ON DISENTANGLED REPRESENTATION LEARNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAZAL MOĞULTAY ÖZCAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JULY 2024

Approval of the thesis:

ON DISENTANGLED REPRESENTATION LEARNING

submitted by **HAZAL MOĞULTAY ÖZCAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Prof. Dr. Fatoş Tünay Yarman Vural
Supervisor, **Computer Engineering, METU** _____

Prof. Dr. Sinan Kalkan
Co-supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Ramazan Gökberk Cinbiş
Computer Engineering, METU _____

Prof. Dr. Fatoş Tünay Yarman Vural
Computer Engineering, METU _____

Prof. Dr. Pınar Duygulu Şahin
Computer Engineering, Hacettepe University _____

Assoc. Prof. Dr. Emre Akbaş
Computer Engineering, METU _____

Assoc. Prof. Dr. Hacer Yalım Keleş
Computer Engineering, Hacettepe University _____

Date:25.07.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Hazal Moğultay Özcan

Signature :

ABSTRACT

ON DISENTANGLED REPRESENTATION LEARNING

Moğultay Özcan, Hazal
Ph.D., Department of Computer Engineering
Supervisor: Prof. Dr. Fatoş Tünay Yarman Vural
Co-Supervisor: Prof. Dr. Sinan Kalkan

July 2024, 102 pages

Disentanglement is the problem of obtaining a representation, where the underlying sources of variation, generating the data are independently captured. A pioneering DRL method is Beta Variational Auto Encoder (β-VAE), which introduces a new hyperparameter β to weight the disentanglement term of the VAE loss function, which is empirically optimized.

In this thesis, we make three contributions to improve the disentanglement capacity of β-VAE. First, to automatically estimate the β parameter, we propose Learnable VAE (L-VAE). L-VAE, mitigates the hyperparameter optimization problem of β-VAE by learning the relative weights of different terms in the loss function to dynamically control the trade-off between disentanglement and reconstruction. These weights and the parameters of the model architecture are learned concurrently, eliminating the complexity of empirical hyperparameter optimization. Both β-VAE and L-VAE introduce the same weight on all of the dimensions of the representation. However, we show that each dimension has a different degree of disentanglement. In order to dynamically learn an independent weight per dimension, we propose Multi-

Dimensional Learnable VAE (mdL-VAE) as an extension to L-VAE. We show that both methods provide on par or better disentanglement-reconstruction trade-off without tuning a β hyperparameter and that mdL-VAE provide useful insights about the entanglement between the underlying factors of variations. Finally, we introduce a novel correlation-based disentanglement (CbD) measure that allows interpreting the degree of disentanglement for each dimension of the representation robustly. We demonstrate that CbD provides complementary and useful insights about the disentanglement performance of different disentanglement methods.

Keywords: Disentanglement, representation learning, disentanglement measure, hyperparameter optimization

ÖZ

AYRIŞIK GÖSTERİM ÖĞRENME ÜZERİNE

Moğultay Özcan, Hazal
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Fatoş Tünay Yarman Vural
Ortak Tez Yöneticisi: Prof. Dr. Sinan Kalkan

Temmuz 2024, 102 sayfa

Ayrışıklik (Disentanglement), verileri oluşturan temel varyasyon kaynaklarının bağımsız olarak yakalandığı bir gösterim öğrenme problemidir. Bu konuda önde gelen çalışmalardan biri olan Beta Variational Auto Encoder (β-VAE) modeli, VAE modelini bir empirik olarak eniyilenen bir hiperparametre olan β ile genişletir.

Bu tezde, β-VAE modelinin ayrıştırma kapasitesini arttırmayı amaçlayan üç yenilik öneriyoruz. İlk olarak, β değerini otomatik olarak bulmaya yarayan Learnable VAE (L-VAE) modelini öneriyoruz. L-VAE isimli modelin amacı β-VAE modelinde karşılaştığımız parametre eniyilemesi sorununu ortadan kaldırmaktır. L-VAE modeli, kayıp fonksyonundaki ayrıştırma ve yeniden oluşturma terimlerinin ağırlıklarını öğrenerek bu ikisi arasındaki dengeyi dinamik olarak kontrol edebilir. Kayıp fonksyonuna getirdiğimiz bu ağırlıklar mimari ağırlıklarıyla eşzamanlı olarak öğrenildiğinden β parametresinin deneysel olarak eniyilenmesi yükünü ortadan kaldırır. Hem β-VAE hem de VAE modelleri gösterimin bütün boyutları üzerinde aynı ağırlıkları kullanmaktadır. Ancak biz bu tezde her boyutun farklı bir ayrıştırma seviyesinde kalabileceğini gösteriyoruz. Her boyut için farklı bir ağırlık öğrenmek için L-VAE modelini

geliştirerek Multi-Dimensional Learnable VAE (mdL-VAE) isimli ikinci bir model öneriyoruz. Deneysel olarak iki modelin de β eniyilemesine gerek duymaksızın literatürdeki modellere eşdeğer veya daha iyi ayrıştırma/yeniden oluşturma dengesi oluşturduğunu ve mdL-VAE modelinin altta yatan değişkenlik faktörlerini anlama konusunda önemli bilgiler sunduğunu gösteriyoruz. Son olarak, Correlation-based Disentanglement (CbD) isimli korelasyon bazlı bir ayrışıklık ölçüm yöntemi öneriyoruz. Bu yöntem gösterimdeki her boyut için ayrışıklık miktarını güçlü şekilde ölçmekte, ve konu üstünde üstünde yorum yapmayı kolaylatırmaktadır. CbD ile ayrışık gösterim öğrenme modellerinim performanslarını ölçülürken, modeller hakkında önemli bilgiler elde ettiğimizi deneysel olarak gösteriyoruz.

Anahtar Kelimeler: Ayrışıklık, gösterim öğrenme, ayrışıklık ölçütü, hiperparametre eniyileme

To the memory of my dearest grandfather,

ACKNOWLEDGMENTS

This thesis would not be possible without the support of many individuals. First, I would like to express my deepest gratitude to my advisors, Prof. Dr. Fatoş Tünay Yarman Vural and Prof. Dr. Sinan Kalkan, for their constant guidance. They encouraged me to pursue an academic career. I will always be grateful to them and will do my best to carry forward the lessons they have taught me.

I would also like to thank my thesis committee members, Assoc. Prof. Dr. Ramazan Gökberk Cinbiş, Prof. Dr. Pınar Duygulu Şahin, Assoc. Prof. Dr. Emre Akbaş, and Assoc. Prof. Dr. Hacer Yalım Keleş, for their valuable comments and guidance.

I am thankful to Kıvılcım Başak Vural for helping me understand the mathematics behind many problems. I also thank Ekin Sağlıcan and Ezgi Ekiz Emiroğlu for standing by me through all the ups and downs.

I would like to thank my best friends, Dilek Koptekin and Duha Alioğlu. I feel so privileged to have you as friends, and despite the distance between us, I always feel your support. Our night-long discussions sparked many new ideas, and I am thankful for every moment we shared.

I want to express my deepest gratitude to my parents; their support and love are the foundation of all that I achieved.

Last but not least, I want to thank my husband, Erkmen Özcan. Life is a journey, and ours is just beginning. You continually show me how to make the most of every moment, turning each day into something memorable. Your presence gives me strength and courage, and I am deeply grateful for every moment we share. I could not complete this thesis without your unwavering support.

I also gratefully acknowledge the computational resources kindly provided by METU-ROMER, Center for Robotics and Artificial Intelligence.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

VAE	Variational Auto-Encoder
β-VAE	β Variational Auto-Encoder
ELBO	Evidence Lower BOund
MI	Mutual Information
KL Divergence	Kullback-Liebler Divergence
L-VAE	Learnable Variational Auto-Encoder
mdL-VAE	Multi-Dimensional Learnable VAE

CHAPTER 1

INTRODUCTION

Deep learning architectures inherently possess limitations concerning their capacity for generalization to large datasets and interpretability of the learned models [4, 22, 50, 66]. A promising approach to mitigating these limitations, especially interpretability, is *disentanglement* [14, 21, 42]. Disentanglement assumes each image sample in a dataset is generated by the mixture of independent factors of variations, which contribute to the data generation process that can represent the implicit attributes of the data, such as rotation, translation, shape, or shadow [4, 22]. The main task is to find low-dimensional embeddings, which are also called representations, of images in the feature space such that different dimensions learn different factors of variations of the data generation process.

Figure 1.1 shows an example disentanglement problem by Wang et al. [62]. The dataset in this example has six attributes that are caused by the factors of variations in the data-generating process, namely object size, shape, color, wall color, floor color and viewing angle. The main task of disentangled representation learning is to identify and learn these attributes in separate dimensions, such that the learned representation is interpretable.

Disentanglement assumes the data is generated by the mixture of independent factors of variations. Disentangled representation learning (DRL) methods estimate low-dimensional embeddings of high-dimensional data while modelling these factors. However, in most of the practical problems, this process is unobservable. In these cases, for the sake of mathematical tractability, we may assume that each independent generative factor can be modelled by an unobservable low-dimensional Gaussian distribution, which embeds the properties hidden in the data. DRL methods

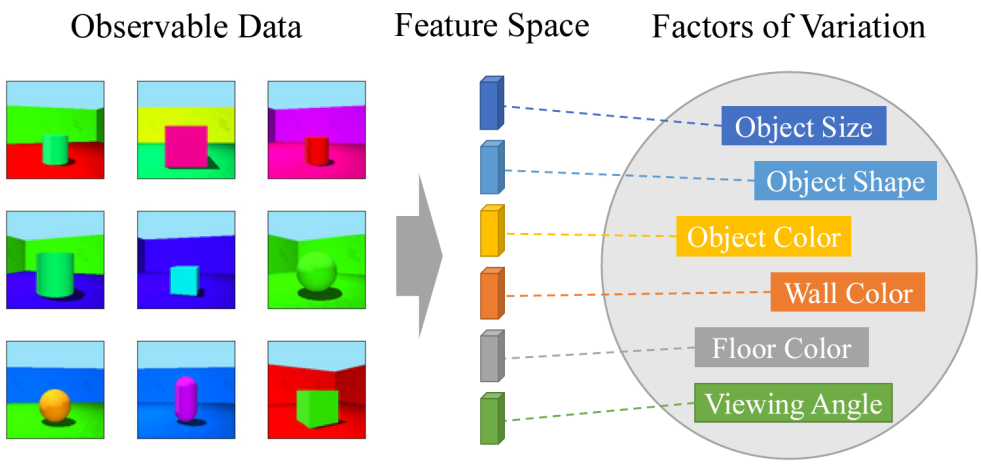


Figure 1.1: Disentangled representation learning aims to identify the underlying sources of variation generating the data. The simple image at the left shows the squares, cylinders and spheres in different size, color, view angle and floor color. [Figure Source: [62]]

attempt to learn a latent representation, modelling for these Gaussian distributions, which are assumed to be the source of the underlying variations among the samples of the same category. DRL methods aim at capturing each factor of variation independently [3, 4].

Disentanglement has received significant attention within the computer vision community. This approach has been utilized in a wide range of problems, including facial image analysis [38], image dehazing [15], face hallucination [16], video frame generation [11, 25, 67], identity learning [38, 48], image-to-image translation [35], and face forgery detection [19].

1.1 Beta Variational Auto-Encoders

Variational Auto-Encoders (VAE) are one of the pioneering models for learning disentangled representations. Rather than learning a representation with entangled properties, zero mean unit variance Gaussian priors are enforced on each dimension of the latent properties. For this purpose, the Kullback-Leibler (KL) Divergence term between the priors and the learnt distribution is minimized alongside the reconstruction

loss to enforce the disentanglement in the learnt representation. The representations learned in this model indicate disentanglement properties in their latent subspaces [62]. There are several disentanglement models based on VAEs [9, 29]. In order to enforce stronger regularization, hence better disentanglement, Higgins et al. proposed a new model called β-VAE where they weight the KL Divergence term by a hyperparameter β [6, 22]. They show that this new model learns representations with better disentanglement properties compared to VAEs. Augmenting the original VAE architecture with a hyperparameter β on the KL Divergence emphasizes the importance of the KL term providing stronger independence among the latent properties. However, there are still some limitations of β-VAEs for learning a disentangled representation.

The first limitation of β-VAE is to find an optimal hyperparameter β by empirical methods, which is a computationally exhaustive problem. Even if we can estimate an optimal β parameter, minimization of the loss function and the KL divergence with a fixed β parameter does not necessarily result in the minimum loss with maximum disentangled representation with respect to some disentanglement metrics.

The second drawback of the β-VAE model is that, in most cases, it is observed that β-VAE increases the reconstruction loss for the sake of better disentanglement [9], which may result in poor data representation. In other words, with a fixed β parameter, it is not possible to control the disentangled factors of variations for minimum reconstruction loss.

In summary, β-VAE is not proven to capture complex interactions between different factors of variation and does not assure the minimum reconstruction loss. Therefore, further research is needed to improve the effectiveness of β-VAE and to address the limitations of VAEs in disentangled representation learning problems. In this study, our main goal is to address these issues with two new extensions to the β-VAE, as summarized below.

1.2 Thesis Scope and Contributions

In this thesis, we focus on the problem of DRL and introduce novel methods and measures that make DRL easier and more interpretable compared to the state-of-the-

art disentanglement methods. To be specific, we make the following contributions:

Contribution 1: Analysis of the effect of β Hyperparameter on disentangling the hidden attributes. We conduct an analysis on β-VAE and highlight several critical observations regarding the β hyperparameter. For example, we observe that β-VAE is highly sensitive to the β hyperparameter. Furthermore, in some datasets, $\beta < 1$ can surprisingly provide better disentanglement quality.

Contribution 2: A Correlation-based Disentanglement Measure. In order to analyse and compare the disentanglement properties of different DRL methods, we propose a novel disentanglement measure called Correlation-based Disentanglement (CbD). The proposed CbD measure computes pairwise Pearson Correlations between the dimensions of the latent representation and the underlying sources of variation of the dataset. Assuming each dimension should be correlated with a single factor and vice versa, we evaluate the disentanglement score per model. Since CbD uses Pearson Correlation, it produces robust outcomes that do not have any model selection dependencies. Moreover, the pairwise Pearson Correlation values serve as a guide to analyse the disentanglement in each dimension of the representation.

Contribution 3: Learnable VAE for Easier DRL. In order to reduce the complexity of hyperparameter optimization introduced by the β-VAE model and the reconstruction abilities, we propose a new model called Learnable Variational Auto-Encoder (L-VAE). L-VAE minimizes the reconstruction loss and maximizes disentanglement while learning the weight parameters of the loss function all at once. In this sense, L-VAE can be considered as an extended version of β-VAE. L-VAE introduces an automated weighting method for β-VAE, in which the model learns the trade-off between the reconstruction loss and the amount of disentanglement on the learned space, controlled by the KL divergence term. It eliminates the burden of optimization of the hyperparameter β for minimizing the reconstruction loss while maximizing disentanglement based on the original loss function. We compare L-VAE to 5 different disentanglement models, namely, VAE, β-VAE, Control VAE, Dynamic VAE and σ-VAE. Our analysis suggests that L-VAE produces lower reconstruction losses than β-VAE with $\beta > 1$. Moreover, L-VAE consistently provides the best or second-best performance in terms of disentanglement without empirically tuning the β hyperpa-

rameter. We further demonstrate that the learned weights of L-VAE are aligned with β-VAE.

Contribution 4: Multi-Dimensional Learnable VAE for Easier DRL. In real-world scenarios, the data generation process is typically hidden, which poses a challenge in accurately determining the number of disentangled attributes. Furthermore, it is difficult to ascertain whether the factors are uniformly disentangled. However, β-VAE based models assume all factors can be equally disentangled since they give equal weight to each dimension of the representation. We argue that, in some practical situations, this assumption may not hold. In order to relax this assumption, we propose Multi-Dimensional Learnable VAE (mdL-VAE), which learns independent weights per dimension and, therefore, allows several dimensions to learn entangled attributes if necessary. Since the weights of the loss function are learned alongside the network parameters, mdL-VAE effectively eliminates the need for hyperparameter tuning, addressing the first limitation of β-VAE. We show that mdL-VAE provides the best or second-best disentanglement performances across several disentanglement measures. We also show that mdL-VAE achieves separation between the learned weights, i.e. learned weights are higher for the disentangled dimensions. Therefore, these weights can be used as a tool to discriminate the disentangled dimensions or select the number of dimensions, further increasing the interpretability of the learned model.

Our contributions have been disseminated with the following publications:

- H. Mogultay, S. Kalkan, and F. T. Yarman Vural. "L-VAE: Variational auto-encoder with learnable beta for disentangled representation", Under Review.
- H. Mogultay, S. Kalkan, and F. T. Yarman Vural. "mdL-VAE: Multi-Dimensional Learnable Variational Autoencoder", Under Review.

1.3 Thesis Outline

In Chapter 2, we provide a mathematical definition of disentanglement. Moreover, we provide a taxonomy for the disentanglement studies in the literature. We discuss

the disentanglement methods available in the literature based on this categorization in Chapter 2.

Prior to the detailed explanation of the methods, we present the datasets we use in the experimentation and the architecture details in Chapter 3. We explain the neural network architecture, used throughout the thesis, keeping the same for all of the methods and datasets. In the rest of the chapters, we explain our proposed methods and validate the results by thorough experiments on popular datasets.

In Chapter 4, we provide an analysis of the effect of the hyperparameter β on the reconstruction loss and the degree of disentanglement achieved by the β-VAE model. We discuss the relative changes of reconstruction loss and KL divergence with empirically changing β. We also discuss the effect of β on the effort of disentanglement on the relatively entangled attributes in this chapter.

In Chapter 5, we propose a new measure, Correlation-based Disentanglement (CbD). We also provide an explanation of the available measures and compare them to CbD. Finally, we study the characteristics of these measures for different scenarios and datasets.

In Chapter 6, we introduce our first proposed model, called Learnable-VAE (L-VAE) model. We conduct extensive comparisons with five other methods, namely, VAE [31], β-VAE[22], ControlVAE [58], DynamicVAE [57], and σ-VAE [54], providing a comprehensive understanding of the respective strengths and weaknesses of L-VAE.

Finally, in Chapter 7, we introduce our second proposed model called Multi-Dimensional Learnable VAE (mdL-VAE). Following the same analyses in Chapter 6, we conduct extensive comparisons with five other methods, namely, VAE [31], β-VAE[22], ControlVAE [58], DynamicVAE [57], and σ-VAE [54]. We discuss the pros and cons of mdL-VAE over other DRL methods.

Chapter 8 concludes the work with discussions on this study and possible future directions.

CHAPTER 2

DISENTANGLEMENT IN COMPUTER VISION

In this chapter, we provide an overview of disentangled representation learning within the domain of computer vision. We start by discussing the definition of disentanglement in this area. Then, we combine the available definitions to formally introduce our definition of disentanglement. With this definition, we delve into the literature and examine the disentanglement studies under two main categories:

1. **Explicit models:** These models assume a prior information and/or labels about the hidden attributes embedded in each category of the dataset. Hence, supervised or self-supervised methods can be employed to disentangle the hidden attributes.

2. **Implicit models:** These models aim to learn the hidden attributes without any supervision. Variational Auto-Encoder (VAE) [31] and its variations are prominent examples of such models. The loss function of the VAEs consists of two terms: reconstruction loss and Kullback-Leibler divergence. The second term leads the representation to disentangle hidden attributes of the dataset.

Since VAEs form the baseline of our study, we investigate VAEs and their subsequent variations in detail, focusing on their working mechanisms and analysing their drawbacks. Finally, we discuss the main properties of disentanglement and studies that focus on measuring these properties.

2.1 Definition(s) of Disentanglement

In this section, we explain the different definitions of the term disentanglement from the literature. The first definition is based on group theory by Higgins et al. [21]. In this definition, they decompose the representation space into subspace of hidden attributes and explain disentanglement analysing these subspaces. On the other hand, VAE-based approaches define disentanglement based on the independence of the dimensions of the representation. Therefore, the group theoretic definition is reduced to a special set with the statistical definitions adopted by VAE-based approaches. After discussing these definitions, we provide a formal definition for the term which we use for the rest of this thesis.

2.1.1 What is Entanglement and Disentanglement?

According to Oxford dictionary [23], the term *entanglement* is defined as the act of becoming caught or twisted in something, while *disentanglement* denotes the process of freeing something from an entangled situation. In quantum physics, the term *entanglement* describes a phenomenon in which a pair of subatomic particles are intrinsically connected to each other, yet their quantum properties remain interdependent even when they are separated by vast distances [24]. Bengio et al. [3] borrowed this term from physics and introduced it into the field of representation learning. They use *disentanglement* to explain a set of properties within a learned representation. Specifically, they use the term to explain the relationship among different dimensions of the representation. Numerous mathematical definitions are formulated to define these properties within the conceptual framework of machine learning [4, 6, 9, 21, 22, 41].

One of the well-regarded perspectives on disentanglement is provided by Higgins et al. [21]. They propose a group-theoretic definition which models the inherent symmetries, such as rotation, translation and size of objects by symmetry groups. This definition suggests decomposing the representation space into subspaces of hidden attributes. Each of which is characterized by the variation induced by a specific subgroup. In this approach, the actions of all other subgroups must leave the respective subspace invariant. Although this rigorous definition has significant theoretical merit,

it has limited impact in finding the disentangled representation in practice. Consider the example given in Figure 1.1, which shows a toy image from [62]. In this specific example, the images are composed of objects with different size, shape, color with varying wall and floor colors and different viewing angles. These attributes are disentangled from each other. For instance, any alteration in the wall color does not affect the object's color, or any other properties. According to the symmetry group definition, provided by Higgins et al., each of these attributes is represented in a specific subspace of the representation space, and any action on a specific subgroup leaves the rest of the subspaces invariant.

Alternatively, a mathematically tractable approach to defining disentanglement is based on the assumption that the data consists of a set of hidden attributes [10, 33]. These hidden attributes are one of the major causes of the variations among the objects of the same category. In the above example provided in Figure 1.1, the hidden attributes correspond to the colors or sizes of the shapes. However, it is important to note that the hidden attributes could be shared across categories, such as viewing angle or rotation of the object. Consequently, disentanglement is defined as the process of separating these embedded attributes within the observations, while simultaneously learning a representation function. In this study, we adopt this definition, emphasizing the computational aspect of disentanglement.

Within this framework, observations are assumed to be generated by a mixture of independent Gaussian sources that can represent the attributes of a sample. These sources are not observable and are often called *underlying generative factors of the data-generating process* or *hidden sources of variation*. We will refer to these as generative factors or sources of variation throughout this thesis. The main goal of disentanglement is then to learn a representation, where each of these generative factors is modelled compactly within non-overlapping subspaces of the overall learned representation. This approach allows for the interpretation of generative factors through the independent dimensions inherent in the representation. Notice that this approach defines disentanglement based on independence.

2.1.2 Disentanglement: A Formal Definition

Formally, consider a dataset with M samples

$$\mathcal{D} = \{\mathcal{X}, \mathcal{W}, \mathcal{Y}\},$$

where,

$$x^{(i)} \in \mathcal{X} \in \mathbb{R}^{W \times H},$$

denotes an $W \times H$-dimensional image sample i, and each image sample $\mathbf{x}^{(i)}$ is generated by the mixture of K *conditionally independent* and *unobserved* factors of variation denoted as,

$$\mathbf{w}^{(i)} \in \mathbb{R}^{K}.$$

In other words, the sample image $\mathbf{x}^{(i)}$ can be simulated using its source distributions, expressed as,

$$\mathbb{P}(x^{(i)}) \sim Sim(\mathbf{w}^{(i)}).$$

These distributions $\mathbf{w}^{(i)}$ are assumed to induce the dataset attributes such as color or shape. A dataset can also include the labels of some or all of these attributes $\mathbf{w}^{(i)}$ as

$$\mathbf{y}^{(i)} \in \mathbb{R}^{N}.$$

Note that if a model directly uses the labels of the attributes, $\mathbf{y}^{(i)}$ to produce a representation, we will refer to them as supervised in terms of disentanglement. Note that the number of dimensions in the labels, N is generally smaller than the actual number of generative factors, K.

Disentanglement is then defined as the problem of learning an L-dimensional representation,

$$\mathbf{z}^{(i)} \in \mathbb{R}^{L},$$

for image sample $\mathbf{x}^{(i)}$, with,

$$\mathbb{P}(\mathbf{x}^{(i)}|\mathbf{z}^{(i)}) \sim Sim(\mathbf{w}^{(i)}).$$

Each dimension j of the representation $\mathbf{z}^{(i)}$, $\mathrm{z}_j^{(i)}$, learns a *single* source of variation $\mathrm{w}_j^{(i)}$. This definition has two implications. First, altering a single source of variation in the data-generating process will result in a change of a single dimension in the

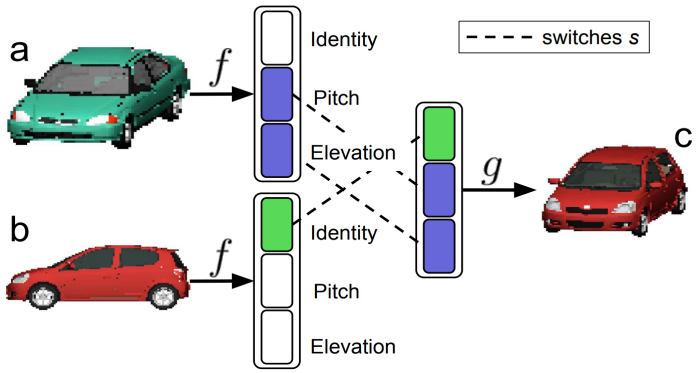


Figure 2.1: Consider the example of learning disentangled representations for cars. An encoder-decoder architecture is trained to learn the identity, pitch, and elevation attributes of cars. The same encoding function, f, is used to infer a representation for two distinct cars (top: green and bottom: red). In the next step, different chunks of the representations are used to construct a single representation, and decoding function g is used to reconstruct the image. The resulting image has the identity of the red car. However, it resembles the same elevation and pitch attributes as the green car. [Figure Source: [52]]

representation space. Second, since all hidden factors are learned independently and compactly, it would be possible to analyze and interpret them through their compact representation. Note that in theory, the number of dimensions in the representation L should be selected as $L \geq K$ to capture the information about all of the hidden attributes.

Figure 2.1, provided by [52] illustrates a toy scenario. In this example, we assume a dataset of car images, where each car possesses three distinct attributes: identity, camera elevation, and pitch (rotation of the car). An encoding function, f, generates a disentangled representation for sample images, and these encoded representations are fed into a decoding function, g, to regenerate images. Assuming the encoding function f yields a disentangled representation, each of the three attributes is learned independently and compactly in separate chunks of this representation (See blue and green chunks of the representation vector in Figure 2.1. Therefore, interchanging these chunks will alter the attributes in the decoded image.

In this example, the identity attribute is taken from the representation of a red car,

denoted as b; whereas elevation and pitch attributes are taken from the green car, labeled as, a. These chunks are then rearranged to construct a representation, and this representation is fed into the decoding function, g, to generate an image labelled as c. The output image c reflects the identity of the red car, b since the identity-related information is taken from the representation of the image, b. Similarly, elevation and pitch attributes are the same with the green car, a. This example demonstrates the importance of disentangled representation for machine learning problems.

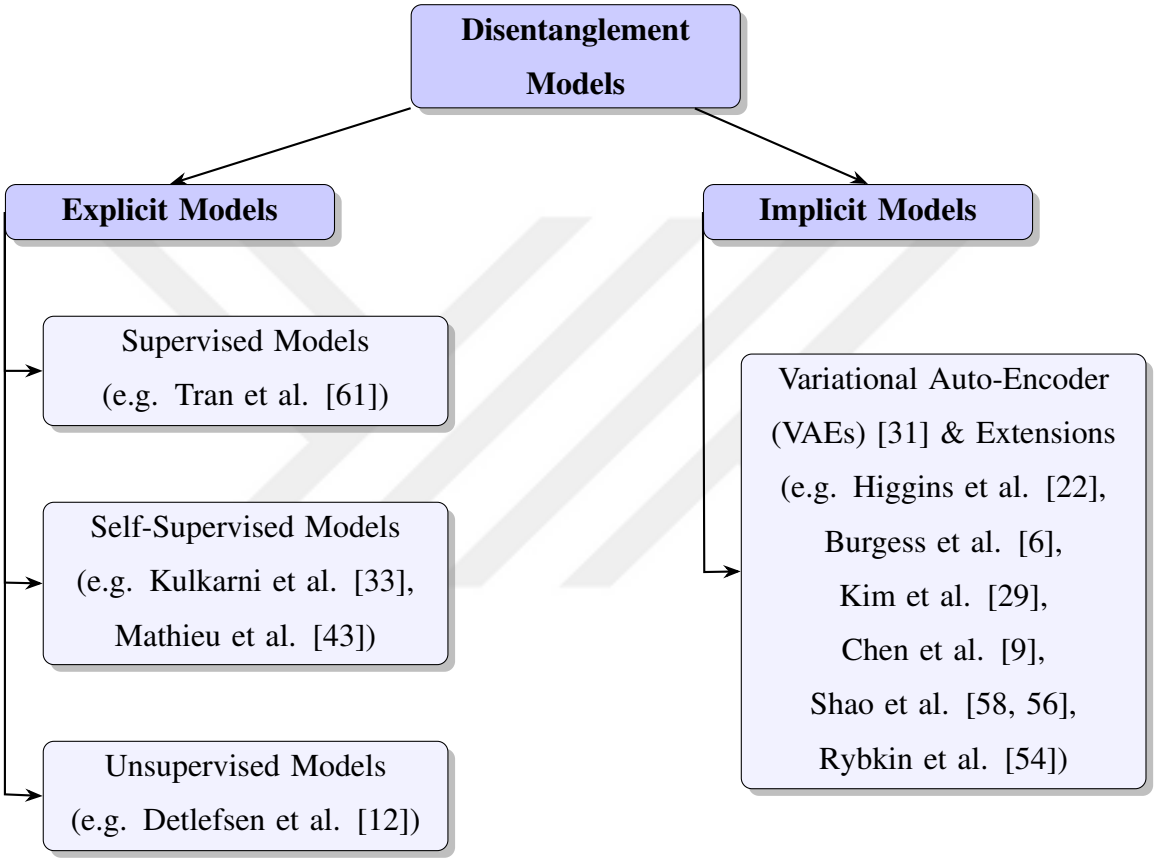


Figure 2.2: A brief taxonomy of disentanglement models.

2.2 Disentanglement in Machine Learning Literature: A Taxonomy and Review

Various methodologies for disentangled representation learning are proposed, relying on frameworks, such as the ones based on Variational Auto-Encoders and its variants [9, 22], Generative Adversarial Networks [10, 46, 26, 27], and causality based models

[55, 65].

Disentanglement is generally studied without supervision, i.e. without using any attribute labels y_i caused by the underlying generative factors. However, supervised disentanglement has also been studied (see, e.g., [61]).

In this thesis, we systematically categorize the studies of disentanglement learning into two primary categories, namely, explicit and implicit models, to establish a brief taxonomy shown in Figure 2.2. In explicit models, the primary objective is to disentangle two or more attributes instead of solely acquiring a representation. Therefore, the explicit models are formalized as a supervised learning problem for disentangling the hidden factors of variation. Alternatively, they infer the available information to enforce disentangling the hidden attributes, such as rotation [12].

In contrast, implicit models are designed to acquire a representation for a given dataset without relying on any of the attribute labels y_i or without enforcing any information onto the model. However, further examination of different dimensions of the representation reveals that they disentangle hidden attributes of the dataset. In the following sections, we will examine these two categories in detail.

2.2.1 Explicit Disentanglement Models

Explicit disentanglement models aim to learn a representation such that two or more dataset attributes are explicitly disentangled. These models can either directly use the attributes labels $\mathbf{y^{(i)}}$ in a supervised fashion [61], or embed the attribute information into the model through self-supervised approaches [8, 13, 42].

Supervised models require the availability of attribute labels $\mathbf{y^{(i)}}$ to guide the model into learning predefined dataset attributes. For instance, Tran et al. [61] proposed a GAN-based model, called DR-GAN, for pose invariant face identity recognition and face synthesis. In their approach, they have access to ground truth values for identity and pose attributes. Unlike conventional GANs, the discriminator tries to classify the identity and the pose of both the original and synthesized images. Since this model uses attribute labels for pose and identity, we categorize it as a supervised explicit model in terms of disentanglement.

Self-supervised models, on the other hand, does not use attribute labels $\mathbf{y^{(i)}}$ directly [17, 33, 59]. However, these models enforce the disentanglement of two or more attributes through inferred information. For example, Kulkarni et al. generate mini-batches, such that the samples in a batch diverge only in a single attribute, and the rest of the attributes are shared across the batch. The model is forced to discriminate the distinct attributes in the representation space. Similarly, Mathieu et al. [43] uses triplets of examples in a training iteration, namely $\mathbf{x}^{(1)}$, and $\mathbf{x}^{(2)}$ having the same label, and $\mathbf{x}^{(2)}$ having any other label $\mathbf{y}_j^{(1)} = \mathbf{y}_j^{(2)} \neq \mathbf{y}_j^{(3)}$. Swapping these unobserved variables among the samples does not change the observed classification. However, it alters the unobserved factor.

Unsupervised models can also achieve explicit disentanglement without access to the attribute information of the dataset. For example, Detlefsen et al. [12] propose a VAE-based approach to model the appearance and perspective of given images. They train two parallel branches, where they learn appearance and perspective, respectively. The output of the perspective branch serves as an affine transformer on the appearance layer, modifying the perspective of the objects in an image. Despite lacking access to attribute labels of perspective or appearance, imposing an affine transformation characterizes this model as an explicit disentanglement model. Similar to supervised or self-supervised explicit models, unsupervised explicit models also demonstrate the capacity to disentangle only a limited number of attributes since they require a specific architecture per attribute.

Explicit disentanglement has also been utilized to solve tasks in spatio-temporal domain, such as video frame generation or prediction [11, 25, 36, 64, 68]. The core assumption behind these models is that the content of a video scene is the same across different frames, and only the positions of the objects change [32]. This foundational principle guides explicit models into manipulating attributes dynamically.

2.2.2 Implicit Models

As we previously stated, implicit models acquire a representation for a given dataset without relying on any factor labels, denoted as $\mathbf{y}^{(i)}$. The models we examine in this subgroup are the Variational Auto-Enocoder models. Considering the fact that our

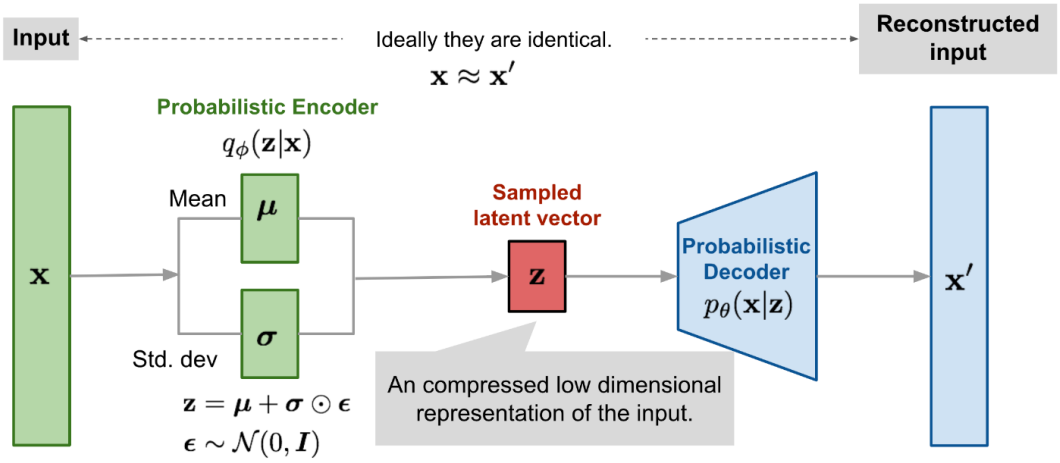


Figure 2.3: Encoder-Decoder architecture of VAE. [Figure Source: [63]]

suggested models are based on VAE models, we examine this group of models in detail in the following sections.

2.3 Variational Auto-Encoders and its Variants

Variational Auto-Encoders (VAEs) [31] are widely used for disentangled representation learning due to their relative simplicity and disentangling ability of the hidden attributes. VAEs are generative neural network architectures that encode and decode a disentangled representation for a given data sample. Employing an encoder-decoder architecture, VAEs estimate a Gaussian distribution of the sources of variation in the representation space. Figure 2.3 shows the encoder-decoder architecture of the original VAE, suggested by Weng et al. [63].

VAEs aim to find a representation $\mathbf{z}$ for an image sample $\mathbf{x}$ with an encoder-decoder architecture. The output of the encoder is called representation, $\mathbf{z}$. The decoder uses this representation $\mathbf{z}$ to recover the image sample $\mathbf{x}'$.

Formally, given a dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{W}, \mathcal{Y}\}$, the representation $\mathbf{z}$ is approximated by a posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$. This distribution is parametrized by a neural network called the encoder. Simultaneously, the decoder parametrizes the likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ and reconstructs the image from the representation $\mathbf{z}$. Instead of maximizing the log-likelihood, VAEs optimize ***E**vidence **L**ower **BO**und (ELBO)* as a loss function defined

as:

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = E_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] - D_{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})\Big), \tag{2.1}$$

where the first term enforces the reconstruction of the images, compelling the generated images to match the original image. The second term acts as a regularizer, compelling the learned distributions to approach a predefined distribution $p(\mathbf{z})$, creating an information bottleneck. This predefined distribution $p(\mathbf{z})$ is generally selected to be a Gaussian distribution with zero mean and unit variance, $p(\mathbf{z}) = \mathcal{N}(0, I)$.

In essence, assuming the size of the representation vector is L, an encoder network estimates the mean $\mu \in \mathbb{R}^L$ and variance $\sigma \in \mathbb{R}^L$ of a Gaussian distribution $\mathcal{N}(\mu_j, \sigma_j)$ for each dimension j in the representation. The representation $\mathbf{z}$ is then sampled from these Gaussian source distributions as $\mathbf{z} = \mu + \epsilon\sigma$. The random variable ϵ is represented by the normal distribution $\mathcal{N}(0, I)$. A decoder is then used to reconstruct the estimated images as $\bar{\mathbf{x}}$ given the representation $\mathbf{z}$. ELBO is employed at the training step to optimize the encoder and decoder.

It is shown that VAEs have the ability to learn a representation that is disentangled to a certain degree, making them an appealing approach for improving the generalization capacity and interpretability of the model in an unsupervised setting [62]. Many studies explore the generalization properties of VAEs and propose extensions to the above VAE architecture, called vanilla VAE [22, 29].

2.3.1 Beta Variational Auto-Encoders (β-VAEs)

β-VAE is one of the pioneering extensions to vanilla VAE [31]. Higgins et al. argued that VAEs do not provide a sufficient degree of disentanglement on the hidden attributes [22]. In order to increase the disentanglement capabilities of the model, Higgins et al. proposed β-VAE, where they augment the original loss function of VAEs (Equation 2.1) with a hyperparameter β on the KL divergence term to encourage disentanglement:

$$L_{\beta-VAE}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]] - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \tag{2.2}$$

When the hyperparameter $\beta > 1$, the model emphasizes the importance of the KL term relative to reconstruction loss, promoting stronger independence among the hid-

den attributes of the learned representation [22]. Notice that when the hyperparameter $\beta = 1$, the model reduces to the original VAE.

Note that the reconstruction loss and KL divergence term minimize two numerically incompatible properties of the model. While the reconstruction loss minimizes the squared error between the data and its estimated version, the KL divergence minimizes the difference between two distributions namely $p_\phi(z|x)$ and $p(z)$. The discrepancy between these two terms results in an imbalance during the minimization of the overall loss function. β-VAE also addresses this issue by weighting the second term to compensate for the discrepancy between the two terms of the loss functions. However, the weight parameter β can only balance the ranges of the reconstruction loss and KL divergence. Furthermore, the convergence speed of the overall loss function mostly is not proportional to that of the two individual terms. We revisit these problems and analyze the incompatibilities between the two terms of the overall loss function, by thorough experiments in Section 4.2.

Although β-VAE is used to learn disentangled representations in many applications [29], it has certain limitations, which we discuss in the next section.

2.3.2 Limitations of β-VAE

β-VAE demonstrates successful performance across various disentanglement tasks [29]. Nevertheless, it has two significant drawbacks, as listed below:

1. β-VAE tends to increase reconstruction loss for $\beta > 1$ in order to provide stronger disentanglement on the representation space [9, 29]. It is observed that β-VAE reduces the quality of reconstruction, due to the trade-off between the reconstruction loss term and the KL divergence term of the cost function [9, 29].

2. β-VAE increases the computational complexity of the model. Adding a new hyperparameter enlarges the optimization space, which results in an increase in the complexity of the search space [54]. Moreover, despite the fact that Higgins et al. restrict the range of this parameter to $\beta > 1$ [54], several studies show that this range is not universally optimal for all datasets (i.e. optimal

disentanglement is achieved when $\beta < 1$ for some datasets) [18]. Extending the range of β from numbers greater than one to all positive numbers (i.e. $\beta > 0$, instead of $\beta > 1$), further complicates an already expensive and complex problem of hyperparameter tuning.

Although the aforementioned problems limit the performances obtained from the β-VAE model, there are possible solutions in the literature that address these issues, as described in the following subsections.

2.3.3 Solutions Addressing Limitations of β-VAE

In this section, we describe the methods for mitigating the limitations of β-VAE.

2.3.3.1 Solutions for Increasing the Reconstruction Quality

As we stated in the previous section, β-VAE is observed to potentially reduce the quality of reconstruction [9, 29]. Several solutions are proposed to overcome this problem [6, 9, 29]. For example, Burgess et al. propose a control mechanism on the "information capacity" of the model to regain the models reconstruction abilities [6]. They argue that by gradually increasing the model's capacity from zero, they can increase the disentanglement ability of the model while conserving the reconstruction loss. They augment the original loss function of β-VAE in Equation 2.2 as follows:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]] - \gamma|D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - C|, \tag{2.3}$$

where C denotes the information capacity, which is gradually increased to a predefined, empirically selected value to achieve the desired goal of increased reconstruction quality. However, this implementation introduces a new hyperparameter C to the problem.

Kim et al. also argue that β-VAE decreases the quality of reconstruction [29]. They reformulate the KL term in the original loss function of β-VAE (See Equation 2.2) as follows:

$$D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) = MI(\mathbf{x}, \mathbf{z}) + D_{KL}(q(\mathbf{z})||p(\mathbf{z})), \tag{2.4}$$

where $MI(\mathbf{x}, \mathbf{z})$ is the mutual information (MI) between data point $\mathbf{x}$ and its latent representation $\mathbf{z}$. Kim et al. show that increasing β in the β-VAE model results in decreased MI between data and its representation, leading to an increase in the reconstruction error. In order to overcome this problem, they propose a new model, called Factor-VAE, in which they introduce a discriminator to the existing architecture. The main purpose of the discriminator is to modulate the trade-off between disentanglement and reconstruction. The overall cost function increases the independence among latent variables without affecting the minimization of mutual information. This objective is formalized by the following loss function;

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}[\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] - \gamma D_{KL}(q(\mathbf{z})||\bar{q}(\mathbf{z})), \quad (2.5)$$

where $\bar{q}(\mathbf{z}) = \prod_{j=1}^{d} q(\mathrm{z}_j)$ increases the independence among the dimensions of the latent representation.

Similarly, Chen et al. investigate the source of success in β-VAE and decompose the Evidence Lower BOund (ELBO) loss term into three parts; namely index code mutual information, total correlation, and dimension-wise KL divergence [9]. These terms correspond to the mutual information between data and latent representation, dependence among variables, and Kullback-Liebler divergence of the terms from the priors, respectively. They argue that the success of β-VAE in terms of disentanglement stems from the total correlation term embedded in the ELBO. Then, they propose the model β-TCVAE, an extension to β-VAE where they introduce weights to each of these three terms. The cost function is the same as [29], but the two total correlation terms are estimated by different methods.

A good resource that compares different VAE-based models is the study by Locatello et al. [41]. They analyze the performances of the aforementioned models in a compelling scenario. They conduct extensive experiments for measuring the performance of various models, executing 12,000 models and providing comparative results on six different disentanglement metrics. Their findings emphasize that hyperparameter selection of the model is more crucial than model selection in terms of disentanglement.

2.3.3.2 Solutions for Controlling the Model Complexity

As we mentioned earlier in Section 2.3.2, β-VAE increases the complexity of the problem by introducing a new hyperparameter β to the optimization space. In fact, this issue extends to other VAE-based models, where the introduction of new hyperparameters presents a similar challenge. Locatello et al. [41] address this problem in their review by emphasizing the difficulty of the parameter tuning problems of β-VAE. Therefore, hyperparameter tuning is an important problem in the field of disentangled representation learning. Various studies explore automatic hyperparameter tuning in VAE-based models, with a specific focus on balancing the reconstruction loss and KL divergence terms in the loss function [56, 57, 58].

Shao et al. propose the ControlVAE algorithm with the aim of automatically tuning the hyperparameters, specifically focusing on dynamically adjusting the β in the β-VAE algorithm [58, 56]. They employ a Proportional–Integral controller (PI Controller), a variation of PID controller [1], to automatically update β at each time step, rather than using a static β. Shao et al. update the loss function in Equation 2.2 as follows:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]] - \beta(t) D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \tag{2.6}$$

In order to automatically tune the hyperparameter $\beta(t)$, they assume that the second term of this loss function should converge a desired KL value, v_{kl}. At each iteration, they compute the error between this desired value and the current value of the KL Divergence as $e(t) = v_{kl} - v_{kl}(t)$, where $v_{kl}(t)$ is the second term in the loss function (i.e. $v_{kl} = D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$). The parameter $\beta(t)$ is then updated using PI control algorithm to minimize this error $e(t)$, as shown in the following equation:

$$\beta(t) = \frac{K_p}{1 + \exp(e(t))} - K_i \sum_{j=0}^{t} e(j) + \beta_{min}, \tag{2.7}$$

where K_p and K_i are constants, and β_{min} is the lower limit of $\beta(t)$. Shao et al. further refine the ControlVAE model and proposed DynamicVAE [57]. DynamicVAE utilizes a similar PI Controller. However, it iteratively reduces $\beta(t)$ from a large value rather than increasing it. Moreover, the error between the desired KL divergence v_{kl} and its current value $v_{kl}(t)$ is computed as a moving average, taking previous values of the

KL Divergence into consideration, as follows:

$$e(t) = v_{kl} - \sum_{i=t-T}^{T} \alpha_i v_{kl}(i). \tag{2.8}$$

These updates enable a smoother change in the KL term. Although ControlVAE and DynamicVAE can efficiently learn $\beta(t)$, they both introduce several new parameters to optimize, such as the desired value of the KL divergence v_{kl}, the initial value of β, and the constants of the PID controller, K_p and K_i.

Moreover, Rybkin et al. propose another model called σ-VAE, which learns the variance of the decoder with network parameters [54]. Their algorithm learns the weight of the reconstruction loss instead of the KL divergence, as follows:

$$\mathcal{L} = \ln(\sigma) + \frac{1}{2\sigma^2}\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]] - D_{KL}(q_\phi(z|x)||p(z)). \tag{2.9}$$

The parameter σ is learned along with the network parameters. Although they did not analyze σ-VAE as a disentanglement model and focused only on the reconstruction quality of the model, their work is similar to ours since they learn the relative weight of the reconstruction term.

2.4 Measuring the Amount of Disentanglement

The concept of disentanglement has a wide range of properties, hence it is not quantifiable with a scalar metric. Measuring the amount of disentanglement is a challenging problem, and unfortunately, there is no consensus on how to quantify the performances of the models. The well-known measures include β-VAE [22], and FactorVAE [29]. Fairness can also be used to analyze the disentanglement [40]. There are also several reviews to compare and analyze the available measures [7, 41]. In Section 5.1 we explain these measures in detail to provide a comprehensive comparison with the measure we propose.

In their review of disentanglement metrics, Carbonneau et al. [7] argue that a disentangled model should be evaluated concerning three major properties. These properties and the corresponding measures can be listed as follows:

- **Explicitness** measures models ability to recover the input from the representation. The measures that focus on explicitness include the [53].
- **Compactness** requires a small portion of the representation (ideally one dimension) to represent a single attribute. A popular measure in this category is the *SAP score* [10, 34].
- **Modularity** requires a change of a factor to affect only a single dimension of the representation. Modularity measures include [22, 29].
- The measures available in the literature can measure two or more of the above properties, in which case the measure is called holistic [7]. We have selected *IRS* [60], that measures modularity and explicitness.

Although there are several measures available in the literature, they all measure different properties of disentanglement by their definitions. Thus, they do not provide consistent results across different datasets. In Section 5.1 we will discuss the measures we have considered in this thesis.

2.5 Chapter Summary

In this chapter, we provide a detailed review of the models of disentangled representation, available in the literature. We first provide a formal definition of the term disentanglement. Following this definition, we examine the disentanglement models from the literature under two categories, namely explicit and implicit disentanglement models. Under explicit models, we study several supervised, self-supervised and unsupervised models. Within the implicit models category, we focus on VAE-based models. In this thesis, since we propose two VAE-based models, we elaborate on the limitations and possible solutions of the models in this category. Finally, we provide a brief introduction to disentanglement measures. We explain the available measures for quantifying the disentangling models in detail in Chapter 5.

CHAPTER 3

EXPERIMENTAL SETUP

This chapter introduces the datasets and experimental settings, used to analyse and validate the proposed models. Throughout the thesis, we compare several disentangled representation learning models (DRLs) on popular datasets. We use equivalent architectures and optimize models on the same hyperparameter space. This chapter provides a detailed explanation of the experimental setup so that the experiments can be reproduced using the explained setup. In the first section, we introduce the datasets with examples. In the second section, we briefly remind the methods we use as a baseline for our studies. The following sections describe the model architecture, hyperparameter spaces and model selection methodologies.

3.1 Disentanglement Datasets

There are several disentanglement datasets with available attribute labels in the literature [2, 49, 51]. We conduct our experiments on four different disentanglement datasets listed below (See Figure 3.1 for sample images):

dSprites [44] is a 2D shapes dataset with 700K 64x64 images containing white 2D shape images (heart, square, ellipse) on black background. There are five labelled attributes to disentangle: shape, scale, orientation, and X, Y positions of the object.

MPI3D-complex [20] consists of four real-world objects moving on a robotic arm leading to 460K color images with the size 64×64. There are seven labelled attributes to disentangle: color shape, size, camera height, background color, and horizontal and

vertical axes. We will refer to this dataset as MPI3D.

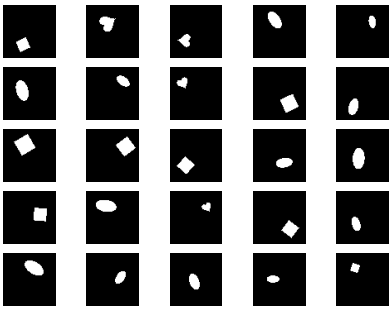
Falcor3D [47] is a dataset of images of a living room containing with different lighting conditions containing 233K images with the size 128×128. There are seven labelled attributes to disentangle: lighting intensity, directions x, y, and z of lighting, and x, y, and z coordinates of the camera position.

Isaac3D [47] is a dataset containing a robotic arm holding an object in a kitchen. Lighting conditions, camera position, and the position of the arm are altered. The dataset contains 737K images with size 128×128. There are nine labelled attributes to disentangle: Objects shape, scale, and color, wall color, camera height, robotic arms' x and y positions, lighting intensity and direction.

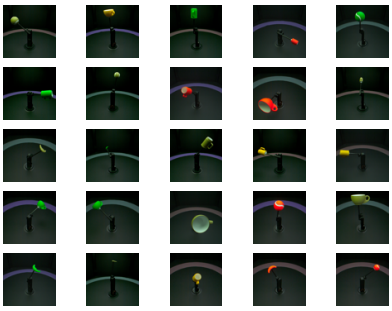
CelebA [39] consists of 202K images of celebrity faces labelled with facial attributes. There are 40 facial attribute labels, such as baldness, wearing eyeglasses, and pale skin. we provide qualitative results for L-VAE on the CelebA dataset. We cropped the background from the images and downscaled the dataset to size 128×128 following [9]. Furthermore, we have used 12% of the dataset to train the L-VAE model for simplicity.

3.2 Experimental Settings

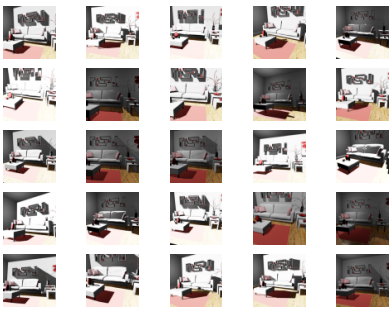
Through the thesis, we introduce two novel DRL models, namely L-VAE and mdL-VAE (See Chapters 6, and 7 for details). In this section, first, we explain the methods with which we compare our proposed methods L-VAE and mdL-VAE. Next, we describe the model details and the performance measures employed in our experiments. Recall that we perform our experiments with four different datasets, dSprites, MPI3D, Falcor3D, and Isaac3D, which are explained in Section 3.1.



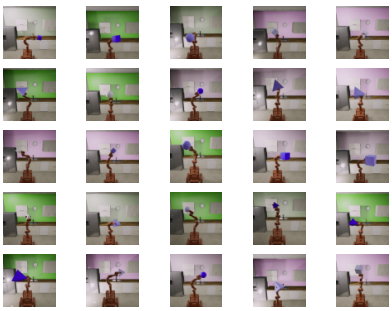
(a) dSprites [44]



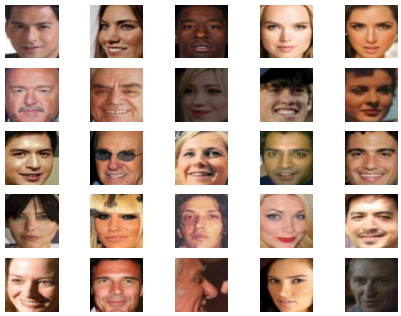
(b) MPI3D-complex [20]



(c) Falcor3D [47]



(d) Isaac3D [47]



(e) CelebA [39]

Figure 3.1: Sample images from the dataset we have used in our experiments.

3.2.1 Disentanglement Methods for Comparison

We compare the proposed L-VAE and mdL-VAE with five well-known disentanglement methods, which are explained in Chapter 2 in detail. The list of models are as follows:

1. Variational Autoencoder (VAE) [31], which optimizes reconstruction loss and KL divergence with equal weights.

2. β Variational Autoencoder (β-VAE) [22], which utilizes an empirically tuned hyperparameter β to weight KL divergence.

3. ControlVAE [56, 58], which learns the β parameter, based on a PID control algorithm at each iteration.

4. DynamicVAE [57], which is a slightly modified version of ControlVAE.

5. σ-VAE [54], which can be considered as an instantiation of our model L-VAE. It learns the weight of the reconstruction loss, whereas our model L-VAE learns the weights of both reconstruction loss and KL divergence.

3.2.2 Architecture and Training Details

For a fair comparison with prior models, we follow the architecture and training details as suggested by Burgess et al. [6]. More specifically, we perform our experiments with a Multi-Layer Perceptron (MLP) Encoder-Decoder architecture for the dSprites dataset and a CNN Encoder-Decoder architecture for the MPI3D, Falcor3D, and Isaac3D datasets (see Table 3.1 for the architecture details). The architecture uses the ReLU activation function on all hidden layers and the Sigmoid activation function at the decoder output. We use Adam optimizer [30] with $\beta_1, \beta_2 = (0.9, 0.999)$, and $\epsilon = 10^{-8}$. Mean Square Error (MSE) is used as the reconstruction loss following the work of Burgess et al. [6].

Table 3.1: The encoder and decoder architectures we used in our experiments. For the fully connected (FC) layers, the number of output features is given in parentheses. For convolutional layers (Conv) and transposed convolutional layers (ConvT), the number of input and output features is also given in parentheses.

Model	Model Details
MLP Encoder	FC(1200), FC(1200),FC(2xLatent_dim)
MLP Decoder	FC(1200), FC(1200),FC(HxWxC)
CNN Encoder	Conv(32x4x4, stride=2), Conv(32x4x4, stride=2), Conv(64x4x4, stride=2), Conv(64x4x4, stride=2), Conv(32x4x4, stride=1),FC(2xLatent_dim)
CNN Decoder	FC(256), ConvT(64x4x4, stride=2), ConvT(64x4x4, stride=2),ConvT(32x4x4, stride=2), ConvT(32x4x4, stride=2),ConvT(3x4x4, stride=2), ConvT(3x4x4, stride=2)

We empirically tune the hyperparameters, namely, batch size, learning rate, and iteration count for all of the models using cross validation. We selected the dynamic ranges experimentally using the Weights and Biases (WandB) tool [5]. We plan to enlarge these ranges in the future. The current dynamic ranges that we use for cross validation are as follows;

- We tune **Batch size** within the set {64, 128, 256} for all models.
- We tune **Learning rate** within the set {1e-04, 1e-05} for all models.
- We tune **Number of iterations** within the set {5,10,15,20,25,30}$\times 10^4$ for all models.
- For β-VAE model [22], we tune β within the set {2,4}.
- For ControlVAE (following the work of [58]):
  - we set the desired KL value to 18,
  - we set K_p to 0.0.01,

  - we set K_i to -0.001,
  - we initialize β with 0.

- For DynamicVAE (following the work of [57]):
  - we set the desired KL value to 18,
  - we set K_p to 0.0.01,
  - we set K_i to -0.005,
  - we initialize β with 150.

Configuring experiments with this set of hyperparameters for cross-validation, VAE, L-VAE, and mdL-VAE experiments are carried out independently for 36 models (considering 3 different values for batch size, 2 different values for learning rate, and 6 different values for number of iterations). For β-VAE, β is selected from {2,4} (higher β results in higher reconstruction losses for the datasets we have experimented on (see Section 4.2 for details), which leads to 72 independent models trained for β-VAE.

In Table 3.1, it is shown that encoders have different numbers of nodes in the last layer, which is denoted as *latent_dim*. This number represents the number of dimensions we use to represent images. Since each dataset has different characteristics, we set this number for each dataset t be equal to the number of attributes. More specifically, we set the latent dimension size to five for the dSprites dataset, seven for the MPI3D and Falcor3D datasets, and nine for the Isaac3D dataset following the number of labelled attributes provided with the datasets [20, 44, 47] (See Section 3.1 for the details on datasets). All the σ_i values introduced in our suggested model (See Chapters 6 and 7) are initialized with one (1) for L-VAE and mdL-VAE experiments.

3.2.3 Critique on Model Selection

As stated in Section 3.2.2, we have conducted our experiments to cover a relatively large number of hyperparameter set (i.e., selection of batch size, learning rate, and iteration count). There are two significant problems in determining the best set of hyperparameters of the models in the validation set: Firstly, finding a balance between the reconstruction loss and KL divergence in the overall loss function (Equation 6.2)

is still an open research issue. Secondly, choosing a suitable measure for quantifying the amount of disentanglement brings a strong bias toward the selected measure. Therefore, empirically finding an "optimal" hyperparameter set is not a well-defined problem in disentangled representation learning. For the sake of being fair in our comparisons, we empirically selected the hyperparameters based on the highest CbD score, which is explained in Chapter 5. We use the best scores achieved on the validation set for the hyperparameter selection for all methods.

3.3 Chapter Summary

In this chapter, we provide details on the experimental setup we use to compare different disentanglement models. First we explain the datasets, and we provide architecture and hyperparameter details. In all of our experiments we use cross-validation to select hyperparameters. However, the model selection is another important problem in disentanglement literature. We use the measure that we propose (See Chapter 5) for model selection.

CHAPTER 4

AN ANALYSIS ON β VARIATIONAL AUTO-ENCODER

β-VAE is one of the pioneering extensions of vanilla VAE, generating a representation for the datasets given in [6, 22]. Employing an encoder-decoder architecture, β-VAEs estimate a Gaussian distribution of each factor of variation in the latent space. Although β-VAE is commonly used across several representation learning tasks, it has two major drawbacks, as discussed in Section 2.3.2. Specifically, β-VAE is known to decrease reconstruction quality while increasing model complexity through the hyperparameter β. In order to substantiate these identified limitations and form a basis for our studies, we carry out several experimental analyses, focusing on the effect of the hyperparameter β on the disentanglement and reconstruction abilities of the β-VAE model. In this section, we discuss the key findings from our analysis.

4.1 Overview of the Analysis

We start with an analysis of the effect of the hyperparameter β on the learning dynamics of β-VAE and provide a basis for the thesis. First, we investigate the learning dynamics between reconstruction and KL-divergence terms of the β-VAE loss function given below;

$$L_{\beta-VAE}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]] - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \quad (4.1)$$

We analyze the relative changes of these terms during the training phase for different values of the hyperparameter β.

In order to provide a comprehensive analysis, we perform experiments on four datasets introduced in Section 3.1. To minimize variables in this controlled experiment, we

used one model architecture across all datasets. In this architecture, the output of the encoder is called the representation. The number of dimensions of this representation is dataset-specific, where it is equal to the number of labelled attributes within a dataset. For each dataset, we keep the same dimension across all the experiments (See Section 3.2.2 for details). However, to get the best results in all settings, we tuned the rest of the hyperparameters, specifically the learning rate, batch size, and number of iterations through cross validation. Our main goal in this analysis is to seek answers to the following questions:

- **Question 1:** How do the reconstruction loss and KL Divergence terms behave as we increase β in Equation 4.1, assuming we select $\beta \geq 1$ as suggested by Higgins et al. [22]?
- **Question 2:** How does the disentanglement score of the representation change as we increase β?

4.2 The Effect of the Hyperparameter β on the Learning Dynamics

In this section, we perform a set of experiments by varying β to find answers to the aforementioned questions. The analysis and the resulting observations are described below:

Observation 1: The hyperparameter β helps balance the ranges of the reconstruction loss and the KL divergence. We start our analysis by investigating the relative changes of the two terms of the loss function: Reconstruction loss and KL Divergence terms. Figure 4.1a compares the values of these two terms during the training of a VAE, i.e., β-VAE with $\beta = 1$ for the dSprites dataset. We observe that in the early stages of the training, the values of the reconstruction loss are relatively higher than the KL divergence values. Note that, in the first iterations, reconstruction loss starts from a value that is larger than 1000. This value drops to the range of [10,50] in the first 50000 iterations. However, KL divergence initially starts from [0, 1] and increases in the range of [0, 20]. Figure 4.1a does not show the early iterations for simplicity. Consequently, the reconstruction loss dominates the loss function, especially in the early stages of the training. Therefore, the contribution of

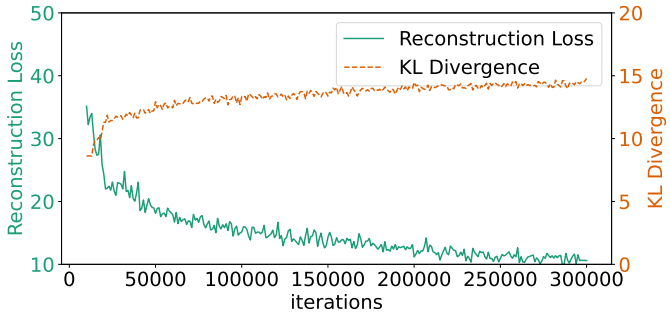
the KL divergence term to the overall loss remains relatively small. Since the reconstruction loss dominates the overall loss, minimizing the KL divergence in parallel to the reconstruction loss becomes a challenge. This imbalance between the values of reconstruction loss and the KL divergence term reduces the effect of KL term in the optimization process of the cost function, which disables the disentangling properties of the Variational Auto-Encoder. The fact is clearly observed in Figure 4.1a. The value of the KL term slightly increases during the iterations. For this particular dataset, when $\beta = 1$, KL term becomes almost redundant in the optimization process.

We believe that the leading cause of VAEs failure (when β = 1) to disentangle the learned representation is the discrepancy between the values of the reconstruction loss and KL divergence term.

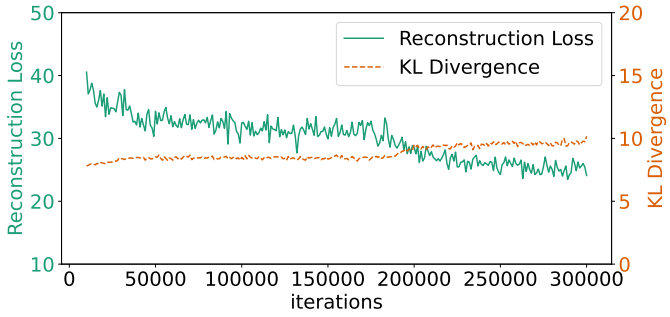
Increasing the value of β increases the importance of the KL Divergence term in the overall loss function. Figures 4.1b and 4.1c show the behaviour of the same losses during the training phase for $\beta = 2$ and $\beta = 4$ respectively. These figures suggest that increasing the hyperparameter β compensates for the discrepancy between the values of the two terms and dampens the increase of the KL term.

Figure 4.2 shows the results of the same experiment with the MPI3D dataset. Similar to the results we obtain with the dSprites dataset, when we compare the plots of the reconstruction loss and KL divergence for different values of β with the MPI3D dataset, we can argue that increasing β compensates for the difference between the reconstruction and KL divergence terms and reduces the overall increase of the KL term.

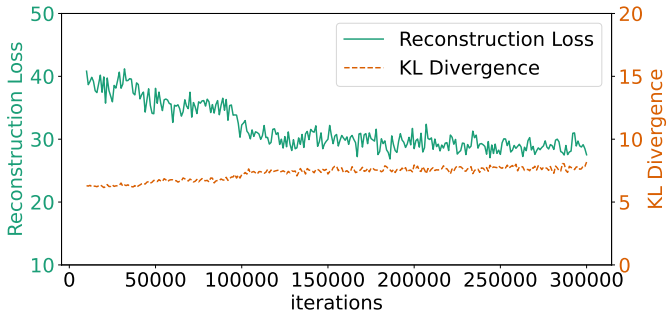
Observation 2: Increasing the hyper-parameter β lowers reconstruction quality. In the same experiment set-up, we compare reconstruction losses as we increase β. Figure 4.3a plots the reconstruction losses for different β values on the dSprites dataset. The figure suggests that weighing the KL term of the loss function by $\beta \geq 1$ results in an increase in the reconstruction loss. When we focus on the later stages of the learning curves, the reconstruction loss converges to a higher value, when $\beta \geq 1$. This empirical finding suggests that increasing β lowers the reconstruction quality, supporting previous findings reported by [9, 29].



(a) $\beta = 1$

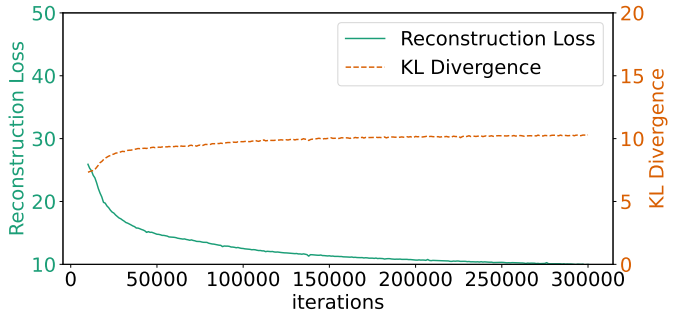


(b) $\beta = 2$

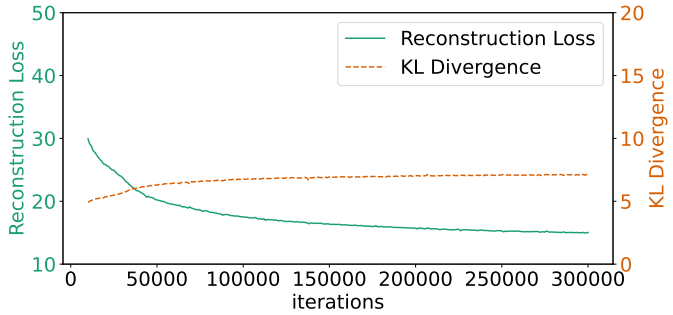


(c) $\beta = 4$

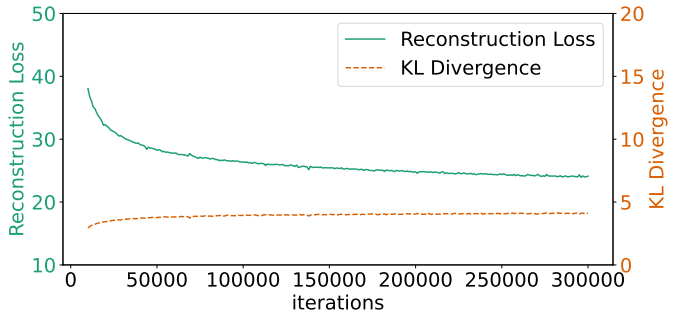
Figure 4.1: Training curves for reconstruction loss and KL-Divergence for (a) VAEö (b) β-VAE with $\beta = 2$ and (c) β-VAE with $\beta = 4$ for dSprites dataset. (a) KL divergence increases during training due to the difference between the values of two loss terms. (b-c) β-VAE controls the difference between the two terms. When $\beta > 1$, we observe KL divergence converges to a lower value. Although the KL-divergence term does not increase as much as we observe in (a), it still does not decrease as expected.



(a) $\beta = 1$



(b) $\beta = 2$

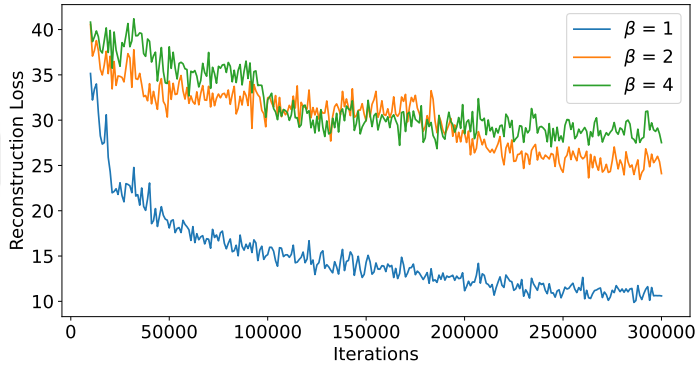


(c) $\beta = 4$

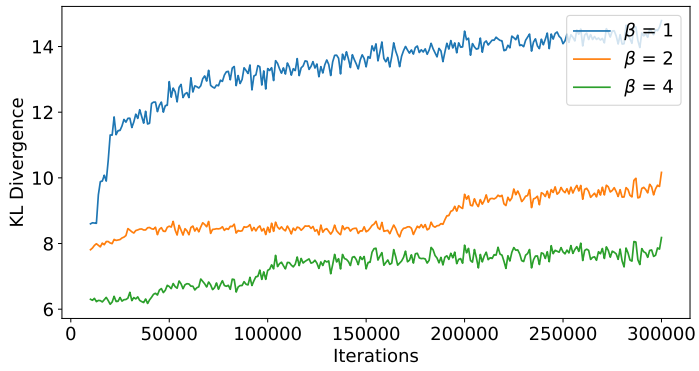
Figure 4.2: Training curves for reconstruction loss and KL-Divergence for (a) VAE and (b) β-VAE with $\beta = 2$ for MPI3D dataset. (a) KL divergence increases during training due to the difference between the two loss terms. (b) β-VAE controls the difference between the two terms and the increase in the second term. Although the KL-divergence term does not increase as much as we observe in (a), it still does not decrease as expected.

Moreover, we can also report that increasing β results in a significant decrease in the KL term. Figure 4.3b shows the behaviour of the KL term. Note that as we increase β value, KL term converges to a relatively lower values, which could be the main cause of increased disentanglement.

We repeat the same experiment on the MPI3D dataset. Figure 4.4 shows the results of this experiment. Figure 4.4a shows the convergence of reconstruction losses for β =1,2 and 4. When we increase β, the reconstruction loss converges to a relatively higher value. Similarly, Figure 4.4b indicates that when $\beta \geq 1$, KL divergence values converge to a relatively lower value.



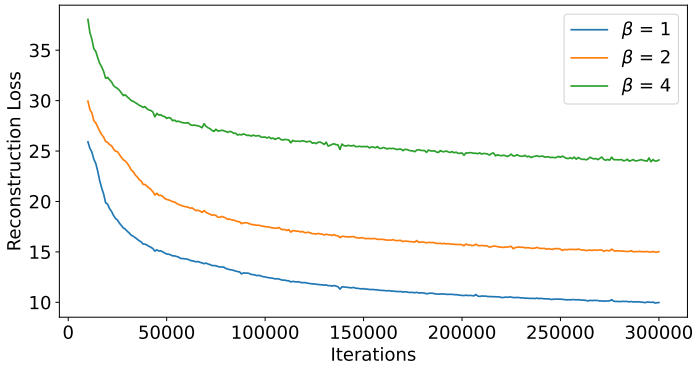
(a)



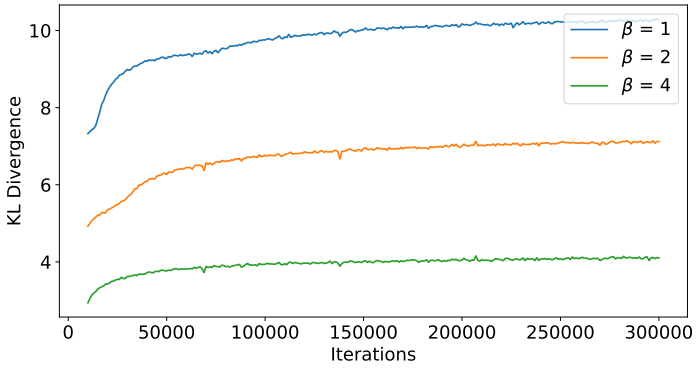
(b)

Figure 4.3: The changes for (a) reconstruction loss and(b) KL Divergence with increased β in dSprites dataset.

We further investigate the effect of β on the relationship between disentanglement and reconstruction. We compare disentanglement properties on four datasets; dSprites



(a)

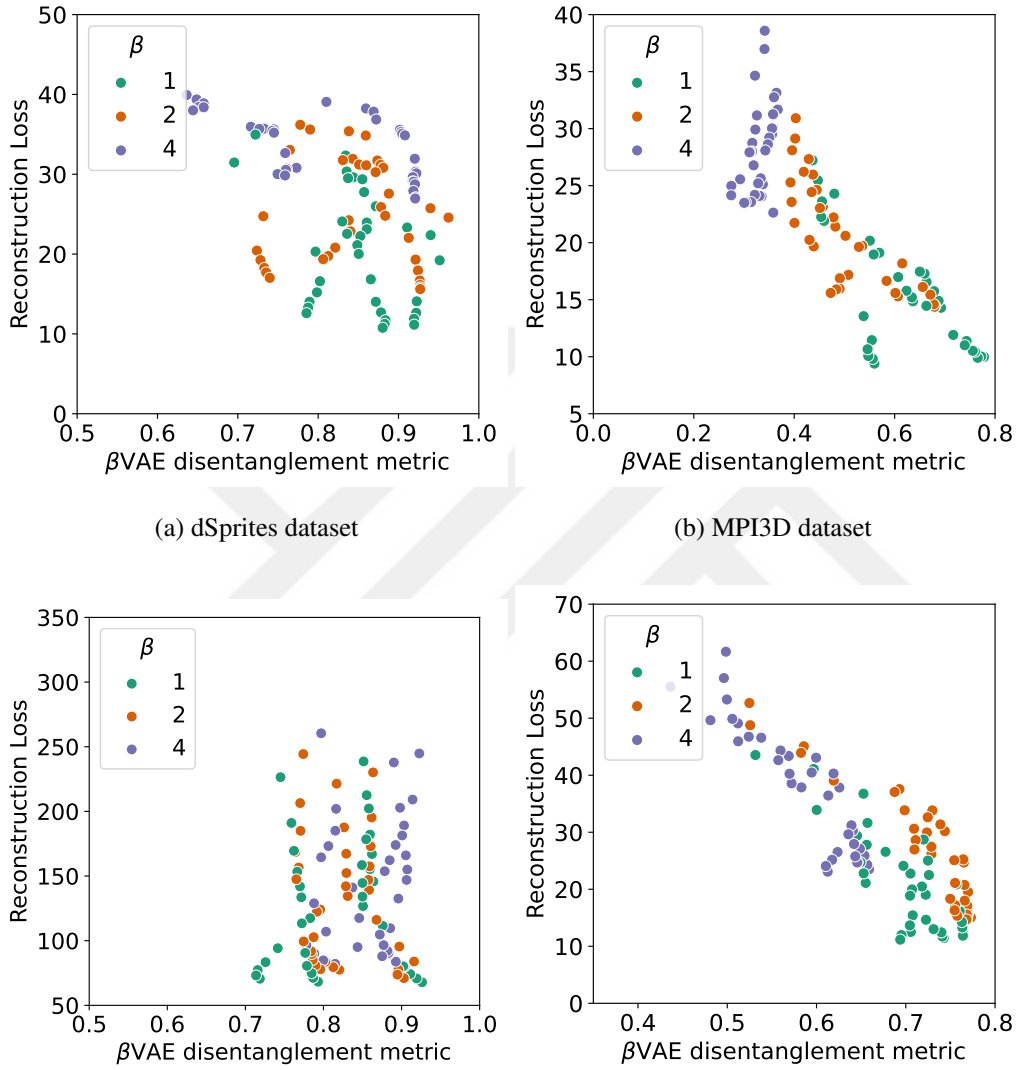


(b)

Figure 4.4: The changes for (a) reconstruction loss and (b) KL Divergence with increased β in MPI3D dataset.

[44], MPI3D [20], Falcor3D [47] and Isaac3D [47] (details in Section 3.1). In this experiment, we have thoroughly investigated the relationship between the reconstruction and disentanglement terms by systematically varying the value of the hyperparameter β across the hyperparameters of the network. Figure 4.5 shows the results of this analysis. We analyze the results of several runs covering a large hyperparameter space consisting of varying learning rates, batch sizes, and iteration counts (details in Section 3.2.2). The x and y-axes show the reconstruction loss and β-VAE disentanglement measure, respectively. Note that higher scores in β-VAE measure indicate better disentanglement. Points are color-coded based on the value of β. Each point represents a model trained with a different hyperparameter setup, with learning rate, batch size, and number of iterations (see Section 3.2.2 for details). Notice that there

are several points with the same β value, which shows models trained with different hyperparameters. In order to visually observe the effect of different β values on the reconstruction loss and disentanglement, the values of other parameters are not specified in Figure 4.5.



(a) dSprites dataset

(b) MPI3D dataset

(c) Falcor3D dataset

(d) Isaac3D dataset

Figure 4.5: Reconstruction loss versus β-VAE disentanglement metric across different β hyperparameters for (a) dSprites, (b) MPI3D, (c) Falcor3D, and (d) Isaac3D datasets. Each point represents a model trained with a set of hyperparameter setup (altered learning rate, number of iterations, and batch size). Points are color-coded based on the value of β.

Observation 3: Disentanglement also depends on hyperparameters of the model other than the hyperparameter β. Figure 4.5a shows that the disentanglement vs reconstruction dynamic highly depends on the selection of the hyperparameters, such as the learning rate or batch size for the dSprites dataset. When we focus on a specific hyperparameter β, a specific color in this figure, we can say that different parameters lead to different reconstruction qualities or different degrees of disentanglement. For example, when $\beta = 4$, which is shown in blue, we can observe reconstruction losses in the range [25,45]. Similarly, disentanglement scores range [0.6,1]. Therefore we can clearly state that selecting a hyperparameter set other than β is very crucial. This phenomenon also occurs on other datasets, see Figures 4.5b, 4.5c, and 4.5d . Thus, finding the optimal hyperparameter set which minimizes the reconstruction loss is an extremely exhaustive and difficult problem. We also observe that increasing the β parameter from 1 to 4 gives better disentanglement values with respect to β-VAE measure with the cost of higher reconstruction loss. These observations support the arguments of [9], who claimed β-VAE increases the reconstruction loss. We observe similar behavior for the Falcor3D dataset (See Figure 4.5c).

Observation 4: $\beta < 1$ can surprisingly increase disentanglement. Further analysis on the MPI3D dataset reveals more insight into the effect of β on disentanglement. We have conducted the same experiment on this dataset (see Figure 4.5b). The results on this dataset complement the dSprites experiments. Higgins et al. [22] argued increasing the importance of KL Divergence yields better disentanglement and hence selected $\beta > 1$. However, our results show that lower values of β tend to provide better results in terms of lower reconstruction loss and higher disentanglement scores in this specific dataset. The flow of Figure 4.5b reveals that $\beta < 1$ values should also be investigated. Notice that as we decrease β we observe relatively lower reconstruction loss values with relatively higher disentanglement scores, which suggests $\beta < 1$ can also enhance the results. These findings support the results in [18], such that for some dataset $\beta < 1$ can produce better quality reconstructions while increasing the disentanglement of the learned model. The behavior of the Isaac3D dataset is similar to the MPI3D dataset (See Figure 4.5d. We observe a decrease in the disentanglement measure as we increase the β parameter.

4.3 Chapter Summary

In this chapter, we analyze the β hyper-parameter of β-VAE and observe high sensitivity on β. We show that β-VAE has four major drawbacks:

1. β-VAE introduces a new parameter β to hyperparameter space, which is to be tuned together with other hyperparameters of the model using expensive empirical methods in a large search space.
2. Selecting a hyperparameter $\beta > 1$ to improve the amount of disentanglement may result in a relatively poor representation [9] (Observation 2).
3. The ratio between reconstruction loss and the amount of disentanglement heavily depends on the network hyperparameters other than β (Observation 3).
4. The optimal value of β can also be less than 1 for some datasets, further increasing the dynamic range of very large search space (Observation 4).

The above drawbacks highlight the importance of an effective learning model for the adjustment of the loss function weights. We believe that an optimal balance between the reconstruction loss and the degree of disentanglement requires a dynamic learning model. Our models L-VAE and mdL-VAE have self-learning mechanisms that can simultaneously optimize the model parameters and the weights of the loss function without any restriction on the range of these weights, as will be described in the Chapters 6 and 7.

CHAPTER 5

A NEW MEASURE FOR ASSESSING THE QUALITY OF DISENTANGLED REPRESENTATION LEARNING: CORRELATION BASED DISENTANGLEMENT MEASURE

Measuring the quality of disentanglement is a key challenge in disentangled representation learning. Although several measures are provided in the literature [9, 22, 29], they quantify different aspects of the given representation, and a universally agreed measure is absent.

In this chapter, we propose a novel measure called Correlation-based Disentanglement (CbD) with the following purposes:

- *Providing dimension-wise evaluations of the given representation.*

  Disentangled representation learning aims to find a representation such that the attributes of the dataset are modelled in specific dimensions of the representation. Therefore, a measure should mathematically present the degree of disentanglement for each dimension in the representation.

- *Ensuring model independence through statistical measurements.*

  Measures are effective tools for comparing different models, and they should be reproducible. Therefore, the usage of statistical tools without any dependencies increases the reproducibility and robustness of the scores.

- *Offering a visual representation for increasing the interpretability of the representation.*

  Disentanglement learning increases the interpretability of the learned model since each dimension aims to learn a different attribute. A measure should further extend this and provide a visual understanding to the user.

In this chapter, first, we discuss the available measures in the literature. In the second part of this chapter, we introduce our proposed measure, called CbD.

5.1 Commonly-used Disentanglement Measures

As we state in Section 2.4, Carbonneau et al. categorized disentanglement measures in their review [7] into three categories based on the properties they measure. The first property is called *modularity*, which measures whether a change in an attribute affects a single dimension on the representation. The second property is called *compactness*, which measures whether a small portion (ideally one dimension) of the representation represents a single attribute. The last property is called *explicitness*, which corresponds to the ability to recover input from the representation.

In this section, we describe the disentanglement measures we use to measure our proposed models. In these measures, we assume that a given model learns an L dimensional representation $\mathbf{z}^{(i)} \in \mathbb{R}^L$ for a given image $\mathbf{x}^{(i)}$. The measures we describe are supervised with respect to the labels of the attributes. In other words they use N dimensional attribute/factor labels $\mathbf{y}^{(i)} \in \mathbb{R}^N$.

β-VAE measure [22] is designed to quantify modularity of the learned representation. It iteratively selects an attribute y_j, and forms image pairs, $\mathbf{x}^{(i_1)}$ and $\mathbf{x}^{(i_2)}$, such that $\mathrm{y}_j^{(i_1)} = \mathrm{y}_j^{(i_2)}$ and infers representations $\mathbf{z}^{(i_1)}$ and $\mathbf{z}^{(i_2)}$ for these images. Assuming $\mathbf{z}^{(i_1)}$ and $\mathbf{z}^{(i_2)}$ will have the same value in the dimension, modelling y_j, these pairs are represented by absolute difference in their representation, $|\mathbf{z}^{(i_1)} - \mathbf{z}^{(i_2)}|$. They repeat the computation of this difference for several image pairs. They take the average of these differences to form a single point for a linear classification, with label y_j. Repeating the same process for several attributes and image pairs, they form a new dataset. Finally, a linear classifier is used to compute the disentanglement measure. Kim et al. [29] visually explain the flow of computing of this measure as provided in Figure 5.1. The pipeline at the top represents the β-VAE measure. First, they fix an attribute (referred as factor in the figure) and select image samples. They encode representations for these attributes and take absolute differences. Finally taking the mean of these differences produces a sample point for a linear classifier. As a result,

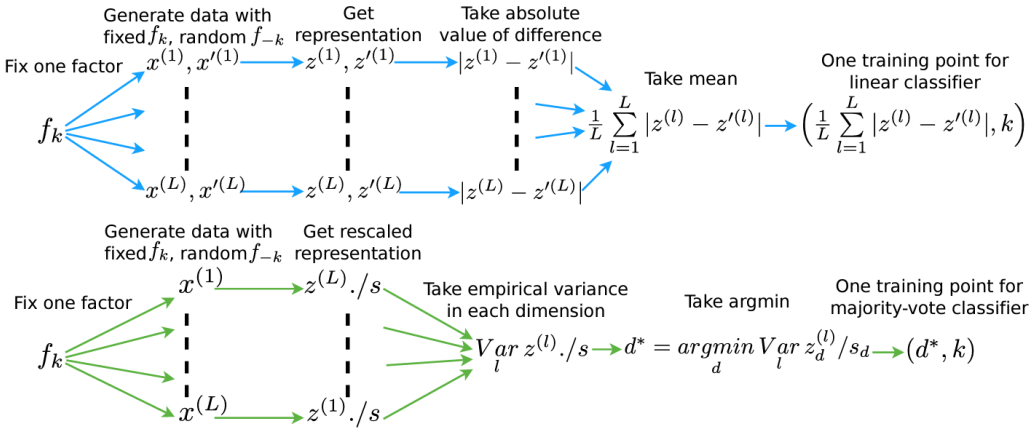


Figure 5.1: Flow of computation for (top) β-VAE and (bottom) Factor-VAE measures. [Figure Source [29]]

β-VAE measure uses a linear classifier to measure disentanglement. Therefore, the result depends on the used classifier.

Factor-VAE [29] also measures modularity and uses a similar strategy to β-VAE measure. However, after selecting an attribute y_j, they first normalize learned representations $\mathbf{z}^{(i)}$ with their empirical variance computed over the entire dataset. Rather than forming image pairs and computing the difference between their representations as in β-VAE measure, they form points for classification by selecting the index with the lowest variance. The bottom pipeline in Figure 5.1, provided by Kim et al. [29], shows the flow of computing Factor-VAE measure.

Mutual Information Gap (MIG) [9] measures compactness of the learned representation. It computes Mutual Information (MI) between the dimensions of the representation $\mathbf{z}_j$ and the attributes $\mathbf{y}_k$ as $MI(\mathbf{z}_j, \mathbf{y}_k)$. For each dimension $\mathbf{z}_j$, letting $MI(\mathbf{z}_j, \mathbf{y}_*)$ is the highest, and $MI(\mathbf{z}_j, \mathbf{y}_o)$ is the second highest mutual information for that specific dimension, the gap is computed as:

$$MIG = \frac{MI(\mathbf{z}_j, \mathbf{y}_*) - MI(\mathbf{z}_j, \mathbf{y}_o)}{H(\mathbf{z}_j)}, \tag{5.1}$$

where the entropy of $\mathbf{z}_j$ is $H(\mathbf{z}_j)$. It is used to normalize the gap. Notice that the value of MIG depends on the number of bins of discretization and the selected estimation

method.

MIG_SUP [37] measures modularity as an extension over MIG. Rather than computing mutual information gap across dimensions, MIG_SUP computes the gap across attributes. In other words, when MI($\mathrm{z}_*, \mathrm{y}_j$) is the highest mutual information for attribute y_j, and MI($\mathbf{z}_o, \mathbf{y}_j$) is the second highest mutual information, the gap is computed by $MI(\mathbf{z}_*, \mathbf{y}_j) - MI(\mathbf{z}_o, \mathbf{y}_j)$.

Explicitness score [53] measures explicitness property of a representation. They argue that in an explicit representation, it should be possible to recover the attributes from the representation itself, using only a simple classifier. They employed one-versus-rest logistic-regression classifier to classify attributes from the learned representation. They use the ROC Area Under the Curve (AUC) to measure the explicitness of a given representation. Notably, this evaluation metric operates under the assumption that all factors possess discrete values, and the result depends on the classifier selection.

Modularity score [53] measures modularity of the learned representation by first estimating MI between each latent dimension $\mathbf{z}_j$ and each attribute $\mathbf{y}_k$ as $MI(\mathbf{z}_j, \mathbf{y}_k)$. They assume a modular dimension $\mathbf{z}_j$ will have a high Mutual Information score with a single attribute $\mathbf{y}_*$; $MI(\mathbf{z}_j, \mathbf{y}_*)$, and all other MI values will be zero (when the set of attributes is denoted with Y, they assume $MI(\mathbf{z}_j, \mathbf{y}_{Y-\{*\}}) = 0$). They first create a *template vector* per factor as $\mathbf{t}_{jk}$, such that

$$\mathbf{t}_{j*} = MI(\mathbf{z}_j, \mathbf{y}_*),$$

and

$$\mathbf{t}_{j,Y-\{*\}} = 0.$$

For each factor $\mathbf{z}_i$, they compute deviations from this template vector as follows;

$$\delta_j = \frac{\sum_k (MI(z_j, y_k) - t_{jk})^2}{MI(\mathbf{z}_j, \mathbf{y}_*)^2 (N-1)}, \tag{5.2}$$

where N is the number of attributes, and 0 deviation refers to perfect modularity. They use $1-\delta$ as the modularity score. Notice that similar to the MIG measure, modularity score also depends on the estimation method of mutual information.

Separated Attribute Predictability (SAP) score [34] measures compactness of each attribute in the representation space by computing classification (or regression) score s_{jk} for each dimension $\mathbf{z}_j$, given attribute label $\mathbf{y}_k$. Similar to MIG measure, for any attribute $\mathbf{y}_k$ they compute the difference between the highest classification accuracy s_{j*} and the second highest classification accuracy s_{jo}, $s_{j*} - s_{jo}$ to measure compactness. Notice that the result of SAP score depends on the classification algorithm.

Table 5.1: A comparative summary of the suggested CbD with existing measures listing assumptions and dependencies of the measures

Method	Assumptions	Dependencies of the output
β-VAE [22]	×	Classifier selection
Factor-VAE [29]	×	Classifier selection
MIG [9]	×	number of bins of discretization and the selected MI estimation method
MIG_SUP [37]	×	number of bins of discretization and the selected MI estimation method
Explicitness [53]	×	Classifier selection
Modularity [53]	×	number of bins of discretization and the selected MI estimation method
SAP [34]	×	Classifier selection
CbD (OURS)	Gaussian	×

Table 5.1 shows the disentanglement measures we discuss and use in our experiments. We explicitly state the dependencies of the computed score per measure in this table. As shown, all of the measures available in the literature have dependencies. The outcome may depend on the classifier selection, or the estimation method for the Mutual Information estimation method. As we require a more robust disentanglement measure, we propose to compute the degree of disentanglement using Pearson Correlation Coefficients. The last row of Table 5.1 shows our proposed CbD measure. In the next section, we define CbD disentanglement measure.

5.2 Our Novel Correlation-based Disentanglement Measure

We propose two new measures to quantify modularity and compactness of the learned representation. Then, we combine these measures to define a new hybrid measure, called Correlation-based Disentanglement (CbD). The suggested measure aims at

(i) providing dimensionwise evaluations so that we can interpret the degree of disentanglement for each dimension in the representation,

(ii) using statistical measurements to ensure robustness on the measure by eliminating model dependencies,

(iii) providing both a scalar measurement and a visual representation for easy interpretation on the learned model.

Our main goal is to eliminate model dependence that is to remove any classifier or estimator-based dependencies we encounter in similar measurements as in SAP or MIG scores (See Section 5.1 for details). Table 5.1 shows the assumptions and dependencies of the measures from the literature and our measure CbD. We based CbD on Pearson Correlation to provide a more robust score with a statistical measure.

Pearson correlation coefficient serves as a robust foundation for the proposed CbD measure. We start by computing the Pearson correlation coefficient (PCC) between each dimension $\mathbf{z}_j$ in the representation space and factor labels $\mathbf{y}_k$, as follows:

$$r_{jk} = \text{PCC}(\mathbf{z}_j, \mathbf{y}_k) = \frac{\sum_c \left(z_j^{(c)} - \bar{z}_j\right)\left(y_k^{(c)} - \bar{y}_k\right)}{\sqrt{\sum_c \left(z_j^{(c)} - \bar{z}_j\right)^2 \sum_c \left(y_k^{(c)} - \bar{y}_k\right)^2}}, \tag{5.3}$$

where c runs over different samples; $\bar{z}_i$, and $\bar{y}_i$ are the mean values over the samples. Pearson Correlation estimates a correlation value r within the range $[-1, 1]$. The coefficients $|r_{jk}| = 1$ show strong positive and negative relations, respectively, whereas the coefficient value of $r_{jk} = 0$ shows that there is no relationship between the variables. We take the absolute of value r_{jk} since we are interested in both the positive and negative correlations as follows:

$$P_{jk} = \text{abs}(r_{jk}). \tag{5.4}$$

Assuming latent representation has N dimensions and the dataset contains M labelled attributes, we form an $N \times M$ dimensional Pearson correlation (PCC) matrix:

$$\mathbf{P} = [P_{jk}], \tag{5.5}$$

where each entry is the pairwise correlation coefficients between the dimensions of the representation and the factors of the dataset,

We visualize the PCC matrix in Figure 5.2. Let us now analyze the rows and columns of this matrix in terms of the compactness and modularity of the latent dimensions and the overall learned representation.

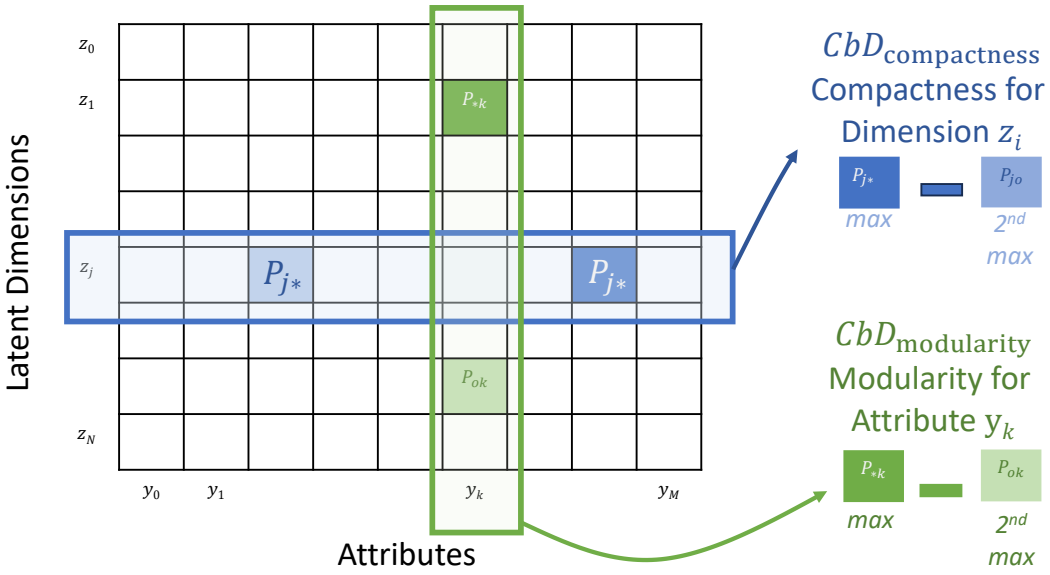


Figure 5.2: Visualization of Correlation-based Disentanglement. We compute compactness per dimension as the difference between the highest and second highest values in a row (see vertical box covering dimension z_j). Similarly, we compute modularity per attribute as the difference between the highest and second highest values in a column (see horizontal box covering attribute y_k).

Recall from Section 2.4 that compactness measures whether an attribute is represented in a single dimension of the model. If any given dimension $\mathbf{z}_j$ is compact, we expect the corresponding row of the PCC matrix $\mathbf{P}_j$ to have a single peak value, i.e. high correlation with a single dimension. In other words, dimension $\mathbf{z}_j$ is said to compactly learn attribute $\mathbf{y}_k$, when P_{jk} is close to one, and other values P_{jm} are close to zero for all $m \neq k$. Therefore, the row P_j of the PCC matrix, shown in Figure

5.2 as a horizontal blue box, give us an important clue about the compactness of the associated dimension. In order to evaluate the compactness of a latent dimension, we suggest a heuristic scalar score by measuring the "sharpness" of the peak among the entries of each row. First, we select the highest entry per row, represented as P_{j*} and then select the second highest value in that row, represented as P_{jo}. If the dimension $\mathbf{z}_j$ is compact, we expect the difference $P_{j*} - P_{jo}$ to be high.

Our first measure, which quantifies the compactness of the overall representation is defined as follows:

$$\text{CbD}_{\text{compactness}} = \frac{1}{N}\sum_{j=1}^{N}(P_{j*} - P_{jo}), \tag{5.6}$$

where N is the number of dimensions in the representation. We repeat the same process across all rows of the matrix and take the average.

Similarly, we can define a scalar score to measure the modularity of the learned representation by analyzing the columns of the PCC matrix, shown in Figure 5.2 as a vertical green box. For a modular representation, each attribute should be represented by a single latent dimension, such that any change in a single attribute would affect a single dimension. By focusing on the column vectors P_k, instead of rows of the PCC matrix, $\mathbf{P}$, we can measure modularity of the learned representation, since we expect each attribute to have a high correlation with a single dimension. Similar to compactness, we select the highest and second highest values per column as P_{*k} and P_{ok}, and we expect this difference to be high for a modular representation. Hence, we define a heuristic modularity score, as follows:

$$\text{CbD}_{\text{modularity}} = \frac{1}{M}\sum_{k=1}^{M}(P_{*k} - P_{ok}), \tag{5.7}$$

where M is the number of attributes in the dataset. We repeat the same process across all columns of the matrix and take the average.

It is possible to combine the above compactness and modularity scores to define an overall disentanglement score as follows:

$$\text{CbD} = \alpha \cdot \text{CbD}_{\text{compactness}} + (1-\alpha) \cdot \text{CbD}_{\text{modularity}}. \tag{5.8}$$

A simple way of combining these two scores is to take the arithmetic average of them where $\alpha = 0.5$. Hence, we define a new disentanglement measure, called,

correlation-based disentanglement CbD score, as follows:

$$\text{CbD} = \frac{1}{2}(\text{CbD}_{\text{compactness}} + \text{CbD}_{\text{modularity}}). \tag{5.9}$$

The proposed CbD score measures both the compactness and modularity. Based on the classification of Carbonneau et al. [7], CbD is classified as a holistic measure. If the main goal is to measure a specific property of a given model, both $\text{CbD}_{\text{modularity}}$ or $\text{CbD}_{\text{compactness}}$ can be used individually, or the weight α can be selected such that it gives higher weights on either of the terms. For simplicity, we will use the measure CbD with $\alpha = 0.5$ throughout this thesis unless otherwise stated.

Significance of the proposed CbD measure. Recall that the disentanglement measures in the literature are sensitive to the selected model (See Section 2.4). For instance, the results of β-VAE and SAP score depend on the classifier. On the other hand, the suggested $\text{CbD}_{\text{modularity}}$, $\text{CbD}_{\text{compactness}}$, and CbD stands out by being model-agnostic, eliminating dependencies on specific model choices. It is crucial to acknowledge that suggested measures are based on the Gaussian assumption for both the latent variables and labels, which is a general assumption in all variants of the VAE methods. This consideration stems from the reliance of Pearson Correlation coefficients on Gaussian assumptions.

Moreover, the PCC matrix allows us to analyse the correlation between each dimension and attribute, which indicates the degree of disentanglement for each attribute. As we experimentally show in Chapter 6, in some cases, the degree of disentanglement changes across different dimensions. With PCC, we can analyse these differences dimensionwise.

Finally, the heatmap derived from the Pearson Correlation Matrix $\mathbf{P}$ serves as a visual aid, offering users an intuitive tool for an in-depth analysis of the disentanglement properties of a model.

5.3 Experimental Analysis of the Suggested Disentanglement Measures

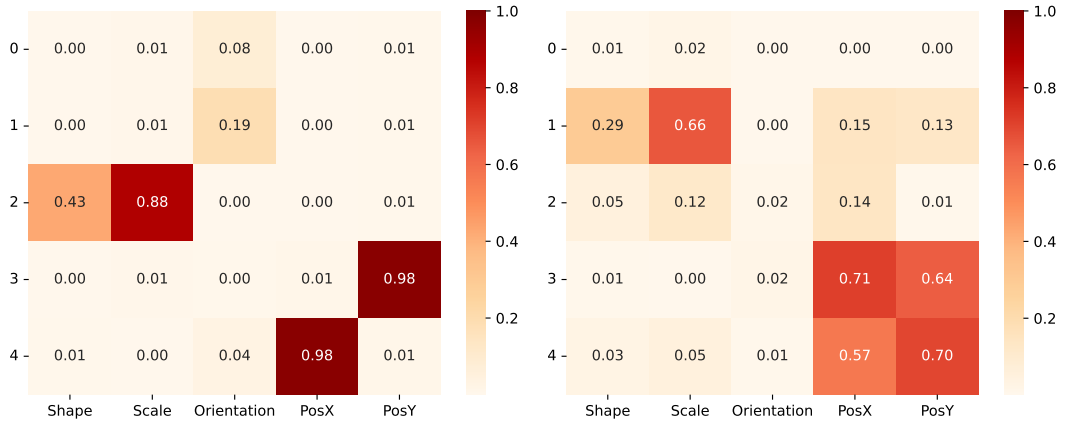
In this section, we conduct a series of experiments to investigate the power of the suggested measures. First, we examine the properties of PCC matrix Next, we will

discuss the outcomes of our measures, focusing on two different trained models. Finally, we will compare our measures with the available measures, we have discussed in Section 5.1.

5.3.1 Analysis on the Entries of PCC Matrix

In this section, we compare the outcomes of PCC matrix in different models. In order to make this comparison, we train two random disentanglement models M^1 and M^2 on the dSprites dataset for simplicity. We extract the Pearson Correlation Matrices $\mathbf{P}^1$ and $\mathbf{P}^2$ for these models for analyses. Figure 5.3 shows the PCC matrices $\mathbf{P}^1$ and $\mathbf{P}^2$. In these matrices, the numbers on the y-axis represent the index of the dimension. For example, the first row with index 0 represents the dimension z_0. Recall that, for higher disentanglement, we want every row and column of this matrix to have a single peak value to achieve modular and compact models, respectively.

First, we visually analyse the PCC Matrix $\mathbf{P}^1$ obtained from the first model M^1 from Figure 5.3a. We can see that the dimensions z_3 and z_4, which are the last two rows, show perfect compactness, where they have a single peak within a row with the value 0.98. We can conclude that these dimensions learns PosX and PosY attributes, in the corresponding column. The dimension z_2, on the other hand, has a high score for the *Scale* attribute; however, this dimension has another high value with 0.43 corresponding to the *Shape* attribute. Therefore, any alteration on this dimension affects *Shape* as well as *Scale* attributes. Similarly, *PosX* and *PosY* attributes show perfect modularity since the corresponding two columns have a single peak within the column. This means that any change in *PosX* and *PosY* attributes does not affect any other dimension than z_3 and z_4. On the other hand, when we consider the PCC matrix $\mathbf{P}^2$ of the model M^2 from Figure 5.3b, we can see that all of the correlation values are relatively low compared to M^1 in Figure 5.3a. Visually exploring the PCC matrices, we can conclude that model M^1 has relatively better disentanglement abilities than model M^2. In order to support this information, we analyse the CbD scores for these two models.



(a) $\mathbf{P}^1$ (b) $\mathbf{P}^2$

Figure 5.3: Sample Pearson correlation matrices $\mathbf{P}^1$ and $\mathbf{P}^2$ for two different random models M^1 and M^2. We can state that the model M^1 in (a) displays better disentanglement properties with higher peak values for each row and column.

5.3.2 Analysis on $\text{CbD}_{\text{Modularity}}$, $\text{CbD}_{\text{Compactness}}$, and CbD Scores

Table 5.2 shows the compactness, modularity, and overall disentanglement scores obtained from $\text{CbD}_{\text{Modularity}}$, $\text{CbD}_{\text{Compactness}}$, and CbD scores for the models M^1 and M^2. Note that, higher values indicate better disentanglement in all of these measures. As Table 5.2 shows, M^1 achieved higher scores on all of the measures. In other words, considering overall scores, we can say that CbD suggests that the model M^1 has better disentanglement abilities than the model M^2, which supports our visual analysis.

It is important to note that all of the measures we have discussed in Section 5.1 unanimously provide higher disentanglement scores for M^1, which states CbD is comparable to other measures from the literature. This concludes CbD is capable of showing the differences between different models qualitatively.

Table 5.2: Scores obtained from CbD for the two random models M^1 and M^2 in Figure 5.3

Model	$\text{CbD}_{\text{Compactness}}$	$\text{CbD}_{\text{Modularity}}$	CbD
M^1	0.66	0.52	0.59
M^2	0.16	0.20	0.12

5.3.3 A Comparison of Proposed Measures with Other Measures

In this section, we compare CbD,$\text{CbD}_{\text{Compactness}}$, and $\text{CbD}_{\text{Modularity}}$ with the measures from the literature given in Section 5.1. In order to achieve this goal, we have measured disentanglement over 1286 different models on four datasets (See Section 3.2.2 for details). After estimating the disentanglement measures for these 1286 independent models, we compute a pairwise person correlation matrix for all of the measures to analyse their relationship. As stated by Carbonneau et al. [7], different disentanglement measures focus on different aspects of disentanglement: compactness, explicitness and modularity (See Section 2.4 for details). In this experiment, we aim to find the correlations among the measures. We first compare CbD with the available measures from the literature. Figure 5.4 shows the heatmap of the Pearson Correlation Matrix across CbD, β-VAE, FactorVAE, Explicitness Score, IRS, MIG, MIG_SUP, Modularity Score and Sap scores.

Figure 5.4 suggests that the measures have varying degrees of correlation, ranging from correlation values very close to 0 to high correlation values around 0.94. This figure depicts that CbD is highly correlated with MIG and MIG_SUP measures. It is noteworthy that CbD is mathematically similar to the summation of MIG and MIG_SUP scores since these two measures also calculate the relationship between representation and the factors. However, Mutual Information-based models depend on binning and estimation methods, which make them more difficult to use than CbD measure. On the other hand, CbD shows no correlation with Explictness. This is an expected result since CbD only measures compactness and modularity. These results verify the suggestion by Carbonneau et al. [7], who show that different measures focus on different aspects of disentanglement, and therefore, the measure should be

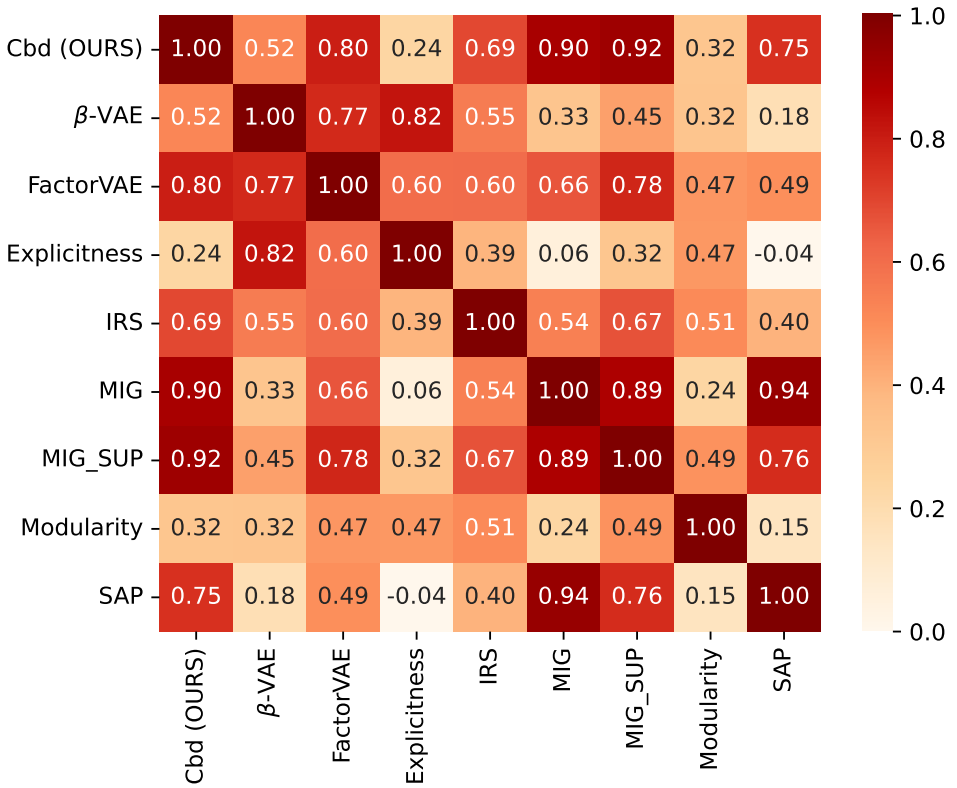


Figure 5.4: Pearson correlation between that available disentanglement measures and CbD that we propose. Note that all of these measure different aspects of disentanglement in a model.

selected carefully based on the task. Note that there is a negative value in this matrix ($P_{Exp,SAP} = -0.04$), which is statistically insignificant with p-value = 0.12.

Since all of the measures focus on different properties of a disentangled model, we carry the same experiment among measures focusing on the same property. In order to achieve this, we first analyse the measures for compactness. Figure 5.5 shows the correlation among these measures, namely $\text{CbD}_{\text{Compactness}}$, MIG, and SAP Score which also measure the compactness property of a learned representation. Note that we also include the composite CbD for completeness. As this figure clearly presents, all of the measures that focus on compactness have relatively higher correlation values except $\text{CbD}_{\text{Compactness}}$ and SAP score with a value of 0.65.

Similarly, we repeat the same experiment with modularity-based measures, namely $\text{CbD}_{\text{Modularity}}$, β-VAE, FactorVAE, MIG_SUP and Modularity score in Figure 5.6.

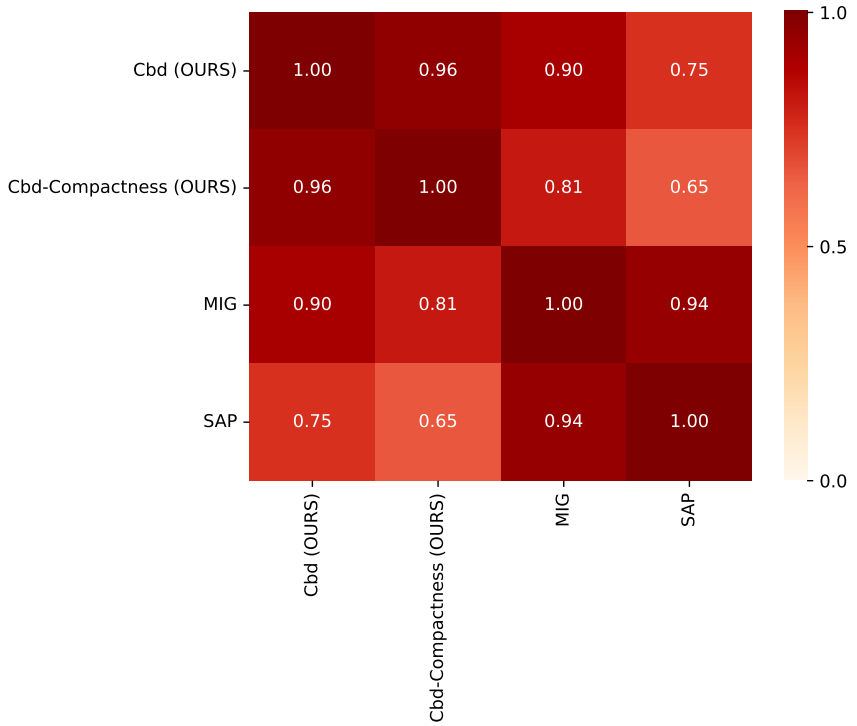


Figure 5.5: Pearson correlation among Our measures and others from the literature that measures compactness.

We also include CbD for completeness in this figure as well. Unlike compactness measures in Figure 5.5, modularity-based measures show a relatively low correlation among themselves. Especially β-VAE and Modularity scores show no correlation with other measures. We can see that both CbD and $\text{CbD}_{\text{Modularity}}$ show a high correlation with MIG_SUP and FactorVAE measures.

Overall, we can say that CbD can measure compactness in correlation with other measures. However, in terms of modularity and explicitness, measure selection will have a high impact on the outcomes of the models.

5.4 Chapter Summary

In this chapter, we first introduce the disentanglement measures from the literature and discuss their disadvantages. All of the measures have dependencies such as model dependencies. In other words, the outcome of these measures depends on the

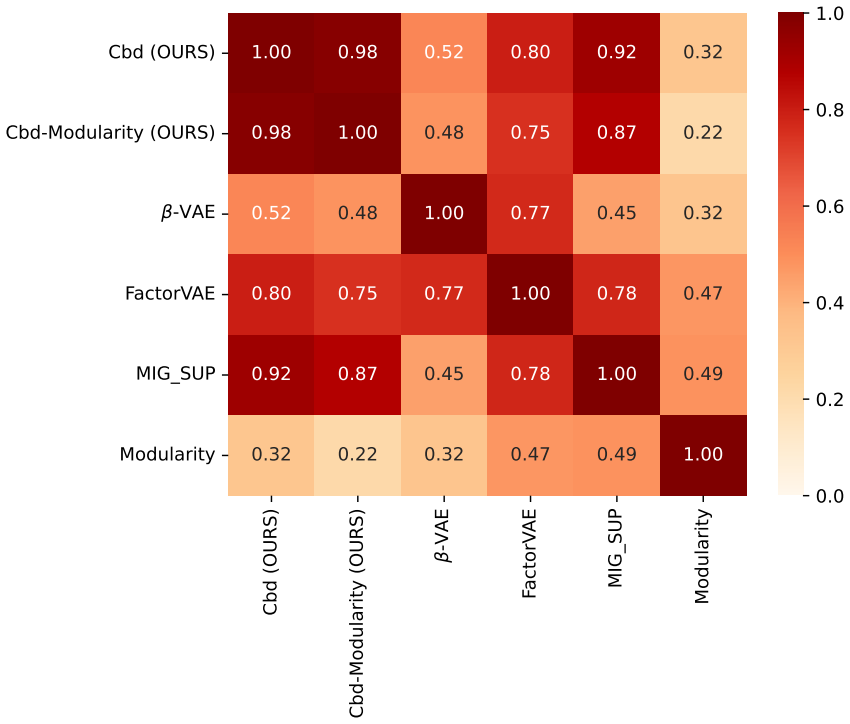


Figure 5.6: Pearson correlation among Our measures and others from the literature that measures modularity.

model selection. In order to provide a more robust measure, we propose $\text{CbD}_{\text{Compactness}}$ $\text{CbD}_{\text{Modularity}}$ to measure compactness and modularity properties of a given disentanglement model relatively. Then we combine these two measures and propose CbD to measure disentanglement of a learned representation. We show that CbD can successfully measure disentanglement for any disentangled representation learning models. We also show that it produces dimensionwise evaluations, and gives crucial information on the level of disentanglement per each dimension in the representation. Moreover, we discuss the correlation between CbD, $\text{CbD}_{\text{Compactness}}$, and $\text{CbD}_{\text{Modularity}}$ and the available measures from the literature and show that $\text{CbD}_{\text{Compactness}}$ highly correlates with the measures that focus on compactness property. However, there seems to be less correlation among measures that focus on modularity. As a result, we argue that measure selection is a crucial problem in disentanglement literature.

CHAPTER 6

LEARNABLE VARIATIONAL AUTO-ENCODERS

As we discuss in Chapter 4, β-VAE model increases model complexity by introducing a new hyperparameter β in the loss function. In order to overcome this limitation, we introduce a new model, called Learnable VAE (L-VAE). The main goal of L-VAE is to reduce the model complexity introduced with the new hyperparameter β. Therefore, this model focuses on dynamically tuning the β parameter, a crucial element of the algorithm. We replace the hyperparameter β with a weight which can be learned dynamically along with the network weights. We assess the learned weights and resulting disentangled representation comprehensively using CbD and conclude that L-VAE can achieve better or on par disentanglement scores with β-VAE, and also learn weights that are aligned with the empirically tuned β of the β-VAE algorithm.

In this chapter, we formally introduce L-VAE and discuss its strengths and weaknesses in detail. First, we analyse the convergence of learned weights. We also compare L-VAE with five methods from the literature, namely VAE, β-VAE, ControlVAE, DynamicVAE, and σ-VAE, on four different datasets, dSprites, MPI3D, Falcor3D, and Isaac3D.

6.1 Formal Representation of Dataset and Its Sources of Variation

Formally, consider a dataset,

$$\mathcal{D} = \{\mathcal{X}, \mathcal{W}, \mathcal{Y}\},$$

with S samples, where the triplet $(\mathcal{X}, \mathcal{W}, \mathcal{Y})$ represents the set of image samples, the underlying sources generating $\mathcal{X}$, and the labels for $\mathcal{X}$, respectively. Image sample i,

denoted by

$$\mathbf{x}^{(i)} \in \mathcal{X} \in \mathbb{R}^{W \times H},$$

which is an $W \times H$-dimensional matrix. Each image sample $\mathbf{x}^{(i)}$ is generated by the mixture of K *conditionally independent* and *unobservable* factors of variation denoted as,

$$\mathbf{w}^{(i)} \in \mathcal{W} \in \mathbb{R}^{K}.$$

In other words, sample $\mathbf{x}^{(i)}$ can be simulated using its source distributions, expressed as,

$$\mathbf{x}^{(i)} \sim Sim(\mathbf{w}^{(i)}).$$

The dataset is assumed to be generated by the independent factors $\mathbf{w}^{(i)}$, and these factors are assumed to be the causes of the attributes of an image sample, such as the shape, orientation or colour of an object. Dataset $\mathcal{D}$ can include attribute labels $\mathcal{Y}$, such that $y_j \in \mathbb{R}^M$ represents the label for the j^{th} factor $w_j \in \mathbf{w}^{(i)}$. It is important to note that, in most cases, $\mathcal{W}$ is unobservable. Hence, the size of the labels are lower than the size of the factors, that is $M < K$.

6.2 Learnable VAE (L-VAE)

In this section, we describe the proposed Learnable VAE (L-VAE), which circumvents empirical optimization of the hyperparameter β of β-VAE. The success of the β-VAE model lies behind the fact that reconstruction loss and KL divergence inherently possess different numerical ranges, necessitating a balance between the optimized values of each term. However, this approach adds the extra hyperparameter β into the system, thereby complicating the optimization problem at hand.

In order to improve the performance and increase the efficiency of the underlying optimization problem, we impose weights on both the reconstruction loss and KL divergence terms, with the goal of attaining minimal reconstruction loss while concurrently ensuring a robust disentanglement for the hidden variables. This approach enables us to find a relatively more flexible and adaptive solution for optimal disentanglement.

Our method is based on the multi-task learning method of Kendall et al. [28], where they learn the relative weights of different tasks in a loss term by augmenting it with additional learnable weights. Without loss of generality, let us assume that the overall loss function consists of two terms, $L_0(\mathbf{x})$ and $L_1(\mathbf{x})$, each of which are the functions of the input vector, $\mathbf{x}$. Then, the trade-off between the terms of the loss function can be dynamically learned from the following analytical form:

$$L(\mathbf{x}) = \frac{1}{\sigma_0^2} L_0(\mathbf{x}) + \frac{1}{\sigma_1^2} L_1(\mathbf{x}) + \log(\sigma_0) + \log(\sigma_1), \tag{6.1}$$

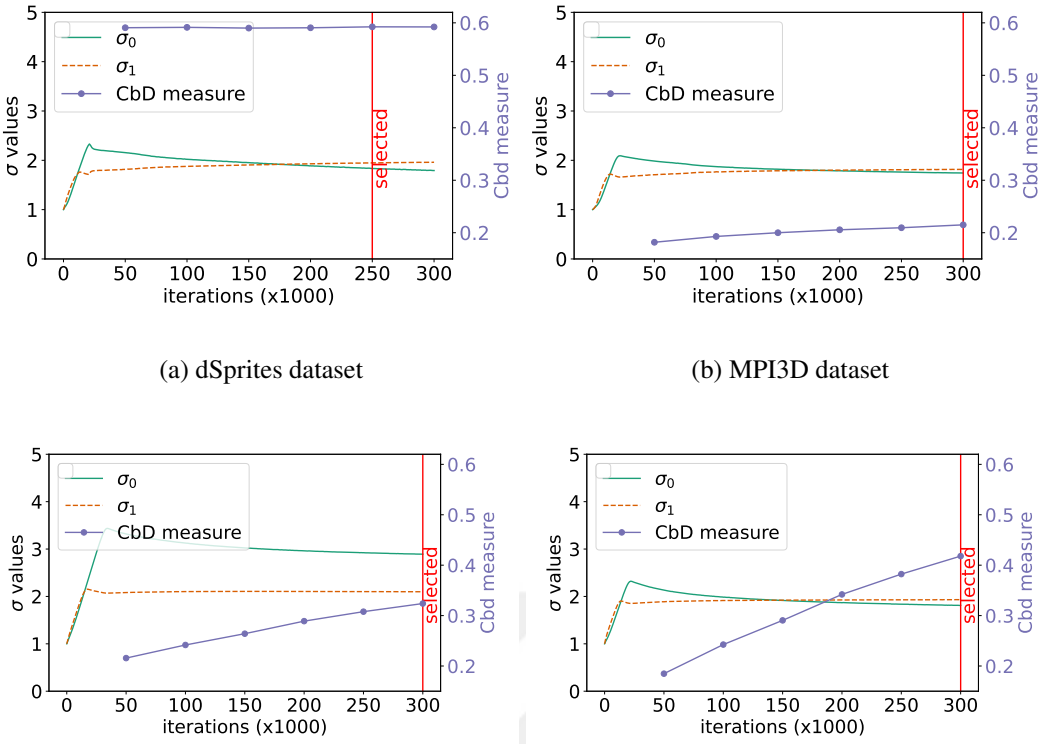
where $1/\sigma_i^2$, for $i = 0, 1$, are the balancing weights between $L_0(\mathbf{x})$ and $L_1(\mathbf{x})$. The parameters σ_i are optimized and updated with other learnable weights of the neural network. The last two terms of Equation 6.1 regularize the learned weights, σ_0 and σ_1. The common practice is to optimize the log variance ($s := \log \sigma_i^2$) for numerical stability [28].

Building upon the same framework of Kendall et al. [28], we suggest a cost function for L-VAE, such that it learns the relative weights of the terms of the VAE loss function by updating Equation 2.2, as follows:

$$\begin{aligned} L_{L-VAE}(\theta, \phi; \mathbf{x}, \mathbf{z}) = & \frac{1}{\sigma_0^2} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log p_\theta(\mathbf{x}|\mathbf{z}) \Big] \\ & - \frac{1}{\sigma_1^2} D_{KL} \Big(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}) \Big) \\ & + \sum_{i=0,1} \sigma_i^2. \end{aligned} \tag{6.2}$$

In order to regulate the upper limit of the weight parameter σ_i in Equation 6.2, we introduce a regularization term $\sum \sigma_i^2$. The σ_i terms are added to the parameter set of the optimizer. Note that selecting σ_i^2 as a regilarization term rather than $\log(\sigma))$ provides stronger regularization. They are estimated through the simultaneous learning with the network parameters (refer to Section 6.3.1 for the learning curves of σ_i).

Except for the relative weights σ_i in the loss function, the rest of the network architecture is constructed exactly the same as β-VAE. Therefore, a straightforward and fair comparison between β-VAE and L-VAE is possible through a simple conversion of the σ_i parameters to the β parameter. In Section 6.3, we show the correspondence of β and σ_i parameters. During the experiments, we observe that, after the training phase, the ratio of the learned parameters, $\frac{\sigma_0^2}{\sigma_1^2}$ in L-VAE is aligned with the empirically tuned value of β parameter in β-VAE.



(a) dSprites dataset

(b) MPI3D dataset

(c) Falcor3D dataset

(d) Isaac3D dataset

Figure 6.1: Training curves of the learned relative weights (σ_0 and σ_1) of L-VAE model for (a) the dSprites, (b) the MPI3D (c) the Falcor3D, and (d) the Isaac3D datasets. The purple line corresponds to the CbD measure computed at every 50,000 iterations. We note that σ_i peaks around 30K - 50K iterations, after which the change decelerates. We determined the number of iterations (along with other hyper-parameters, learning rate, and batch size) based on the CbD measure on the validation dataset as defined in Section 3.2.3 for details). Red vertical lines show this selection.

6.3 Experiments on L-VAE

In this section, we provide the results of our experiments. Refer to Chapter 3 for details of the dataset and experimental setup we use in these analyses. In this section, first, we analyse the learned weights of the proposed L-VAE. Then, we evaluate the performance of our model in comparison with the baseline VAE-based models. We measure the performances by employing the reconstruction loss and disentanglement measures, which are selected from the literature, as explained in the previous section.

Finally, we carry out an ablation study to explore the effects of a dynamic learning strategy proposed in this paper.

6.3.1 Experiment 1: Convergence of the Weights Learned by L-VAE

In this set of experiments, we analyze how the parameters σ_0 and σ_1 (Equation 6.2) change over time during the learning phase. In all experiments, we select the σ_i values that maximize the CbD measure.

During the derivations of L-VAE, we mentioned that there is a correspondence between the empirically tuned β parameter of β-VAE model and the optimal ratio of the learned parameters σ_i of L-VAE model as follows,

$$\hat{\beta} = \frac{\sigma_0^2}{\sigma_1^2}. \tag{6.3}$$

Hence, we shall also investigate the behavior of the optimal ratio $\hat{\beta}$ for each dataset.

Figure 6.1 shows the learning curves of the σ_i parameters. We notice that, in the dSprites dataset, σ_i exhibits a steep ascent until 50K iterations, reaching a peak value, after which the change attenuates. However, the CbD measure slightly increases as σ_0 decreases. During the cross-validation step, the σ_i values, obtained at 250Kth iteration, are selected. These values correspond to the maximum CbD measure.

The learning curves of σ_i parameters for the rest of the datasets show similar behaviors. However, the CbD measure keeps increasing during the training phase. As CbD measure increase, we observe that the ratio of σ_0^2/σ_1^2 also changes: For the Falcor3D dataset, σ_0 surpasses σ_1, yielding the generally practiced estimation for $\hat{\beta} = \frac{\sigma_0}{\sigma_1} > 1$. On the other hand, in the other datasets, σ_1 overtakes σ_0, yielding an unusual estimation for $\hat{\beta} = \frac{\sigma_0}{\sigma_1} < 1$. This result is rather counter-intuitive because disentanglement is accentuated for $\hat{\beta} > 1$. This observation shows the importance of dynamically learning the σ_i parameter, which approximates $\beta = \frac{\sigma_0}{\sigma_1}$ of β-VAE.

6.3.2 Experiment 2: Comparison with Baseline Methods

In this set of experiments, we compare VAE, β-VAE, ControlVAE, DynamicVAE, σ-VAE (see Section 2.3.3.2 for a brief description of the methods), and L-VAE based on the MSE reconstruction loss and the four disentanglement measures including CbD. Note that the best set of hyperparameters is selected based on CbD. Table 6.1 shows the results for all four datasets. For β-VAE, the β value is determined through a hyperparameter search process described in Section 3.2.3, whereas, for L-VAE, the values of σ_i represent the learned values.

First, we compare the learned value of $\hat{\beta} = \sigma_0^2/\sigma_1^2$ with the empirically tuned β parameter for L-VAE experiments. The learned value of $\hat{\beta}$ is 0.89, 0.92, 1.90, and 0.88 for dSprites, MPI3D, Falcor3D, and Isaac3D datasets, respectively. These values are consistent with our findings in Figure 4.5, which suggests that the optimal value of β might be lower than 1 for the MPI3D and Isaac3D datasets. These results are aligned with the ratio of the weights $\hat{\beta} = \sigma_0^2/\sigma_1^2$ learned by L-VAE.

To assess the disentanglement performances of the different models, we examine their disentanglement properties (modularity, explicitness, and compactness) separately for each dataset. Recall that modularity is measured by $\text{CbD}_{\text{Modularity}}$, CbD, β-VAE, FactoVAE, MIG_SUP, IRS, and modularity scores. Explicitness is measured by IRS and explicitness measures. Compactness is measured by $\text{CbD}_{\text{Compactness}}$, CbD, SAP, and MIG scores. In order to be able to cover all of the disentanglement properties, we select a single measure per disentanglement property. Therefore, we select CbD, β-VAE, MIG, and Explicitness score to compare the models. In Chapter 7, we provide the outcomes of all of the measures when we make a comparison with another model. For the sake of simplicity, we discuss L-VAE with these four disentanglement measures.

Table 6.1 shows the results of the comparisons of four different datasets. Bold numbers indicate the best performances, whereas the underlined values indicate the second-best performances. For reconstruction, we require lower values for the best reconstruction quality. On the other hand, for all of the disentanglement scores, a higher number represents a better disentanglement. We discuss the results for each dataset

below;

Table 6.1: Experiment 2: Quantitative comparison of our model across related models. The best models are selected via CbD.

Dataset	Model	Reconstruction↓	CbD ↑	β-VAE ↑	mig ↑	explicitness ↑
dSprites	VAE	**11.86**	0.58	0.93	0.30	0.56
	β-VAE ($\beta = 4$)	29.11	0.58	0.92	0.30	0.55
	ControlVAE	24.35	**0.60**	**0.97**	0.30	**0.59**
	DynamicVAE	33.75	0.56	0.89	**0.33**	0.50
	σ-VAE	12.30	0.29	0.77	0.07	0.47
	L-VAE ($\hat{\beta} = 0.89$)	**11.77**	0.59	0.96	0.31	0.57
MPI3D	VAE	**9.91**	**0.23**	**0.75**	0.18	0.40
	β-VAE ($\beta = 2$)	16.12	0.21	0.65	0.15	0.36
	ControlVAE	19.74	0.22	0.62	0.12	0.36
	DynamicVAE	18.46	**0.23**	0.69	0.14	0.36
	σ-VAE	12.70	0.22	0.65	0.05	**0.49**
	L-VAE ($\hat{\beta} = 0.92$)	**9.16**	0.22	0.73	**0.20**	0.47
Falcor3D	VAE	68.15	0.27	0.93	0.13	**0.64**
	β-VAE ($\beta = 2$)	71.37	0.37	0.91	**0.16**	0.62
	ControlVAE	237.69	**0.39**	0.87	0.15	0.45
	DynamicVAE	205.04	0.24	0.77	**0.16**	0.34
	σ-VAE	**65.60**	0.20	0.87	0.06	0.61
	L-VAE ($\hat{\beta} = 1.90$)	70.67	0.32	**0.94**	0.15	0.63
Isaac3D	VAE	11.88	0.48	0.76	0.32	**0.59**
	β-VAE ($\beta = 2$)	14.71	0.50	**0.77**	**0.33**	0.55
	ControlVAE	34.56	0.26	0.65	0.15	0.46
	DynamicVAE	39.80	0.19	0.51	0.13	0.40
	σ-VAE	26.31	0.20	0.61	0.04	0.54
	L-VAE ($\hat{\beta} = 0.88$)	**10.73**	0.42	0.75	0.26	0.56

- For **dSprites** dataset, it can be seen that L-VAE achieves the best reconstruction quality. In terms of disentanglement, L-VAE achieves the second-highest scores for all of the measures. On the other hand, ControlVAE algorithm achieves the best scores for three out of four disentanglement measures.

- For the **MPI3D** dataset L-VAE achieves the best reconstruction quality. Moreover, L-VAE achieves the best score in MIG measure, and the second best in the rest of the measures. However, we can not argue that any of the models outperform the others based on these results, since none of the models perform the best disentanglement scores across different measures.

- For **Falcor3D** dataset, L-VAE can achieve the highest score for β-VAE measure, and it achieves the second best results for MIG and Explicitnes scores. However, there is no other method that performs the best consistently across all measures.

- For **Isaac3D** dataset L-VAE can achieve the best reconstruction quality, and second-best quality in terms of Explicitness score. On the other hand, β-VAE model achieves two best scores across four disentanglement measures.

Overall, the results suggest that L-VAE can learn weights (σ_i) consistent with our preliminary observations where we show that increasing β can lead to higher reconstruction loss and $\beta < 1$ can produce higher disentanglement scores in some datasets (See Observations 2 and 4 in Section 4.2, and Figure 4.5). Although we make comparisons on several measures and datasets, we can conclude that none of the methods that we experimented, outperform others based on the disentanglement measures we use. We argue that the reason behind this is that all of the methods that we have discussed so far, enforce the same weight across all of the dimensions of the representation. We will discuss this phenomenon in Chapter 7, and propose a solution.

6.3.3 Experiment 3: Ablation Study: Investigating Learned Weights of L-VAE

In order to further investigate the effect of learning the hyperparameters of L-VAE on the reconstruction loss and the disentanglement, we train a standalone β-VAE model

with the learned weights as $\hat{\beta} = \sigma_0^2/\sigma_1^2$ as an ablation study. The results of these experiments are shown in Table 6.2 for all datasets.

Notice that the reconstruction loss of β-VAE trained with the learned parameter $\hat{\beta} = \sigma_0^2/\sigma_1^2$ is relatively higher than that of L-VAE. Moreover, training β-VAE with the $\hat{\beta}$ weights learned with L-VAE exhibits inferior performance than L-VAE regarding the disentanglement measures, which signifies the importance of a dynamic weighting strategy.

Table 6.2: An ablation study, where we train a standalone β-VAE with $\beta = \hat{\beta} = \sigma_0^2/\sigma_1^2$. The weights σ_i are learned from the suggested L-VAE model. The value of $\hat{\beta} = \sigma_0^2/\sigma_1^2$ are given in the second column. This setup results in a higher reconstruction loss and lower β-VAE measure than we have obtained with L-VAE, indicating the importance of a dynamic learning strategy.

Dataset	Model	Reconstruction↓	CbD ↑	β-VAE ↑	mig ↑	explicitness ↑
dSprites	β-VAE ($\beta = \hat{\beta}$)	21.43	0.58	0.93	**0.31**	**0.58**
	L-VAE ($\hat{\beta} = 0.89$)	**11.77**	**0.59**	**0.96**	**0.31**	0.57
MPI3D	β-VAE ($\beta = \hat{\beta}$)	**9.84**	**0.24**	0.71	0.19	0.45
	L-VAE ($\hat{\beta} = 0.92$)	**9.16**	0.22	**0.73**	**0.20**	**0.47**
Falcor3D	β-VAE ($\beta = \hat{\beta}$)	75.30	0.30	0.92	**0.15**	**0.67**
	L-VAE ($\hat{\beta} = 1.90$)	**70.67**	**0.32**	**0.94**	**0.15**	0.63
Isaac3D	β-VAE ($\beta = \hat{\beta}$)	11.75	0.41	**0.77**	**0.28**	**0.58**
	L-VAE ($\hat{\beta} = 0.88$)	**10.73**	**0.42**	0.75	0.26	0.56

6.4 Visual Inspection of the Learned Subspace

Following the common practice in the literature [9, 22, 57], we provide qualitative results for L-VAE on the CelebA dataset [39]. CelebA consists of 202K images of celebrity faces labelled with facial attributes. There are 40 facial attribute labels, including baldness, wearing eyeglasses, and pale skin. We cropped the background from the images and downscaled the dataset to size 128×128 following [9]. Further-

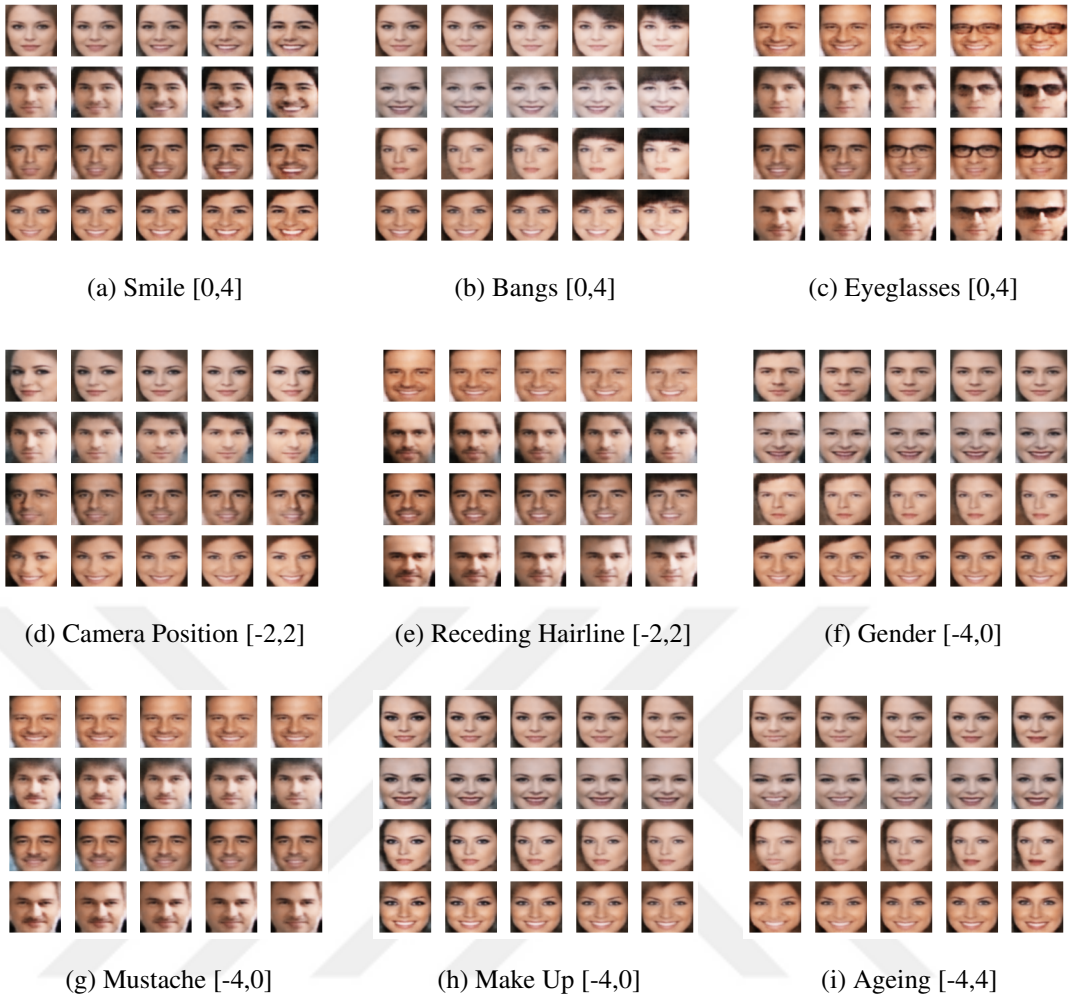
more, we use 12% of the dataset to train the L-VAE model for simplicity. We train the CNN Encoder-Decoder architecture defined in Table 3.1, with a batch size of 128 and a learning rate of 1e-4 for 1M iterations.

In order to assess the disentanglement ability of L-VAE visually, we analyze the latent-space traversals through reconstructions as suggested in the literature [9, 22, 57]. After encoding sample images, we acquire latent representations for images. We select 9 out of 40 informative dimensions of the representation experimentally. In order to visually inspect the effect of a specific dimension, we alter its dimension and keep the rest of the representation still, which is also referred to as latent traversals in the literature. After each alteration, we feed the altered representation to the decoder and inspect the decoded image visually.

Figure 6.2 shows the results of latent traversals for nine different dimensions. Each subimage shows the latent traversal of a single dimension. The traversal range is shown in brackets. For example, for Figure 6.2a, we altered the value of the dimension within the range [0,4]. We select these values experimentally to be able to present visually informative parts of the traversals. As shown in the Figure, altering a single latent dimension consequently changes a single attribute in facial images; for example, Figure 6.2i shows that people age as we alter the value of a specific dimension.

6.5 Chapter Summary

In this chapter, we propose Larnable VAE (L-VAE), which learns the trade-off between the reconstruction and KL Divergence terms of the loss function of the β-VAE model. We initialize learnable weights for these two terms and train them alongside weights of the architecture. We first analyse the learned weights of L-VAE. We show that the learned weights are aligned with the empirically optimized hyperparameter β of the β-VAE model. Moreover, we also compare L-VAE with five methods from the literature, namely VAE, β-VAE, ControlVAE, DynamicVAE, and σ-VAE, on four different datasets, dSprites, MPI3D, Falcor3D, and Isaac3D based on their reconstruction abilities, and disentanglement scores measured for four different measures.



(a) Smile [0,4] (b) Bangs [0,4] (c) Eyeglasses [0,4]

(d) Camera Position [-2,2] (e) Receding Hairline [-2,2] (f) Gender [-4,0]

(g) Mustache [-4,0] (h) Make Up [-4,0] (i) Ageing [-4,4]

Figure 6.2: Latent traversals of the L-VAE model for the CelebA dataset. We select a single dimension and acquire reconstructions while altering its value. Note that other dimensions are kept unchanged in these experimens. Each subfigure corresponds to the traversal of a single latent dimension. The ranges of the alteration are shown in brackets.

Although L-VAE can achieve superior or on-par disentanglement performances, we can not conclude any of the models outperform others based on these disentanglement measures. However, all of the models that we experimented, enforce the same weight on all of the dimensions of the representation. We argue that the dimensions of the learned representation can have different degrees of disentanglement. Therefore, they should have different weights. In order to achieve this we propose a new model, which we explain in the next chapter.

CHAPTER 7

MULTI DIMENSIONAL LEARNABLE VARIATIONAL AUTO-ENCODER

The L-VAE method, we propose in the previous chapter assumes that each dimension should have the same degree of disentanglement and applies the same weight on the KL Divergence term, which is responsible for disentangling the dimensions. However, our results show that different dimensions can have different degrees of disentanglement. We argue that enforcing the same weight on all of the dimensions can decrease both the reconstruction quality and the degree of disentanglement. In order to address this problem, we propose a new model called Multi-Dimensional Learnable VAE (mdL-VAE).

mdL-VAE, incorporates unique weight per dimension in the learning phase. Since the weights are dimension-specific, the loss terms of the entangled dimensions are not forced to have a higher degree of disentanglement and do not affect other dimensions. Similar to the L-VAE model, these unique weights are dynamically learned and do not require any hyperparameter tuning.

Moreover, we empirically show that learned dimension specific weights are direct indicators of the degrees of disentanglement in each dimension of the representation. We can assess the relationships between the learned weights and the degree of disentanglement of a dimension, comprehensively using the PCC matrix of CbD, since it measures dimension-wise entanglement.

7.1 Multi-Dimensional Learnable VAE (mdL-VAE)

Recall that, disentangled representation learning aims to identify and model the sources of variation in the data-generating process independently. However, the process of data generation in real-world scenarios is often hidden and the number of dimensions required to model the sources of variations is unknown. Furthermore, in these scenarios, it is difficult to accurately determine whether all of the factors are uniformly disentangled. Both β-VAE and L-VAE assume that all factors are equally disentangled, giving equal weight to each dimension of the representation. However, this assumption may not always hold in practice. To address this, we propose Multi-Dimensional Learnable VAE (mdL-VAE), which uses the same strategy as L-VAE but learns a different weight for each dimension. We use an encoder-decoder architecture to model an N-dimensional representation $\mathbf{z}$ for samples. We optimize the following loss function:

$$\begin{aligned}\mathcal{L}_{\text{mdL-VAE}} = &- g(\sigma_r)\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] \\ &+ \sum_{i=0}^{N} f_i(\sigma_i) D_{KL}\Big(q_\phi(z_i|\mathbf{x}) \parallel p(\mathbf{z})\Big) \\ &+ r_0(\sigma_r) + \sum_{i=0}^{N} r_i(\sigma_i),\end{aligned} \tag{7.1}$$

where $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big]$, parametrized by the decoder of the network, is the reconstruction loss and enforces the decoded images to be closer to the original image. We use MSE loss as reconstruction loss similar to prior study of Rybkin [54]. On the other hand, the second term $D_{KL}\Big(q_\phi(z_i|\mathbf{x}) \parallel p(\mathbf{z})\Big)$ minimizes the divergence between the variational distribution represented by the encoder and a predefined distribution. The functions $g(\sigma_r)$, and $f_i(\sigma_i)$ are weighting functions of σ_i, for reconstruction loss, and the dimensions of the representation respectively. Notice each term has a specific σ_i value, including the reconstruction loss. The final terms $r_i(\sigma_i)$ regularize all σ_i to prevent them from collapsing to zero. In the following section, we provide useful insights on how to select $f_i()$ and $r_i()$ functions.

We conjecture that each learned σ_i value signifies the relationship among underlying factors of variation. In other words, large values of σ_i indicate relatively weaker entanglement, allowing the associated attribute to be disentangled from the other at-

tributes. Conversely, if σ_i value of any dimension approaches to 0, the corresponding factor is strongly entangled, so that it is not possible to separate this attribute from the others.

We expect that the learned σ_i values to be an indicator of the disentanglement abilities of the corresponding dimension. Furthermore, analysing these values could provide us information on the necessary number of dimensions for a disentangled representation. We show that mdL-VAE achieves separation between the learned weights, i.e. learned weights are closely related to the entanglement of relative dimensions. We analyse the learned weights in Section 7.3.1.

7.2 Selection of Functions $f_i(\sigma_i)$ and $r_i(\sigma_i)$

The selection of the weighting function $f_i()$ and the regularization function $r_i()$ of the Equation 7.1 is the most important problem with mdL-VAE. Note that the values of σ_i are learned based on the derivatives of these functions. Therefore, the selection of these functions directly affect the learning curves of σ_i.

The selection of the weighting and regularization functions depends on the nature of the hidden attributes, embedded in the datasets. When the hidden attributes are rather strongly disentangled, we expect that $f_i(\sigma_i)$ functions yield high values. On the other hand, when the hidden attributes are rather entangled, the $f_i(\sigma_i)$ functions approach to zero. In order to enforce disentanglement we need a stronger regularize. In this section, we will discuss several functions. As a conclusion, we propose two main versions of Equation 7.1, and provide a guideline on how to select one of these versions.

The function $f_i(\sigma_i)$ is used to weight the terms of the loss function in Equation 7.1. Following L-VAE, the first choice for the weighting function $f_i(\sigma_i)$ is

$$f_i(\sigma_i) = \frac{1}{\sigma_i^2}.$$

However, during the experiments, we observed that some of the σ_i values converge to 0. Hence, rather than weighting each term with $1/\sigma_i^2$, as we performed in the L-VAE

model (Equation 6.2), we use,

$$f_i(\sigma_i) = 1/(1 + \ln(1 + \sigma_i^2),$$

as dimensional weights, to avoid

$$\lim_{\sigma_i \to 0} \frac{1}{\sigma_i^2} \to \infty.$$

Similarly, following L-VAE, the choice for regularization function $r_i(\sigma_i)$ is

$$r_i(\sigma_i) = \sigma_i^2,$$

which possess a strong regularization on the weights, σ_i.

Therefore we formulate the overall loss function for mdL-VAE as follows;

$$\begin{aligned} \mathcal{L}_{\text{mdL-VAE-strong}} = &- \frac{1}{1 + \ln(1 + \sigma_r^2)} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \Big[\log p_\theta(\mathbf{x}|\mathbf{z}) \Big] \\ &+ \sum_{i=0}^{N} \frac{1}{1 + \ln(1 + \sigma_i^2)} D_{KL} \Big(q_\phi(z_i|\mathbf{x}) \parallel p(\mathbf{z}) \Big) \\ &+ \sigma_r^2 + \sum_{i=0}^{N} \sigma_i^2, \end{aligned} \tag{7.2}$$

We explicitly notate this loss function as $\mathcal{L}_{\text{mdL-VAE-strong}}$ to be able to discriminate it with the second version of the loss function.

We expect the values of σ_i to be relatively high if they correspond to a disentangled dimension than the values of σ_i that correspond to an entangled dimension. In order to achieve this goal, our model should allow the weights σ_i to be able to increase as well as converge to 0. We experimentally show that if this requirement is not satisfied, both the reconstruction and disentanglement capabilities degrade. In such circumstances where the learned weights σ_i values converge to numbers such that all $\sigma_i < \epsilon$, in other words, if the σ_i values can not increase, we suggest to select a weaker function as a regularization function $r_i()$.

Figure 7.1 shows two alternatives for regularization functions. The blue curve shows the regularization function $r_i()$ we used for $\mathcal{L}_{\text{mdL-VAE-strong}}$. The orange curve, on the other hand, shows a weaker regularization function. Therefore, if the learned σ_i

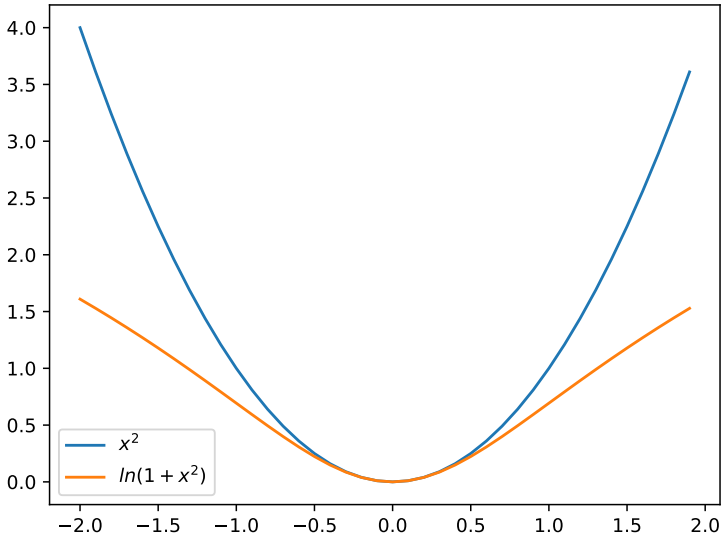


Figure 7.1: Different types of functions that can be used to regularize learned weights. The blue curve represents x^2, which provides higher penalties as the variable increases. However, the orange curve, which corresponds to $\ln(1+x^2)$, applies lower penalties and therefore allows the weights to increase.

values converge to values $\sigma_i < \epsilon$ with $\mathcal{L}_{\text{mdL-VAE-strong}}$ we suggest to use the following loss function which allows σ_i to increase through training.

$$\begin{aligned}\mathcal{L}_{\text{mdL-VAE-weak}} = &- \frac{1}{1+\ln(1+\sigma_r^2)}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] \\ &+ \sum_{i=0}^{N} \frac{1}{1+\ln(1+\sigma_i^2)} D_{KL}\Big(q_\phi(z_i|\mathbf{x}) \parallel p(\mathbf{z})\Big) \\ &+ \sigma_r^2 + \sum_{i=0}^{N} \frac{\ln(1+\sigma_i^2)}{N}.\end{aligned} \tag{7.3}$$

Note that in the above equation, we retain the regularization function for the σ_0 that weights the reconstruction loss as $r_0(\sigma_0) = \sigma_0^2$, since we need the change to affect only the disentanglement term.

In conclusion, we suggest using $\mathcal{L}_{\text{mdL-VAE-strong}}$ by default, and use $\mathcal{L}_{\text{mdL-VAE-weak}}$ when the first loss function produces $\sigma_i < \epsilon$. We empirically select $\epsilon = 0.5$. In future

work, we plan to investigate the effects of the dataset bias on function selection. Furthermore, one can select a wide range of functions as weighting and regularizing functions. In Section 7.3.4, we provide an ablation study by using different functions for weighting each dimension.

7.3 Experiments on the mdL-VAE

In this section we present the results of the experiments on mdL-VAE. We start with an analysis of the learned weights and their relationship to disentanglement. Later, we compare mdL-VAE with other DRL methods, mainly VAE, β-VAE, ControlVAE, DynamicVAE, and L-VAE, based on available disentanglement measures. The following section provides more comprehensive analysis on this specific dataset. Finally we provide an ablation study, showing the effect of different loss functions. We use cross validation to select learning rate, batch size and number of iterations. We use CbD score for selection. Refer to Chapter 3 for details on the experimental setup and datasets.

7.3.1 Experiment 1: Weights Learned by mdL-VAE

In this section, we analyse the learned weights of mdL-VAE.

Results on the dSprites dataset. For dSprites dataset, we used the strong loss function of mdL-VAE $\mathcal{L}_{\text{mdL-VAE-strong}}$ given in Equation 7.2. We first consider the learned weights for the dSprites dataset. Figure 7.2 shows the Pearson Correlation coefficients matrix $\mathbf{P}$ we obtained from CbD (See Chapter 5 on how to obtain $\mathbf{P}$). We observe that dimensions z_3 and z_4 are highly disentangled since they are strongly correlated with only one attribute, where these dimensions represent PosY ($P_{3,PosY} = 0.97$), and PosX ($P_{3,PosX} = 0.97$) attributes, respectively. The high correlations confirm the highest learned weights for these dimensions: $\sigma_3 = 0.76$ and $\sigma_4 = 0.76$. This value is followed by dimension z_2 by $\sigma_2 = 0.56$. Although dimension, z_2, appears to be representing the Scale attribute ($P_{2,Scale} = 0.88$), the Shape attribute is also captured in this dimension ($P_{2,Shape} = 0.41$). Therefore its degree of disentanglement (σ_2) is lower than those for z_3 and z_4. The remaining dimensions (z_0 and z_1) do not appear

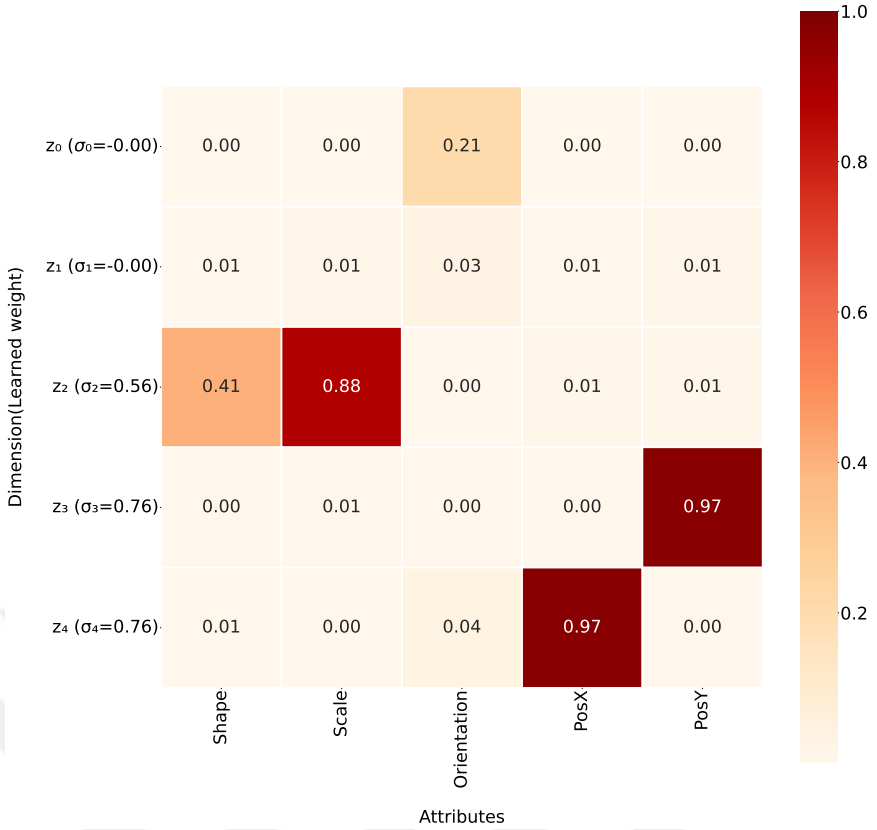


Figure 7.2: Pearson Correlation Matrix, where each entry shows the person correlation between variable z_i and the hidden attribute for optimal σ_i values (indicated in the paranthesis of each z_i) for dSprites dataset. High values indicate that dimensions are highly correlated with the corresponding attributes. The numbers on the left-hand side within parentheses show the learned weights σ_i.

to strongly represent any of the attributes, which is quantified by the weights learned by our method: $\sigma_0 = 0, \sigma_1 = 0$.

Figure 7.3 shows the same Pearson Correlation coefficients matrix $\mathbf{P}$ we obtained from CbD for L-VAE model. Notice the similarities between Figures 7.2 and 7.3. Both models can disentangle the same attributes to the same degree. However, mdL-VAE provides dimensional weights, which can be used to interpret different dimensionalities. For example, the value $\sigma_3 = 0.76$ is an indicator that dimension z_3 is more disentangled than z_0 with $\sigma_0 = 0$. Therefore, we can conclude that mdL-VAE increases the interpretability of the overall model, compared to the baseline model of L-VAE.

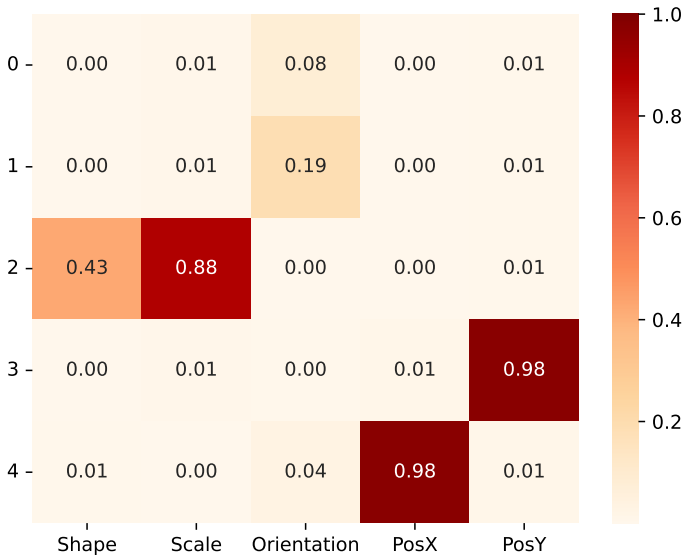


Figure 7.3: Pearson Correlation Matrix $\mathbf{P}$ (See Chapter 5) estimated for L-VAE model for dSprites dataset. High values indicate that dimensions are highly correlated with the corresponding attributes.

Results on the Falcor3D dataset. For Falcor3D dataset, we use the strong loss function of mdL-VAE $\mathcal{L}_{\text{mdL-VAE-strong}}$ given in Equation 7.2, since it can learn $\sigma_i > \epsilon$. Figure 7.4 shows the results of the same experiment explained for dSprites dataset, applied to the Falcor3D dataset. Dimension z_5 has the highest correlation with an attribute ($P_{5,camera\_y} = 0.93$). Moreover, this dimension is not correlated with other attributes indicating that it is nicely disentangled from other dimensions. The corresponding learned weight, $\sigma_5 = 0.77$, is also the highest among the learned weights, supporting our argument that the learned weights are indicative of the disentanglement degrees of dimensions. Other attributes except *lighting intensity* are also learned in separate dimensions. However, their degree of disentanglement is lower than *camera_y* since their correlation values are lower.

Results on the Isaac3D dataset. For Isaac3D dataset, we use the strong loss function of mdL-VAE $\mathcal{L}_{\text{mdL-VAE-strong}}$ given in Equation 7.2. Figure 7.5 provides the results of the same experiment for the Isaac3D dataset. We see that the distribution among the learned σ_i values in the Isaac3D dataset is similar to that of the dSprites dataset. Dimensions z_0, z_1, z_2, z_7 and z_8 are strongly correlated with only one attribute, indicating high degree of disentanglement, which is captured by high σ_i values. On the

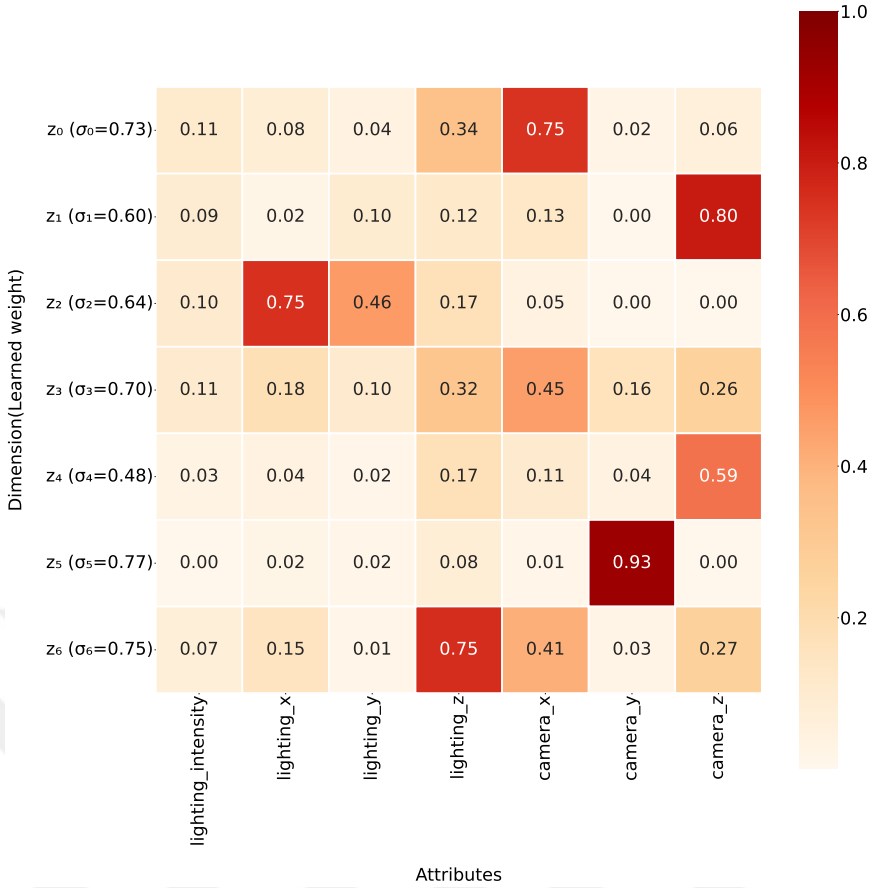


Figure 7.4: Pearson Correlation Matrices $\mathbf{P}$ (See Chapter 5) versus the learned dimensional weights, σ_i of mdL-VAE (See Equation 7.1) for Falcor3D dataset. High values indicate that dimensions are highly correlated with the corresponding attributes. The numbers on the left-hand side within parentheses show the learned weights σ_i.

other hand, dimensions z_3, z_4, and z_5 are not correlated with any of the attributes, with σ values equal to 0.

Results on the MPI3D-Complex dataset. For MPI3D dataset, we start with the strong loss function of mdL-VAE $\mathcal{L}_{\text{mdL-VAE-strong}}$ given in Equation 7.2. We first analyse the weights learned by mdL-VAE model on this dataset in Figure 7.6. We see that all of the learnt weights σ_i are less than 0.5 (See the σ_i values given in parentheses in the y-axis in Figure 7.6). Therefore we repeat the experiment with the loss function with a weak regularization $\mathcal{L}_{\text{mdL-VAE-weak}}$ given in Equation 7.3.

The results we obtain with the weak regularization $\mathcal{L}_{\text{mdL-VAE-weak}}$ given in Equation 7.3 are slightly better in terms of disentanglement. Figure 7.7 shows the Correlation

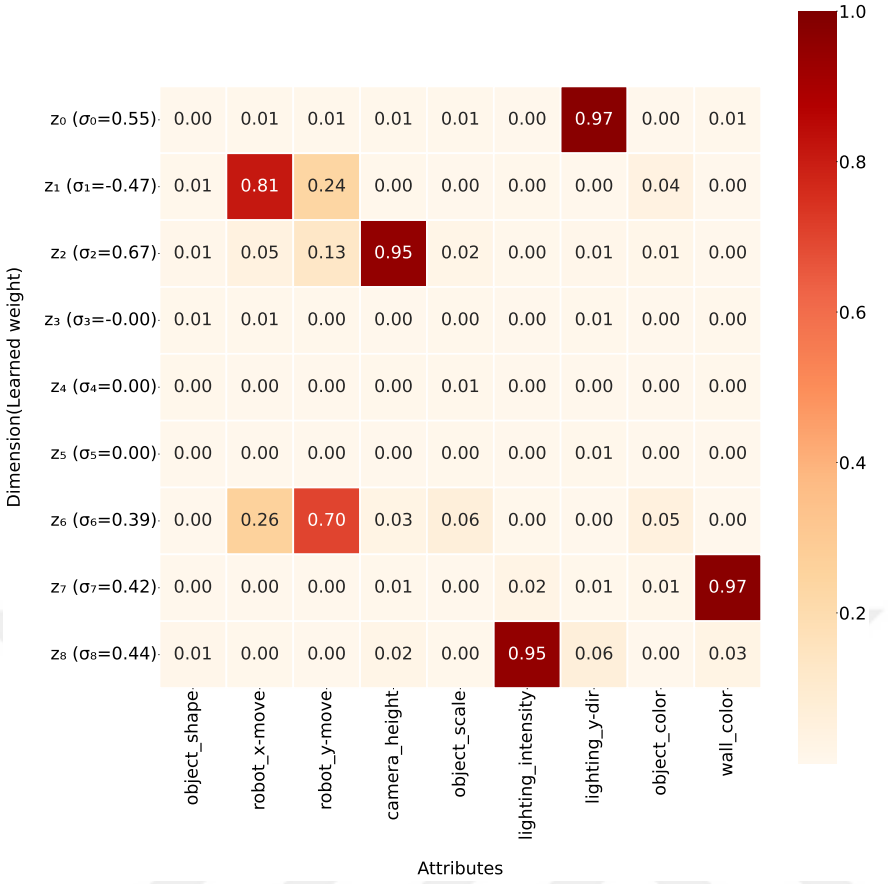


Figure 7.5: Pearson Correlation Matrix $\mathbf{P}$ (See Chapter 5) versus the learned dimensional weights, σ_i of mdL-VAE (See Equation 7.1) for Isaac3D dataset. High values indicate that dimensions are highly correlated with the corresponding attributes. The numbers on the left-hand side within parentheses show the learned weights σ_i.

matrix $\mathbf{P}$ of this experiment. Notice the correlation values for the attributes *object color*, *horizontal axis,* and *vertical axis* are higher than we have obtained with mdL-VAE-strong which is given in Figure 7.6. On the other hand, it is important to note that both models disentangle *Camera height* relatively better than other attributes.

The results on four different datasets suggest that the learned σ_i values align very well with the observed disentanglement properties of the dimensions and that the σ_i values can be used as a tool to analyse the disentanglement properties of dimensions.

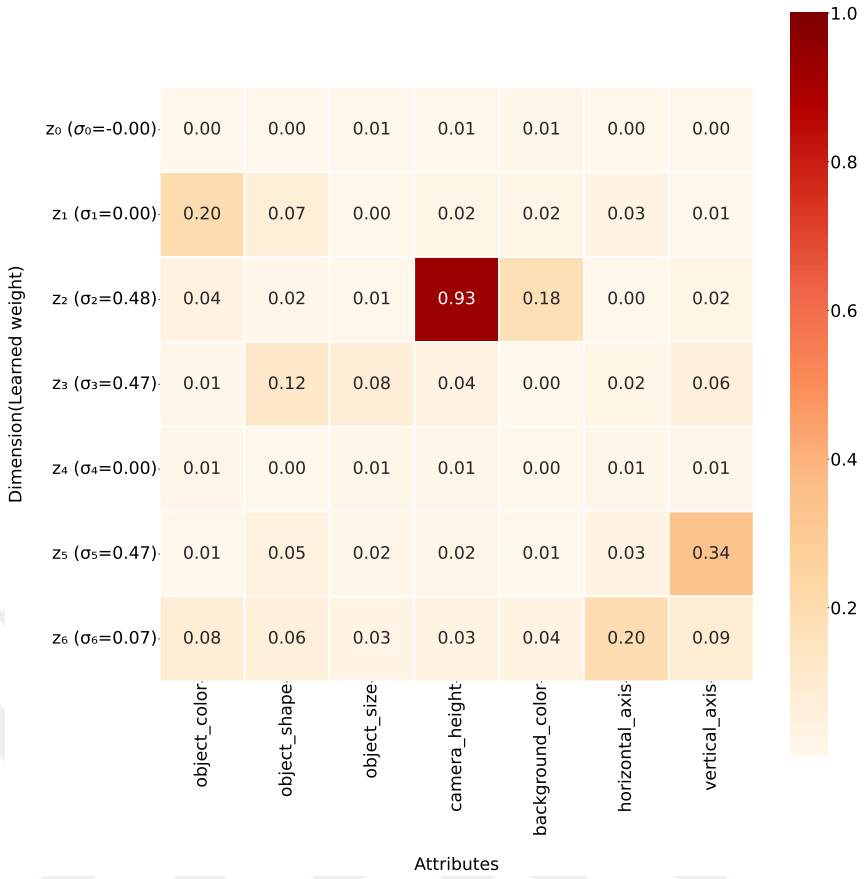


Figure 7.6: Pearson Correlation Matrix $\mathbf{P}$ (See Chapter 5) versus the learned dimensional weights, σ_i of mdL-VAE-strong (See Equation7.2) for MPI3D dataset. Note that only a single dimension z_2 could be sufficiently disentangled.

7.3.2 Experiment 2: Comparison with Existing Methods

In this subsection, we compare the proposed mdL-VAE with six disentanglement methods available in the literature: VAE, β-VAE [22], ControlVAE [58], DynamicVAE [57], σ-VAE [54], and L-VAE [45] (See Section 2.3.3.2 for the details about these models), on four different datasets; dSprites, Falcor3D, Isaac3D, and MPI3D. See Section 3.1 for details on the datasets.

Note that for dSprites, Falcor3D, and Isaac3D datasets, we use mdL-VAE-strong version of the model, for which the loss function is given in Equation 7.2. In this model, we use stronger regularization terms to control the weights σ_i. However, for the MPI3D dataset, the values of σ_i converged to values that are less than 0.5, but we

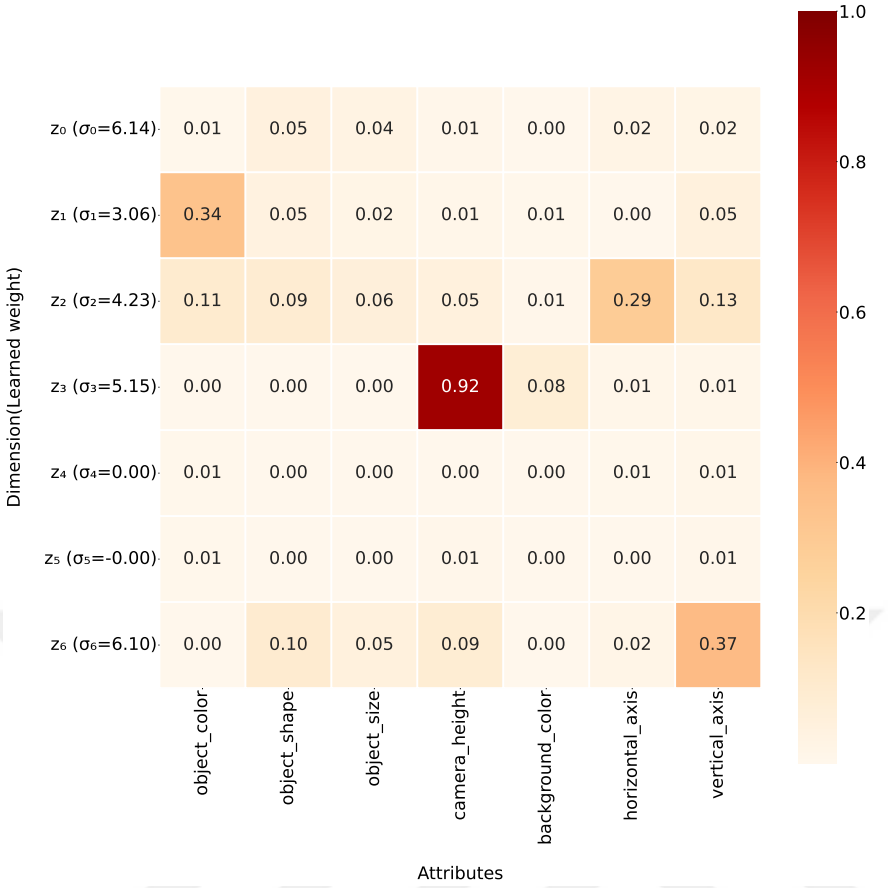


Figure 7.7: Pearson Correlation Matrix $\mathbf{P}$ (See Chapter 5) versus the learned dimensional weights, σ_i of mdL-VAE-strong (See Equation7.3) for MPI3D dataset. Note that only a single dimension z_3 could be sufficiently disentangled.

expect these values to be as diverse as possible to assure the disentanglement. Therefore, we used mdL-VAE-weak for this dataset.

For model selection, we use CbD. Recall that we have several hyperparameters to tune; namely learning rate, batch size, number of iterations, and β for β-VAE model. We use cross validation to select these hyperparameters with CbD as the objective. Section 3.2.2 gives details on model selection. Note that CbD (Equation 5.9) averages modularity ($\mathcal{M}_{\text{modularity}}$ in Equation 5.7) and compactness ($\mathcal{M}_{\text{compactness}}$ in Equation 5.6). We can use either of these measures for model selection. However, we use the composite CbD (See Equation 5.9) for simplicity.

Table 7.1: Experiment 2: Quantitative comparison of our model across related models. The best models are selected via CbD.

Dataset	Model	Recon↓	CbD ↑	β-VAE ↑	mig ↑	explicitness ↑
dSprites	VAE	**11.86**	0.58	0.93	0.30	0.56
	β-VAE ($\beta = 4$)	29.11	0.58	0.92	0.30	0.55
	ControlVAE	24.35	**0.60**	**0.97**	0.30	**0.59**
	DynamicVAE	33.75	0.56	0.89	**0.33**	0.50
	σ-VAE	12.30	0.29	0.77	0.07	0.47
	L-VAE ($\hat{\beta} = 0.89$)	**11.77**	0.59	0.96	0.31	0.57
	mdL-VAE-strong (Ours)	19.99	**0.60**	**0.97**	0.31	**0.59**
Falcor3D	VAE	68.15	0.27	0.93	0.13	**0.64**
	β-VAE ($\beta = 2$)	71.37	0.37	0.91	0.16	0.62
	ControlVAE	237.69	0.39	0.87	0.15	0.45
	DynamicVAE	205.04	0.24	0.77	0.16	0.34
	σ-VAE	**65.60**	0.20	0.87	0.06	0.61
	L-VAE ($\hat{\beta} = 1.90$)	70.67	0.32	0.94	0.15	0.63
	mdL-VAE-strong (Ours)	75.17	**0.41**	**0.95**	**0.17**	0.61
Isaac3D	VAE	11.88	0.48	0.76	0.32	**0.59**
	β-VAE ($\beta = 2$)	14.71	0.50	**0.77**	0.33	0.55
	ControlVAE	34.56	0.26	0.65	0.15	0.46
	DynamicVAE	39.80	0.19	0.51	0.13	0.40
	σ-VAE	26.31	0.20	0.61	0.04	0.54
	L-VAE ($\hat{\beta} = 0.88$)	**10.73**	0.42	0.75	0.26	0.56
	mdL-VAE-strong (Ours)	13.78	**0.52**	0.76	**0.37**	0.57
MPI3D	VAE	9.91	**0.23**	0.75	0.18	0.40
	β-VAE ($\beta = 2$)	16.12	0.21	0.65	0.15	0.36
	ControlVAE	19.74	0.22	0.62	0.12	0.36
	DynamicVAE	18.46	**0.23**	0.69	0.14	0.36
	σ-VAE	12.70	0.22	0.65	0.05	**0.49**
	L-VAE ($\hat{\beta} = 0.92$)	9.16	0.22	0.73	**0.20**	0.47
	mdL-VAE-strong (Ours)	16.60	0.21	0.54	0.12	0.34
	mdL-VAE-weak (Ours)	**8.80**	**0.23**	**0.79**	0.19	0.41

We make comparisons based on the MSE reconstruction loss and the four disentanglement measures. To assess the disentanglement performances of the different models, we examine their disentanglement properties separately for each dataset. In addition to CbD measure, which measures both modularity and compactness, we also include alternative measures β-VAE, MIG, and Explicitness score to quantify modularity, compactness and explicitness, respectively. For the β-VAE model, the β value is determined through an empirical hyperparameter selection process, whereas, for L-VAE and mdL-VAE models, the values of σ_i represent the learned values.

Table 7.1 shows the reconstruction and disentanglement scores for this experiment. Note that the lower values are better for reconstruction, and the higher values are better for all of the disentanglement measures. The best value for each measure is represented as bold fonts, and the underlined values represent the second-best performance. For dSprites, Falcor3D, and Isaac3D datasets we report the results of mdL-VAE-strong model. On the other hand, for MPI3D dataset we report the results of both the mdL-VAE-strong and mdL-VAE-weak models.

We observe that mdL-VAE provides the best performance on all datasets when we compare them based on CbD measure. Recall that CbD measures both modularity and compactness by averaging $\text{CbD}_{\text{Modularity}}$ $\text{CbD}_{\text{Compactness}}$ scores. Moreover, mdL-VAE achieves best or second-best performances on both β-VAE and MIG scores which also measure modularity and compactness, respectively. As for the explicitness score, mdL-VAE achieves the highest and second-highest results for dSprites and Isaac3D datasets. However, the model is outperformed by others in the Falcor3D dataset (See Falcor3D-mdL-VAE vs explicitness score in row 14 = 0.61). This result is expected since we conduct model selection based on modularity and compactness. For the MPI3D dataset, the mdL-VAE-strong algorithm produces lower results than the mdL-VAE-weak algorithm in terms of disentanglement, as expected since the mdL-VAE-strong algorithm does not allow the values of σ_i to increase. We plan to extend this experiment so that we can analyse the effect of regularization in relation to the dataset properties. We discuss this in the next chapter under the future directions of this study.

Overall, we observe that learning dimension-wise σ_i is a promising approach to estimate disentangle representation by Variational Auto-Encoders. mdL-VAE achieves

superior or on par performance with respect to many measures on all datasets, also has the benefit of providing insight on disentanglement degrees of dimensions through learned weights. In this section we analyze the results of mdL-VAE with only four disentanglement measures for simplicity. Recall that we used CbD is the average of $\text{CbD}_{\text{Modularity}}$ $\text{CbD}_{\text{Compactness}}$ scores. In the next section we discuss the results of model selection depending on only the $\text{CbD}_{\text{Modularity}}$ or $\text{CbD}_{\text{Compactness}}$ scores.

7.3.3 Experiment3: Ablation Study on Model Selection

In the previous section, we compare mdL-VAE with the models from the literature. We used the holistic CbD, which equally weights the modularity ($\text{CbD}_{\text{Modularity}}$) and compactness ($\text{CbD}_{\text{Compactness}}$) scores of a given model. In this section, we separate the modularity and compactness. In other words, we analyze these two properties separately. If the goal of any given task is to increase a specific disentanglement property, particularly, modularity, compactness or explicitness, the hyperparameter selection and model comparison should be carried out with this approach.

We first analyse modularity scores, which measures whether a change of an attribute affects only a single dimension or not. To this end, we select the hyperparameters using a cross-validation (See Section 3.2.2 for hyperparameters). In the cross-validation, we select the hyperparameter set that maximizes $\text{CbD}_{\text{modularity}}$ given in Equation 5.7, which measures solely modularity. Given the best hyperparameters, we should also compare the models based on disentanglement scores that measure modularity property of a given model. Table 7.2 provides the comparison of the models based on this experimental setup. This table provides the results for several disentanglement measures, including β-VAE, FactorVAE, MIG_SUP, and Modularity score, where all of them quantify the modularity property(See Section 5.1 for details on the measures). Note that, we include holistic CbD results, which also include compactness, for comparison. As Table 7.2 shows, our proposed models,L-VAE and mdL-VAE can achieve the best or second-best performances. However, it is more important to state that this selection produces different results than we had in the previous section. For example, when we compare the mdL-VAE results for the dSprites datasets in this experiment, given in Table 7.2, and the previous experiment given in

Table 7.2: Experiment 3: Quantitative comparison of our model across related models. The best models are selected via $CbD_{\text{Modularity}}$. Other disentanglement measures also measure the modularity of a given model.

Dataset	Model	Reconstruction↓	CbD ↑	$CbD_{\text{Modularity}}$ ↑	β-VAE↑	FactorVAE↑	MIG_SUP↑	Modularity↑
dSprites	VAE	**11.86**	0.58	0.66	0.93	**0.77**	**0.32**	0.92
	β-VAE	29.11	0.58	0.66	0.92	0.75	0.28	**0.94**
	ControlVAE	24.35	**0.60**	0.67	**0.97**	0.76	0.30	**0.94**
	DynamicVAE	33.44	0.56	0.65	0.89	0.61	0.29	0.86
	σ-VAE	12.30	0.29	0.28	0.77	0.55	0.09	0.90
	L-VAE	**11.77**	0.59	0.66	0.96	**0.77**	**0.32**	0.92
	mdL-VAE-strong	17.14	**0.60**	**0.68**	**0.97**	**0.77**	0.29	**0.94**
Falcor3D	VAE	68.15	0.27	0.31	0.93	0.59	0.13	0.89
	β-VAE	90.80	0.36	0.35	0.89	**0.77**	0.17	0.91
	ControlVAE	237.69	0.39	**0.40**	0.87	0.53	0.14	0.80
	DynamicVAE	205.04	0.24	0.30	0.77	0.45	0.06	0.73
	σ-VAE	**65.60**	0.20	0.15	0.87	0.42	0.09	0.90
	L-VAE	70.67	0.32	0.36	0.94	0.70	0.17	0.95
	mdL-VAE-strong	75.17	**0.41**	0.37	**0.95**	0.65	**0.20**	**0.97**
Isaac3D	VAE	11.88	0.48	0.48	0.76	**0.69**	0.31	0.90
	β-VAE	14.71	0.50	0.52	**0.77**	0.65	0.32	0.90
	ControlVAE	34.56	0.26	0.25	0.65	0.48	0.15	0.95
	DynamicVAE	28.73	0.19	0.20	0.48	0.32	0.07	0.79
	σ-VAE	16.57	0.18	0.14	0.64	0.47	0.18	**0.96**
	L-VAE	**10.73**	0.42	0.41	0.75	0.66	0.26	0.91
	mdL-VAE-strong	13.78	**0.52**	**0.53**	0.76	0.67	**0.36**	0.93
MPI3D	VAE	9.91	**0.23**	0.23	0.75	0.39	0.10	0.90
	β-VAE	14.58	0.21	0.25	0.67	0.45	0.10	0.88
	ControlVAE	19.74	0.22	0.21	0.62	0.38	0.11	0.90
	DynamicVAE	17.81	**0.23**	0.24	0.69	0.45	**0.13**	**0.91**
	σ-VAE	11.24	0.22	0.14	0.65	0.35	0.10	0.88
	L-VAE	9.16	0.22	**0.26**	0.73	0.41	0.10	0.90
	mdL-VAE-strong	14.35	0.21	0.24	0.57	0.36	0.08	0.87
	mdL-VAE-weak	**8.73**	**0.23**	0.24	**0.80**	**0.53**	0.07	0.89

Table 7.1 where we selected hyperparameters based on modularity and compactness, we can see that these two experiments select different models (See the results of reconstruction losses are different in the two). We can conclude that the selection of measures for comparisons and the selection of hyperparameters are important problems in this area.

Finally, we repeat the same experiment for the compactness property, which measures whether a single dimension represents a single attribute or not. We use $\text{CbD}_{\text{compactness}}$ given in Equation 5.6 for hyperparameter selection. Table 7.3 shows the comparisons on disentanglement measures which focus on compactness. The most important outcome of this experiment is the shift in hyperparameter selection. When we select the hyperparameters based on compactness measures, we observe different outcomes than the previous experiment with modularity-based optimization. Notice that in Table 7.3, the reconstruction values are relatively higher than the values in Table 7.2. Although both of these measures can be used for cross-validation, it is important to note that selecting a measure for comparison can affect the outcome of the experiment, and the measure can be decided dataset or model-dependent.

7.3.4 Experiment 4: Ablation Study on Loss Function Selection

Recall that, in Equation 7.1, we define a general formula for the mdL-VAE loss function, where we use a weighting function $f(\sigma_i)$ to weight each term and a regularize $r(\sigma_i)$ to regularize the learnable parameter σ_i. In Section 7.2 we provide specifications for functions $f(\sigma_i)$ and $r(\sigma_i)$, and propose two different versions for these functions in Equations 7.2 and 7.3, namely mdL-VAE-strong and mdL-VAE-weak. These instance functions can be further elaborated. In this section we experiment with three more functions to provide possible outcomes of different loss functions. Note that we only used the dSprites dataset for brevity. The results are provided in Table 7.4. Each row shows the reconstruction and disentanglement scores for different function combinations. We see from the table that alternative configurations provide lower performance compared to the definition of mdL-VAE in Equation 7.2. In other words, mdL-VAE-strong provides the best results in terms of disentanglement and reconstruction compared its the variations considered in the table. However, several

Table 7.3: Experiment 3: Quantitative comparison of our model across related models. The best models are selected via $CbD_{\text{Compactness}}$. Other disentanglement measures also measure the compactness of a given model.

Dataset	Model	Reconstruction↓	CbD ↑	$CbD_{\text{Compactness}}$ ↑	MIG↑	SAP↑
dSprites	VAE	<u>11.11</u>	0.58	0.51	**0.31**	<u>0.30</u>
	β-VAE	28.74	0.58	0.51	<u>0.30</u>	0.26
	ControlVAE	24.35	**0.60**	<u>0.52</u>	<u>0.30</u>	<u>0.30</u>
	DynamicVAE	36.68	0.56	0.48	**0.31**	0.27
	σ-VAE	**7.99**	0.28	0.32	0.09	0.04
	L-VAE	21.58	<u>0.59</u>	**0.53**	0.29	0.25
	mdL-VAE-strong	19.99	**0.60**	<u>0.52</u>	**0.31**	**0.33**
Falcor3D	VAE	80.54	0.25	0.25	0.11	0.08
	β-VAE	71.37	0.37	<u>0.39</u>	**0.16**	<u>0.12</u>
	ControlVAE	237.69	**0.39**	0.38	<u>0.15</u>	**0.15**
	DynamicVAE	279.33	0.22	0.19	0.09	0.10
	σ-VAE	**65.60**	0.20	0.26	0.06	0.05
	L-VAE	<u>70.67</u>	0.32	0.29	<u>0.15</u>	<u>0.12</u>
	mdL-VAE-strong	90.39	<u>0.38</u>	**0.45**	0.13	0.11
ISaac3D	VAE	<u>11.88</u>	0.48	0.47	0.32	<u>0.27</u>
	β-VAE	14.71	<u>0.50</u>	<u>0.49</u>	<u>0.33</u>	<u>0.27</u>
	ControlVAE	36.42	0.26	0.28	0.15	0.13
	DynamicVAE	42.55	0.18	0.19	0.12	0.12
	σ-VAE	13.59	0.19	0.31	0.03	0.02
	L-VAE	**10.73**	0.42	0.42	0.26	0.23
	mdL-VAE-strong	13.78	**0.52**	**0.52**	**0.37**	**0.31**
MPI3D	VAE	<u>9.91</u>	**0.23**	0.22	**0.18**	**0.24**
	β-VAE	19.66	<u>0.22</u>	0.21	0.11	0.10
	ControlVAE	18.19	0.21	0.22	0.12	<u>0.11</u>
	DynamicVAE	18.46	**0.23**	<u>0.24</u>	<u>0.14</u>	0.13
	σ-VAE	**7.77**	0.21	**0.30**	0.03	0.03
	L-VAE	24.34	0.19	0.21	0.12	0.10
	mdL-VAE-strong	18.85	0.15	0.21	0.04	0.04
	mdL-VAE-weak	<u>9.34</u>	**0.23**	0.23	**0.18**	

other functions can be used for these functions. Moreover, as we show for MPI3D dataset, regularization function can be dataset-dependent.

Table 7.4: Experiment 4: Ablation study on different loss functions for mdL-VAE. Note that $\mathcal{L}_R = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big]$ and $\mathcal{L}_{KL} = D_{KL}\Big(q_\phi(\mathbf{z}_{i-1}|\mathbf{x}) \parallel p(\mathbf{z})\Big)$.

Loss Name	Formulation	Reconstruction↓	CbD ↑	β-VAE ↑	explicitness ↑	mig ↑
mdL-VAE-strong	$-\frac{1}{1+\ln(1+\sigma_r^2)}\mathcal{L}_R + \sum_{i=0}^{N}\frac{1}{1+\ln(1+\sigma_i^2)}\mathcal{L}_{KL} + \sigma_r^2 + \sum_{i=0}^{N}\sigma_i^2.$	**19.99**	**0.60**	**0.97**	**0.31**	**0.59**
mdL-VAE-v1	$-\frac{1}{1+\ln(1+\sigma_r^2)}\mathcal{L}_R + \sum_{i=0}^{N}\frac{1}{1+\ln(1+\sigma_i^2)}\mathcal{L}_{KL} + \ln(1+\sigma_r^2) + \sum_{i=0}^{N}\ln(1+\sigma_i^2).$	29.38	0.58	0.94	0.30	0.56
mdL-VAE-v2	$-\frac{1}{\sigma_r^2}\mathcal{L}_R + \sum_{i=0}^{N}\frac{1}{\sigma_i^2}\mathcal{L}_{KL} + \sigma_r^2 + \sum_{i=0}^{N}\sigma_i^2.$	37.86	0.55	0.88	0.29	0.49
mdL-VAE-v3	$-\frac{1}{\sigma_r^2}\mathcal{L}_R + \sum_{i=0}^{N}\frac{1}{\sigma_i^2}\mathcal{L}_{KL} + \ln(1+\sigma_r^2) + \sum_{i=0}^{N}\ln(1+\sigma_i^2).$	36.13	0.55	0.87	**0.31**	0.50

7.4 Chapter Summary

In this study, we propose Multi-Dimensional Learnable VAE (mdL-VAE) as an extension to β-VAE with the argument that the attributes (or factors) hidden in a dataset can have different levels of entanglement and a model should be trained accordingly, allowing dimensions to represent the strength of these attributes. mdL-VAE, learns independent weights per dimension, assuming each attribute has a different degree of disentanglement, unlike its counterparts which assume the same level of disentanglement per dimension (See Section 2.3.3.2). We also argue that the learned weights depict the degree of disentanglement, allowing user to easily interpret the learned representation. In order to assess the weights learned by mdL-VAE, we propose a novel correlation-based disentanglement measure CbD, which measures the degree of disentanglement based on pairwise correlations between the dimensions of the learned representation and the dataset attributes.

Our experiments show that the values of the learned per-dimension weights success-

fully show the degree of disentanglement in each dimension. Moreover, mdL-VAE produces on-par or better disentanglement scores based on our experiments on four different datasets. Moreover, we show that the weights learned by the mdL-VAE model are direct indicators of the disentanglement in independent dimensions. We plan to investigate this relationship between learned weights and the properties of dimensions with a varying number of dimensions.

CHAPTER 8

CONCLUSION

In this thesis, we focus on a fundamental problem in representation learning, called disentanglement, wherein the aim is to obtain a representation of the underlying sources of variation within a data-generating process in an independent and compact manner. We base our study on β-VAEs, one of the pioneering studies in disentangled representation learning. We highlight its significant limitations, while also proposing established solutions to mitigate them.

We first provide a comprehensive analysis on β-VAEs, which weights the Kullback-Leibler term of Variational Auto Encoders with a hyperparameter β. Our analyses in Chapter 4 show that increasing the hyperparameter β provides a balance between reconstruction and KL terms of the loss function. However, high values of β increase the reconstruction loss. Furthermore, introduces an additional hyperparameter that complicates the optimization process. Higgins et al. proposed that β should be greater than one [22]. However, our analysis shows that β can be smaller than one for some datasets, as opposed to the previous proposal of Higgins et al. [22]. Our observations from these analyses steered our research towards developing solutions for mitigating the potential limitations of the β-VAE: We proposed two novel extensions to eliminate the complex hyperparameter optimization problem while enhancing the reconstruction quality.

In our proposed models, we aim to increase reconstruction quality with fewer hyperparameters while increasing the disentanglement properties of the model. In order to achieve this goal, we require a disentanglement measure. There are several disentanglement measures proposed in the literature. We discuss these measures in 5.1. All of these measures have dependencies. For example, some of them use a linear

classifier, and the outcome depends on the classifier selection, while some compute mutual information, and the outcome depends on the estimation method for the mutual information (See Table 5.1 for details on the dependencies). However, we require a more robust disentanglement measure to compare the models that we propose. In order to achieve this goal, we propose a novel correlation-based disentanglement measure called CbD in Chapter 5. CbD measures the disentanglement based on pairwise correlations between the dimensions of the learned representation and the dataset attributes. Since CbD uses Pearson correlation values, we can robustly measure disentanglement. Besides providing a measure for disentanglement, CbD can be used as a visual aid to interpret the learned dimensions. Through the heatmap of the Pearson Correlation values, we can explain important properties of disentanglement, modularity and compactness. In other words, we can interpret whether a single dimension learns more than one attribute or whether a single attribute is represented in a single dimension.

One of the main problems with β-VAE is that it introduces a new hyperparameter β to the loss function, where empirical optimization of the hyperparameter is a computationally exhaustive problem. In order to eliminate the hyperparameter selection problem, we propose a new model called Learnable VAE (L-VAE), which learns the hyperparameter β dynamically along with the network parameters. We introduce two learnable parameters, which weight the reconstruction loss and KL Divergence terms of the loss function of β-VAE. Since these weight are learned along with the architecture weights, L-VAE successfully eliminates the hyperparameter selection problem. We showed that the learned weights are aligned with the empirically optimized β parameter of β-VAE model, which can only be found by hyperparameter search. In other words, we can find the optimal trade-off between the terms of the loss function without requiring any hyperparameter optimization. Moreover, we show that in three of the four datasets we used, L-VAE provides lower reconstruction losses. In terms of disentanglement measures, in almost 70% of the cases, L-VAE achieves the highest or second-highest disentanglement scores. We also analyse the learned representation and show that some of the attributes are inherently entangled. Although we use the same weights on all of the dimensions, we can not force some dimensions to disentangle. This observation leads to our second model.

The second model we proposed is called Multi-Dimensional Learnable VAE(mdL-VAE). Similar to L-VAE, mdL-VAE also learns the weights of the loss function. However, mdL-VAE learns independent weights per each dimension in the representation, assuming each attribute has a different degree of disentanglement. Our experiments show that the learned per-dimension weights indicate degrees of disentanglement in each dimension. In other words, mdL-VAE learns higher weights for disentangled dimensions; on the other hand, learned weights converge to 0 for entangled dimensions. Since learned weights are indicators of the disentanglement, mdL-VAE increases the interpretability of the model. Both L-VAE and mdL-VAE produce on-par or better disentanglement scores based on our experiments on four different datasets.

8.1 Limitations of the Proposed Models and Future Work

We conduct our experiments on several datasets to evaluate the performance of our proposed models. However, for the sake of fair comparison with other disentanglement models, we use the same architecture Burgess et al. proposed [6]. This is a relatively shallow network architecture. Using a shallow network may limit the generalization and scalability of our models to more complex and diverse datasets. Therefore, in the future, we plan to carry out a more comprehensive comparison with deeper network architectures. We also plan to experiment with new datasets.

We proposed two versions for mdL-VAE with a strong and weak regularization on the learned weights. We argue that the regularization function should be dataset-specific. We plan to analyse dataset properties to investigate the effect of different regularization functions.

We also set the number of dimensions equal to the number of attributes in the datasets. We intend to vary the size of the representation and examine its impact on the outcome. As we stated, the weights learned by the mdL-VAE model are direct indicators of the disentanglement in independent dimensions. We plan to investigate the behaviour of the learned weights with a varying number of dimensions of the hidden variables.

REFERENCES

[1] K. J. Åström and T. Hägglund. *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society, 2006.

[2] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769, 2014.

[3] Y. Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[5] L. Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[6] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[7] M.-A. Carbonneau, J. Zaidi, J. Boilard, and G. Gagnon. Measuring disentanglement: A review of metrics. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[8] J. Chen and K. Batmanghelich. Weakly supervised disentanglement by pairwise similarities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3495–3502, 2020.

[9] R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

[10] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[11] E. L. Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*, pages 4414–4423, 2017.

[12] N. S. Detlefsen and S. Hauberg. Explicit disentanglement of appearance and perspective in generative models. *arXiv preprint arXiv:1906.11881*, 2019.

[13] Z. Ding, Y. Xu, W. Xu, G. Parmar, Y. Yang, M. Welling, and Z. Tu. Guided variational autoencoder for disentanglement learning. *arXiv*, pages 7920–7929, 2020.

[14] K. Do and T. Tran. Theory and Evaluation Metrics For Learning Disentangled Representations. 2020.

[15] Y. Dong, Y. Liu, H. Zhang, S. Chen, and Y. Qiao. Fd-gan: Generative adversarial networks with fusion-discriminator for single image dehazing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10729–10736, 2020.

[16] B. Duan, C. Fu, Y. Li, X. Song, and R. He. Cross-spectral face hallucination via disentangling independent factors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7930–7938, 2020.

[17] S. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, G. E. Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pages 3225–3233, 2016.

[18] M. Fil, M. Mesinovic, M. Morris, and J. Wildberger. beta-vae reproducibility: Challenges and extensions. *arXiv preprint arXiv:2112.14278*, 2021.

[19] Z. Fu, X. Chen, D. Liu, X. Qu, J. Dong, X. Zhang, and S. Ji. Multi-level feature disentanglement network for cross-dataset face forgery detection. *Image and Vision Computing*, 135:104686, 2023.

[20] M. W. Gondal, M. Wüthrich, Đ. Miladinović, F. Locatello, M. Breidt, V. Volchkov, J. Akpo, O. Bachem, B. Schölkopf, and S. Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *arXiv preprint arXiv:1906.03292*, 2019.

[21] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

[22] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016.

[23] A. S. Hornby. *Oxford advanced learner's dictionary of current English. 7th edition / wörterbuch mit exam trainer*. Oxford Advanced Learner's Dictionary of Current English. 7th Edition. Cornelsen Verlag, Berlin, Germany, May 2005.

[24] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Reviews of modern physics*, 81(2):865, 2009.

[25] J.-T. Hsieh, B. Liu, D.-A. Huang, L. F. Fei-Fei, and J. C. Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems*, pages 517–526, 2018.

[26] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[27] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[28] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[29] H. Kim and A. Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[30] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[32] A. Kosiorek, H. Kim, Y. W. Teh, and I. Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pages 8606–8616, 2018.

[33] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015.

[34] A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

[35] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, 128(10):2402–2417, 2020.

[36] Y. Li and S. Mandt. Disentangled sequential autoencoder. In *35th International Conference on Machine Learning, ICML 2018*, volume 13, pages 8992–9001, 2018.

[37] Z. Li, J. V. Murkute, P. K. Gyawali, and L. Wang. Progressive learning and disentanglement of hierarchical representations. *arXiv*, (2015):1–14, 2020.

[38] Y. Liu, F. Wei, J. Shao, L. Sheng, J. Yan, and X. Wang. Exploring disentangled feature representation beyond face identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2080–2089, 2018.

[39] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[40] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, and O. Bachem. On the fairness of disentangled representations. In *Advances in Neural Information Processing Systems*, pages 14611–14624, 2019.

[41] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124, 2019.

[42] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020.

[43] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in neural information processing systems*, pages 5040–5048, 2016.

[44] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[45] H. Mogultay, S. Kalkan, and F. T. Y. Vural. L-vae: Variational auto-encoder with learnable beta for disentangled representation. Under Review.

[46] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y. L. Yang. HoloGAN: Unsupervised learning of 3d representations from natural images. *arXiv*, 2019.

[47] W. Nie, T. Karras, A. Garg, S. Debnath, A. Patney, A. B. Patel, and A. Anandkumar. Semi-supervised stylegan for disentanglement learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7360–7369, 2020.

[48] Y. Nitzan, A. Bermano, Y. Li, and D. Cohen-Or. Face identity disentanglement via latent space mapping. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[49] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE in-*

ternational conference on advanced video and signal based surveillance, pages 296–301. Ieee, 2009.

[50] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

[51] A. G. Reddy, B. G. L, and V. N. Balasubramanian. On causally disentangled representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, February 2022.

[52] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[53] K. Ridgeway and M. C. Mozer. Learning deep disentangled embeddings with the f-statistic loss. *Advances in neural information processing systems*, 31, 2018.

[54] O. Rybkin, K. Daniilidis, and S. Levine. Simple and effective vae training with calibrated decoders. In *International Conference on Machine Learning*, pages 9179–9189. PMLR, 2021.

[55] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[56] H. Shao, Z. Xiao, S. Yao, D. Sun, A. Zhang, S. Liu, T. Wang, J. Li, and T. Abdelzaher. Controlvae: Tuning, analytical properties, and performance analysis. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):9285–9297, 2021.

[57] H. Shao, Y. Yang, H. Lin, L. Lin, Y. Chen, Q. Yang, and H. Zhao. Rethinking controllable variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19250–19259, 2022.

[58] H. Shao, S. Yao, D. Sun, A. Zhang, S. Liu, D. Liu, J. Wang, and T. Abdelzaher. Controlvae: Controllable variational autoencoder. In *International Conference on Machine Learning*, pages 8655–8664. PMLR, 2020.

[59] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5541–5550, 2017.

[60] R. Suter, D. Miladinovic, B. Schölkopf, and S. Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pages 6056–6065. PMLR, 2019.

[61] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017.

[62] X. Wang, H. Chen, Y. Zhou, J. Ma, and W. Zhu. Disentangled representation learning for recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):408–424, 2022.

[63] L. Weng. From Autoencoder to Beta-VAE, 2018. `https://www.citedrive.com/https://lilianweng.github.io/posts/2018-08-12vae/`.

[64] W. F. Whitney, M. Chang, T. Kulkarni, and J. B. Tenenbaum. Understanding visual concepts with continuation learning. *arXiv preprint arXiv:1602.06822*, 2016.

[65] M. Yang, F. Liu, Z. Chen, X. Shen, J. Hao, and J. Wang. Causalvae: disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9593–9602, 2021.

[66] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[67] Y. Zhu, M. R. Min, A. Kadav, and H. P. Graf. 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, 2020.

[68] Y. Zhu, M. R. Min, A. Kadav, and H. P. Graf. S3VAE: Self-Supervised Sequential VAE for Representation Disentanglement and Data Generation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6537–6546, 2020.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Mogultay, Hazal
Nationality: Turkish (TC)

EDUCATION

2017 –	Ph.D.	Department of Computer Engineering, METU
2014 – 2017	M.Sc.	Department of Computer Engineering, METU
2009 – 2014	B.Sc.	Department of Computer Engineering, METU

PROFESSIONAL EXPERIENCE

2014 – Present	Research and Teaching Assistant
	Department of Computer Engineering , METU
2015 – 2017	Teaching Assistant Representative
	Department of Computer Engineering , METU
2013	Summer Practice
	Image Processing and Pattern Recognition Lab. , METU
2012	Summer Practice
	Information Processing Department, National Post Office

PUBLICATIONS

- **Mogultay, H.**, & Kalkan, S, & Vural, F. T. Y. (2018). An Analysis on Disentanglement in Machine Learning. In 30th Signal Processing and Communications

Applications Conference (SIU), pp. 1-4, IEEE.

- **Mogultay, H.**, & Vural, F. T. Y. (2018). BrainParcel: A Brain Parcellation Algorithm for Cognitive State Classification. In Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities pp. 32-42, Springer, Cham.

- **Mogultay, H.**, & Vural, F. T. Y. (2017). Cognitive learner: An ensemble learning architecture for cognitive state classification. In 25th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, IEEE. (**IEEE SPS Turkey, Best Student Paper Award**)

- **Mogultay, H.**, Alkan, S., & Vural, F. T. Y. (2015). Classification of fMRI data by using clustering. In 23th Signal Processing and Communications Applications Conference (SIU), pp. 2381-2383, IEEE.

AWARDS

- IEEE Signal Processing Society Turkey, Best Student Paper Award for the following paper:
  Mogultay, H., & Vural, F. T. Y. (2017). Cognitive learner: An ensemble learning architecture for cognitive state classification. In 25th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, IEEE.