning Representations*.

Wu, B., Nair, S., Martín-Martín, R., Fei-Fei, L., and Finn, C. (2021). Greedy Hierarchical Variational Autoencoders for Large-Scale Video Prediction. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2318–2328.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. (2018). MoleculeNet: A Benchmark for Molecular Machine Learning. *Chemical science*, 9(2):513—530.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv*.

Yanardag, P. and Vishwanathan, S. (2015). Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374.

Yang, C., Wang, H., Zhang, K., Chen, L., and Sun, L. (2021). Secure Deep Graph Generation with Link Differential Privacy. In Zhou, Z.-H., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3271–3278. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Yang, T., Arvanitidis, G., Fu, D., Li, X., and Hauberg, S. (2018). Geodesic Clustering in Deep Generative Models. *ArXiv*, abs/1809.04747.

Yildiz, C., Heinonen, M., and Lahdesmaki, H. (2019). ODE2VAE: Deep Generative Second Order ODEs with Bayesian Neural Networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yingzhen, L. and Mandt, S. (2018). Disentangled Sequential Autoencoder. In *International Conference on Machine Learning*, pages 5670–5679. PMLR.

Yoon, J., Jarrett, D., and van der Schaar, M. (2019). Time-series Generative Adversarial Networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. (2020). Graph Contrastive Learning with Augmentations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5812–5823. Curran Associates, Inc.

Zaripova, K., Cosmo, L., Kazi, A., Ahmadi, S.-A., Bronstein, M. M., and Navab, N. (2023). Graph-in-Graph (GiG): Learning interpretable latent graphs in non-Euclidean domain for biological and healthcare applications. *Medical Image Analysis*, 88:102839.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, M. and Chen, Y. (2018). Link Prediction Based on Graph Neural Networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Zhao, C., Fletcher, P. T., Yu, M., Peng, Y., Zhang, G., and Shent, C. (2019). The Adversarial Attack and Detection under the Fisher Information Metric. In *Association for the Advancement of Artificial Intelligence*. AAAI Press.

Zhou, Y., Barnes, C., Lu, J., Yang, J., and Li, H. (2019). On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753.

Zhu, Y., Min, M. R., Kadav, A., and Graf, H. P. (2020). S3VAE: Self-supervised sequential VAE for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547.

Zhu, Y., Xu, Y., Liu, Q., and Wu, S. (2021). An Empirical Study of Graph Contrastive Learning. In Vanschoren, J. and Yeung, S., editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Zitnik, M. and Leskovec, J. (2017). Predicting Multicellular Function through Multi-layer Tissue Networks. *Bioinformatics*, 33(14):i190–i198.