5-2023

kFactorVAE: Self-Supervised Regularization for Better A.I. Disentanglement

Joseph S. Lee
William & Mary

kFactorVAE: Self-Supervised Regularization for Better A.I. Disentanglement

A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science in Computer Science from
William & Mary

by

Joseph S. Lee

Accepted for Honors

Huajie Shao, Director

Qun Li

Daniel Runfola

Williamsburg, VA
May 9th, 2023

kFactorVAE: Self-Supervised Regularization for Better A.I. Disentanglement

By
Joseph S. Lee

Department of Computer Science
William & Mary

Advisor: Professor Huajie Shao

Abstract

Obtaining disentangled representations is a goal sought after to make A.I. models more interpretable. Studies have proven the impossibility of obtaining these kinds of representations with just unsupervised learning, or in other words, without strong inductive biases. One strong inductive bias is a regularization term that encourages the invariance of factors of variations across an image and a carefully selected augmentation. In this thesis, we build upon the existing Variational Autoencoder (VAE)-based disentanglement literature by utilizing the aforementioned inductive bias. We evaluate our method on the dSprites dataset, a well-known benchmark, and demonstrate its ability to achieve comparable or higher disentanglement in significantly fewer training steps against our model's unsupervised counterparts.

Acknowledgments

I express my sincerest gratitude towards my advisor, Professor Huajie Shao, along with two other professors who have also fueled my passion for A.I.: Professor Qun Li and Professor Dan Runfola. They all have introduced to me interesting concepts, areas, and/or applications of machine learning that I will remember for the rest of my life. They gave me wonderful projects and opportunities, all of which have significant relevance to the world's most pressing problems. Of course, I also thank all of them for being on my honors thesis committee.

I also sincerely thank all my friends who have provided moral support and/or prayers for my honors thesis, including Jacob Somer, Willough Sloan, Jakob Ma, and Chris Chun. I thank them for all the wonderful conversations and their willigness to continue befriending and supporting me throughout my time at William & Mary.

My mother and my father have also deeply supported me with compassion, care, support, and most importantly: unconditional love.

In addition, I express deep gratitude towards system administrator Joseph Hause for his incredible technical support on the McGlothlin-Street Linux lab machines, where I set up and ran my experiments.

Finally, I thank God. It was truly a miracle to have accomplished this honors thesis, and I acknowledge and deeply respect all the ways He has been working in my life.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Deep neural networks hold immense potential to revolutionize various industries due to their ability to achieve human-level performance or higher on image-based and text-based tasks. However, their deployment in real-world settings has been far from reliable. These networks may base their decisions on unexpected variables, such as using the background color to predict an animal, rather than the animal's features themselves [2]. They may distinguish shapes in an image based on their location rather than the edges of each shape [2]. Therefore, the representations that these models construct from input data are entangled with simpler features, causing incorrect predictions. We refer to this phenomenon as "shortcut learning" [2]. While shortcut learning can be caused by the nature of the dataset itself (e.g., cows will almost always appear in a grassland), having large volumes of diverse data will still contain numerous chances for shortcut learning [2].

In light of this, disentangled representation learning (DRL) is an active area of research [3]. DRL aims to separate the explanatory, independent factors in the data, much like how we identify an animal by its body parts and overall shape, regardless of its surroundings [3, 4]. By achieving this separation, DRL significantly improves the interpretability of a model's decision-making process. One prominent example of a DRL model is the variational autoencoder (VAE) [1], which is an encoder-decoder pair with an added probabilistic element to capture variation in the data [6]. The encoder converts a high-dimensional input data point into a lower-dimensional Gaussian mean and variance representation vector, while the decoder takes a sample from that distribution and attempts to reconstruct the original input [6]. Through regularization towards a unit multivariate Gaussian distribution, a VAE enables controllable image generation by modifying one element of its representation vector from a fixed range (e.g., from -3 to 3 inclusive) [7]. Ideally, each element in this vector would correspond to just one semantic attribute in the decoder's output. For example, traversing [2] one element might change the color of a shape, without affecting any other factor such as position, background, etc. DRL has a wide variety of applications beyond image generation, such as image classification, segmentation, recommendation systems, video prediction, graphs, and natural language processing [3]. All these other applications use methods that build on top of the literature in disentangled image generation such as, not surprisingly, the VAE [3]. For this reason, my thesis work hones in on the image generation application.

[1] It is rather variations of the VAE that are more suited at disentanglement as opposed to the original VAE model [5]. These variations will be discussed in the literature review.

[2] A traversal means modifying one element of a representation vector across a discrete range (e.g., from -3 to 3 in increments of 2/3) while holding all other elements constant.

Despite the promising results demonstrated by VAEs, an important study has mathematically proven that unsupervised disentanglement with VAEs is impossible for any dataset characterized by an arbitrary underlying generative, factorized distribution [8]. The authors emphasize future work should focus on implementing inductive biases, both on the learning approach side and on the dataset side. This is an intuitive result given the *No Free Lunch Theorem* as well [9]. Contrastive learning, an inductive bias where representations for similar input are encouraged to be close together element-wise and representations for different input and encouraged to be farther apart element-wise through a regularization term measuring distance, has shown promise for disentanglement in at least a few research publications [10, 11, 12, 13]. These studies utilize contrastive learning coupled with other inductive biases in mind, such as model architecture and the objective function. Another study [14] has mathematically proved & empirically demonstrated the broad ability of a contrastive learning loss function to encourage models to find the underlying factors of variation in the data generation process. Some of these studies also use data augmentation [11, 12], another inductive bias. Data augmentation, in general, has shown to be useful for computer vision tasks [15, 16, 17], both with affine transformation (e.g., horizontal flipping and rotation) and with A.I.-based transformations (e.g., neural style transfer).

1.2 Main Contributions & Outline

This thesis is organized as follows:

1. **Background & Related Works**: I start by introducing the fundamental intuition behind VAEs and then survey the trajectory of disentangled representation learning as a whole, highlighting the variety of applications and models mentioned in a survey by Wang et al. (2022) [3]. I discuss also the advantages and limitations of the methods mentioned.

2. **Proposal**: This section elaborates on my idea and connects it back to the related works discussed in the previous section. To the best of my knowledge, I propose a unique method for disentangled representation learning that utilizes contrastive learning in the form of a mean-squared-error regularization term.

3. **Experiments with Discussion**: This section details the experimental design, comparisons with other related methods, and reports the results. I also discuss the implications of these results.

4. **Conclusion**: Finally, I conclude with a summary of the main findings of this thesis and suggest areas for future exploration and research directions.

2 Background

2.1 The (Non-Variational) AutoEncoder

To understand the intuition behind a VAE, we start off with its predecessor: the autoencoder. It is an encoder-decoder pair, where the ideal encoder compresses high-dimensional data (e.g., a $1280 \times 720 \times 3$ image) that captures an image's most essential information/features, and the decoder reconstructs the original input. An encoder's output is referred to as the latent vector and the decoder's output is ideally a reconstruction of the input. The distribution of possible latent vector values is referred to as the latent space.

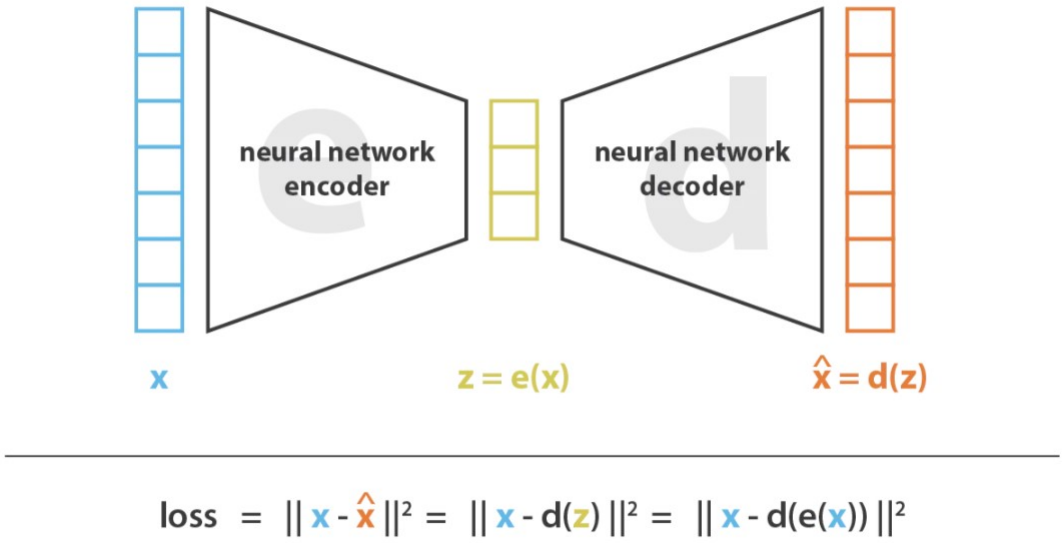


Figure 2.1: Illustration of an autoencoder with its loss function [1]

In the above figure 2.1, x represents an input, z represents the latent vector, and $\hat{x}$ represents the attempted reconstruction from the decoder.

In general, there are multiple choices for the encoder/decoder. Examples include matrices (e.g., principal component analysis) or normalizing flows [18]. In the realm of images, a standard choice for both are convolutional neural networks (CNN), given their expressive power.

An autoencoder inherently fails at disentanglement because it is unclear how the latent space would be organized [1]. There is no guarantee that similar inputs would have similar representations in the latent space (i.e., discontinuity) and thereby that the traversal of a single latent dimension/value (one element of the latent vector) would result in meaningful samples, yet alone be able to define a proper range for the traversal. A visual example is shown in 2.2.

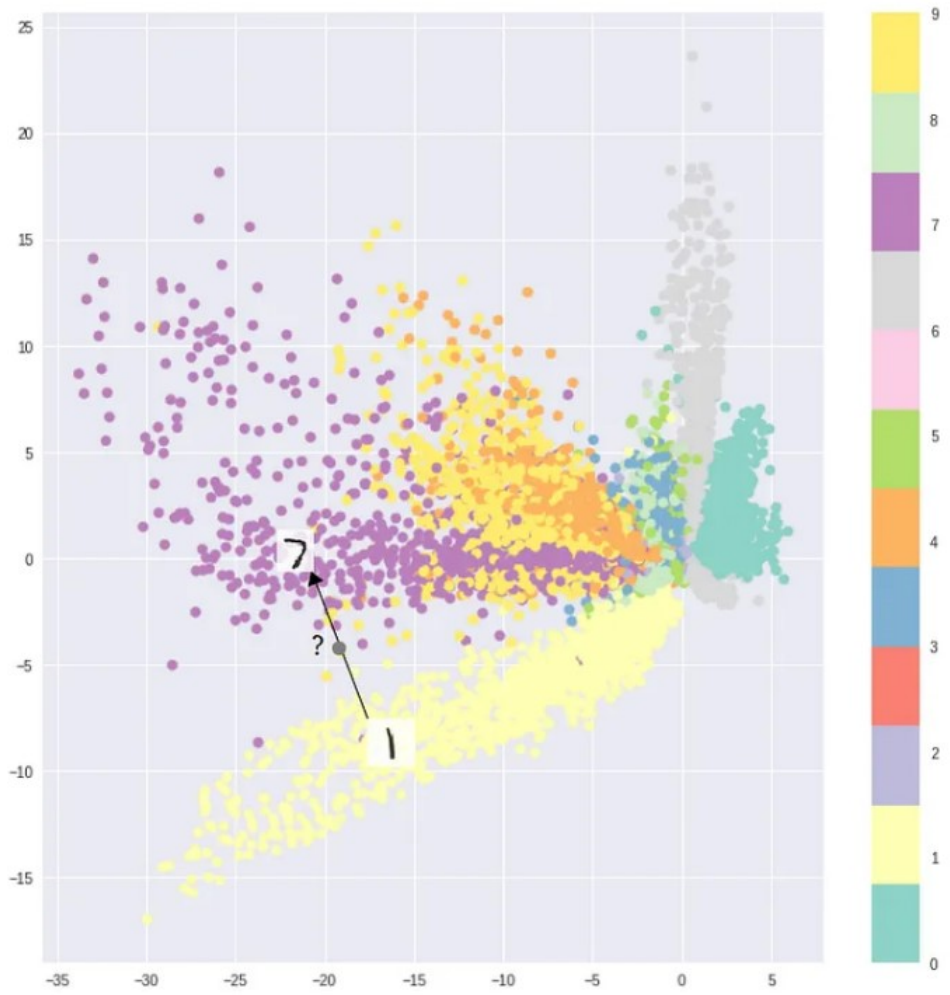


Figure 2.2: A visualization of 2D representation vectors/encodings from an autoencoder trained on the MNIST dataset, purely optimizing for reconstruction error. It is not clear what the reconstructions will look like with a traversal in the latent dimension corresponding to the y-axis upwards or x-axis leftward in the region between the yellow and purple clusters [19].

2.2 Bringing in Variation

To address the discontinuity issue, instead of encoding each data point as merely a low-dimensional vector, we can instead encode each data point as a multivariate normal distribution, parameterized by a low-dimensional mean vector with a low-dimensional standard deviation vector [3]. For example, the mean vector and standard deviation vector could be: [4]

[3] In practice, it could also represent variance or the logarithm of the variance for better optimization.
[4] This example is borrowed from [19].

$$\vec{\mu_s} = \begin{pmatrix} 0.1 \\ 1.2 \\ 0.2 \\ 0.8 \\ \vdots \end{pmatrix} \quad \vec{\sigma_s} = \begin{pmatrix} 0.2 \\ 0.5 \\ 0.8 \\ 1.3 \\ \vdots \end{pmatrix}$$

We would take a sample based on $\vec{\mu_s}$ and $\vec{\sigma_s}$ and pass that sample into the decoder. Our sample would take on the following form:

$$\vec{S} = \begin{pmatrix} S_1 \sim \mathcal{N}(0.1, 0.2^2) \\ S_2 \sim \mathcal{N}(1.2, 0.5^2) \\ S_3 \sim \mathcal{N}(0.2, 0.8^2) \\ S_4 \sim \mathcal{N}(0.8, 1.3^2) \\ \vdots \end{pmatrix}$$

where $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2. $\vec{S} \sim \mathcal{N}(\vec{\mu_s}, \vec{\sigma_s}^2)$ [5], where $\vec{\sigma_s}^2$ is the element-wise square of $\vec{\sigma_s}$. 2.3 provides a visual.

Now, it is possible for a single image to have different encodings. These encodings will not be far apart from one another because of the deterministic operations of the encoder and decoder along with the need to optimize the reconstruction error. Therefore, encodings that are similar in value, element-wise, will have similar decoder outputs. However, it is still not necessarily guaranteed that normal distributions of different data points are close together, as seen below in 2.4.

We will need a way to bring these normal distributions closer together. To accomplish this, we can add a Kullback Liebler (KL)-divergence regularization term to the reconstruction objective. So, our loss becomes:

$$\mathcal{L} = \underbrace{||\vec{x} - \hat{\vec{x}}||^2}_{\text{reconstruction error}} + \underbrace{D_{KL}(Q_x||P)}_{\text{Kullback-Liebler divergence}}$$

where Q_x is our estimated normal distribution probability density function from an input x and P is our choice of a prior distribution density function. A common choice for P is the probability density function for $\mathcal{N}(\vec{0}, \vec{1})$. This Kullback-Liebler divergence term will ensure that each distribution is close to the normal distribution of $\mathcal{N}(\vec{0}, \vec{1})$. Referring back to our MNIST example, when we apply this KL-divergence along with reconstruction, we get a result that would resemble 2.5.

[5] This is an abuse of notation, as a multivariate normal distribution has the form $\mathcal{N}(\vec{\mu}, \Sigma)$, where Σ represents the covariance matrix, and it is needed to account for covariances between two variables. For disentanglement, we want to model the covariances between any two different variables as 0. Therefore, Σ becomes a diagonal matrix and we represent this diagonal matrix as $\vec{\sigma}$.

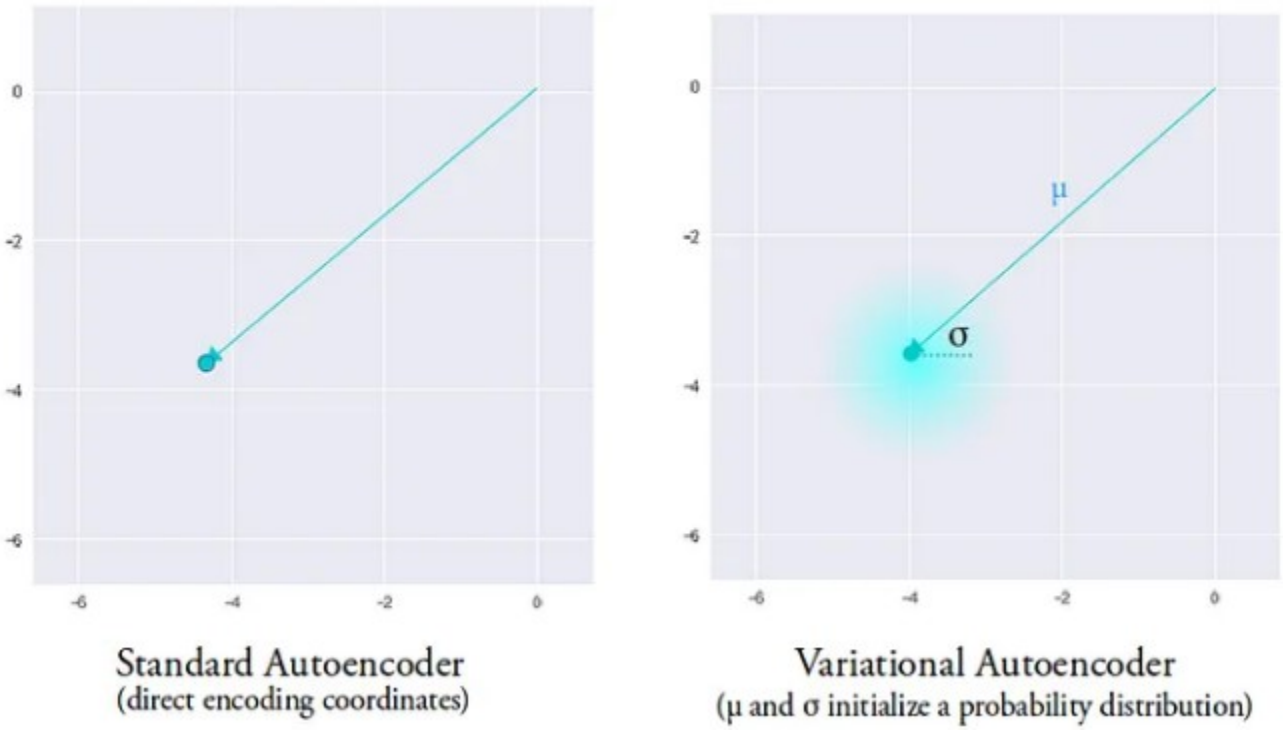


Figure 2.3: A visualization illustrating the difference between the output of a (non-variational) encoder and the output of a variational encoder for a single data point input [19].

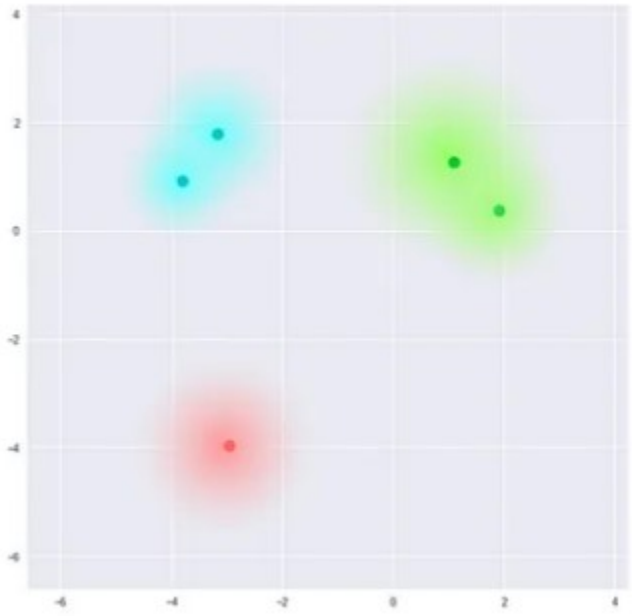


Figure 2.4: A visualization illustrating potential dispersion of normal distributions with just a reconstruction objective [19]. There is a lack of regularization of the locations of these distributions. In other words, the different dots would ideally be positioned close together as one cluster.

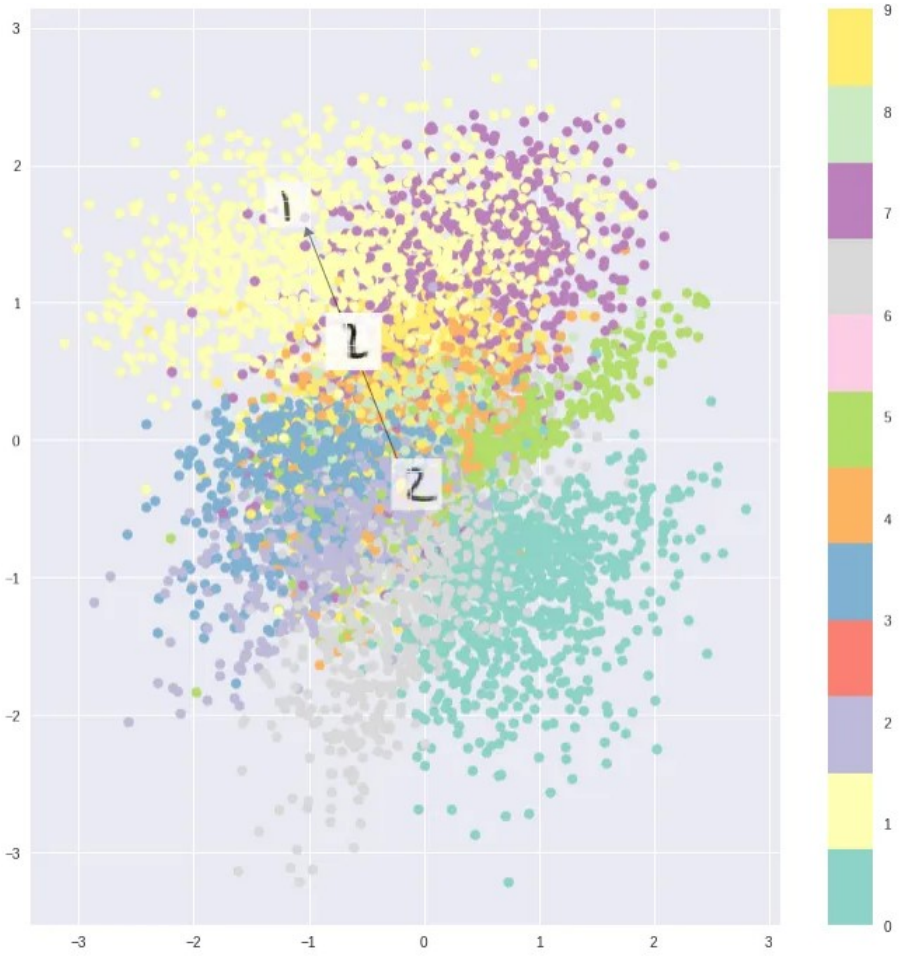


Figure 2.5: A visualization of 2D representation vectors/encodings (i.e., the latent space) from a variational autoencoder trained on the MNIST dataset, optimizing for both reconstruction error and the KL divergence between the estimated normal distribution and the prior normal distribution $\mathcal{N}(\vec{0}, \vec{1})$ [19].

Now we have a continuous, smooth latent space where similar representation vectors will correspond with similar decoder outputs and this space is neatly constrained in a bounding box, more or less, where one can sample each dimension along a fixed range, such as from -2 to 2, and ensure that majority, if not all, of the latent space, is captured. All of this sets up a good precedent for disentanglement.

2.3 The Variational Autoencoder (VAE), Formally

With the above section describing the essence of a variational autoencoder's key components, the formal derivation of its concepts will be noted here. Broadly, VAEs are an attempt to address the problem of generative modeling [6] with the presence of continuous latent variables, intractable posterior distributions (i.e. our estimated normal distributions for each input data point), and large datasets [5]. The idea of optimizing for both reconstruction and the KL divergence between the estimated distribution for a datapoint and the prior distribution comes from a probability/statistics concept known as the variational lower bound [5] or Evidence (Variational) Lower Bound (ELBO) [3].

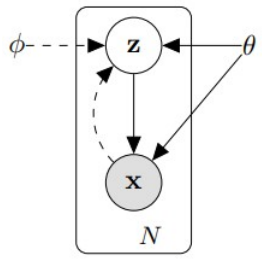


Figure 2.6: Illustration of the relationships between the input data points $\mathbf{x}$ and the hidden/latent/underlying variables $\mathbf{z}$ (where the boldness indicates a vector) with respect to their variational parameters ϕ and the generative parameters θ [5]. $\mathbf{z}$ is the output of an encoder and would represent the succinct attributes that fully capture all the (disentangled) factors of variation of $\mathbf{x}$. This diagram is generalizable to any kind of generative model. For our purposes, we can think of ϕ as the neural net parameters of the encoder and θ as the neural net parameters of the decoder.

To motivate the ELBO, we start off with what all generative models seek to accomplish: to discover or to accurately approximate the marginal likelihood $p_\theta(\mathbf{x})$. In general, this likelihood is intractable as it is equal to the $\int p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})d\mathbf{z}$ [5]. We are interested in estimating the true posterior density $p_\theta(\mathbf{z}|\mathbf{x})$, also known as the distribution of latent variables given an input $\mathbf{x}$. Since $p_\theta(\mathbf{z}|\mathbf{x}) = \dfrac{p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{x})}$, this term is also intractable [5]. Therefore, the expectation-maximization (EM) algorithm cannot be used in this problem [5].

To work around this, we make the assumption that $\mathbf{x}$ is an independent and identically distributed (i.i.d) data point, and all data points are i.i.d. We could then write:

[6] In a probability/statistics context, generative modeling concerns with figuring out the probability density function of a dataset $p(\vec{x}^{(1)}, \ldots, \vec{x}^{(N)})$, usually parameterized. Generative modeling is not limited to just visual image generation.

$$\log p_\theta(\mathbf{x}) = D_{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z} \mid \mathbf{x})\right) + \underbrace{\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z})}_{\text{ELBO}}$$

where

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - D_{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\right),$$

$q_\phi(\mathbf{z} \mid \mathbf{x})$ is the estimated distribution of the latent variables given an input $\mathbf{x}$, and

$p_\theta(\mathbf{z}|\mathbf{x})$ is the true distribution in which q is estimating [5, 3].

$p_\theta(\mathbf{z})$ is the prior distribution of choice. Although parameterized here, the literature usually chooses a fixed prior.

It can be shown that if we assume that the probability distribution of $q_\phi(\mathbf{z}|\mathbf{x})$ is a normal distribution [7], then the expectation term in $\mathcal{L}$ is equivalent to the reconstruction objective as shown earlier. Furthermore, if the choice of the prior distribution for $p_\theta(\mathbf{z})$ is the normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$, then the D_{KL} term becomes:

$$\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\left((\sigma_j)^2\right) - (\mu_j)^2 - (\sigma_j)^2\right)$$

where J is the number of elements or the dimensionality of, $\mathbf{z}$ [5].

What these equations show is that the simplifying assumption of a normal distribution allows tractability. Kingma & Welling (2013, revised 2022) [5] then show how θ and ϕ can be optimized with minibatch stochastic gradient descent, the goal being to maximize $\mathcal{L}$ through minimizing $-\mathcal{L}$. In addition, Kingma et al. [5] go over the reparametrization trick, which helps ensure differentiability, as the sampling operation of a distribution is not inherently differentiable. This mathematical foundation sets the precedent for later literature, which focuses on adding regularization terms to explicitly encourage disentanglement of $\mathbf{z}$.

3 Related Works

We will go over a survey of the literature on the disentanglement of image generation, the vast majority of which are mentioned in [3]. The ideas, advantages, and limitations are discussed. All the below literature assumes a normal/Gaussian distribution as the prior distribution and the choice of distribution of the encoder.

[7] Kingma et al. note that while this is a simplifying assumption, it is not a limitation of their method. [5].

3.1 Unsupervised Methods

VAE-based: Higgins et al. (2016) [7] are the first to bring disentanglement into the table for VAEs. Their overarching idea was to modify the VAE model proposed in [5] to discover interpretable, factorized latent representations in an unsupervised fashion. Their modification, β-VAE, adds the hyperparameter β to the KL divergence term between the estimated distribution and the prior distribution:

$$\mathcal{L}(\theta, \phi, \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - \beta D_{KL} \left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\right)$$

like so [7]. The higher β becomes, the more push there is towards a more efficient and constrained latent representation, which is shown to be disentangled if the underlying factors of variation are independent [7]. β-VAE outperforms other state-of-the-art disentanglement methods, such as the VAE, InfoGAN [20], and DC-IGN [21] on datasets containing RGB-colored celebrity faces [22], chairs [23], and gray, 3D-model-based faces [24] both qualitatively (by inspecting latent traversals) and qualitatively (by their own proposed, classifier-based, disentanglement metric). Choices of β included $\beta = 5$ on the chairs dataset and $\beta = 20$ on the gray, 3D faces dataset [7]. Another contribution of Higgins et al. is that they create a dataset with white shapes on a black canvas as an additional benchmark for disentanglement [7]. This is referred to as the dSprites dataset [25]. The limitation of β-VAE, however, is that the higher β becomes, the higher the reconstruction error becomes. There is therefore a tradeoff between reconstruction fidelity and the quality of disentangled latent representations.

Burgess et al. (2018) [26] sought to address this tradeoff with insights from their rate-distortion theory perspective and their analysis of β-VAE on a dataset of Gaussian blobs in various locations on a black canvas. The idea is to progressively increase latent capacity during training, and this has experimentally shown to lead to more robust learning of disentangled representation in β-VAE without the tradeoffs in reconstruction accuracy. They modify the β-VAE objective as follows:

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

where C is gradually increased from 0 to some manually set upper bound [26] and γ is a hyperparameter to control how much to penalize the deviation between C and the KL divergence between the estimated distribution for input and the prior distribution. In practice, the authors used $\gamma = 1000$, $0 \leq C \leq 25$ with linear increments over $100,000$ training iterations for the dSprites dataset [25]. For the CelebA dataset [22], the range $0 \leq C \leq 50$ was used instead.

Kumar et al. (2018) [27] take a different stance on the limitations raised in [7]. They raise the need to

compute a quantity known as the *inferred prior* or *expected variational posterior*:

$$q_\phi(\mathbf{z}) = \int q_\phi(\mathbf{z}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

and modify the overall objective as follows:

$$\max_{\theta,\phi} \mathbb{E}_{\mathbf{x}} \left[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{z}) \right] - \mathrm{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z}) \right) \right] - \lambda D\left(q_\phi(\mathbf{z}) \| p(\mathbf{z}) \right)$$

[27].

Their model is known as Disentangled Inferred Prior-VAE (DIP-VAE) [27]. Considering multiple avenues for approximating $KL(q_\phi(\mathbf{z})||p(\mathbf{z}))$, since it is intractable, they chose to match the covariance of the two distributions, which will decorrelate the dimensions of $\mathbf{z} \sim q_\phi(\mathbf{z})$ [27]. Looking at covariance matrices of the estimated normal distribution means, they have two hyperparameters λ_{od} and λ_d to control the relative importance of diagonal entries and off-diagonal entries. They have two objectives which each have a different form of regularization as follows:

$$\begin{aligned}
\max_{\theta,\phi} \mathrm{ELBO}(\theta,\phi) &- \lambda_{od} \sum_{i\neq j} \left[\mathrm{Cov}_{p(\mathbf{x})} \left[\boldsymbol{\mu}_\phi(\mathbf{x}) \right] \right]_{ij}^2 \\
&- \lambda_d \sum_i \left(\left[\mathrm{Cov}_{p(\mathbf{x})} \left[\boldsymbol{\mu}_\phi(\mathbf{x}) \right] \right]_{ii} - 1 \right)^2, \\
\max_{\theta,\phi} \mathrm{ELBO}(\theta,\phi) &- \lambda_{od} \sum_{i\neq j} \left[\mathrm{Cov}_{q_\phi(\mathbf{z})} [\mathbf{z}] \right]_{ij}^2 \\
&- \lambda_d \sum_i \left(\left[\mathrm{Cov}_{q_\phi(\mathbf{z})} [\mathbf{z}] \right]_{ii} - 1 \right)^2
\end{aligned}$$

This objective does not conflict between the data log-likelihood (i.e. reconstruction error) and the disentanglement of the inferred latent variables [27]. The authors also propose the SAP score, which overcomes the limitations of the metric proposed in [7]. In most attributes on the CelebA dataset, DIP-VAE outperforms both VAE and β-VAE [27].

Kim et al. (2018) [28] decomposes the KL divergence term from β-VAE, and highlights its dependency on mutual information:

$$\mathbb{E}_{p_{data}(\mathbf{x})}[D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] = I(\mathbf{x};\mathbf{z}) + D_{KL}(q(\mathbf{z})||p(\mathbf{z}))$$

A higher β could mean penalizing mutual information unnecessarily or undesirably for disentangling [28]. They link too much penalization of mutual information [8] as the reason for higher reconstruction errors

[8] Mutual information describes how much one variable reduces the uncertainty of another variable. $I(\mathbf{x};\mathbf{z})$ describes how much the latent variables $\mathbf{z}$ reduce the uncertainty of $\mathbf{x}$. The higher $I(\mathbf{x};\mathbf{z})$ is, the more $\mathbf{z}$ informs about $\mathbf{x}$..

in β-VAE [28]. Therefore, they motivate a regularization term that directly penalizes total correlation, which is a popular measure of dependence among random variables [28]. Less total correlation means more independence, which means less entanglement as each factor of variations stands alone.

The objective is:

$$\frac{1}{N}\sum_{i=1}^{N}\left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x}^{(\mathbf{i})})}\left[\log p\left(\mathbf{x}^{(\mathbf{i})} \mid \mathbf{z}\right)\right] - D_{KL}\left(q\left(\mathbf{z} \mid \mathbf{x}^{(\mathbf{i})}\right) \| p(\mathbf{z})\right)\right] - \gamma D_{KL}(q(\mathbf{z})\|\overline{q}(\mathbf{z})),$$

where

$$\underbrace{D_{KL}(q(\mathbf{z})\|\overline{q}(\mathbf{z})) = \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{q(\mathbf{z})}{\overline{q}(\mathbf{z})}\right] \approx \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{D(\mathbf{z})}{1-D(\mathbf{z})}\right]}_{\text{estimated total correlation (TC)}}$$

and N is the batch size.

where D is a multilayer-perceptron-based discriminator network function that outputs the estimated probability that $D(\mathbf{z})$ is a sample from $q(\mathbf{z})$ rather than from $\overline{q}(\mathbf{z})$ [28]. $\overline{q}(\mathbf{z})$ represents the factorial distribution of latent variables $\Pi_{j=1}^{d} q(z_j)$, where d is the number of latent variables chosen/the dimensionality of the latent space. The factorial distribution is such that changes in latent variables are independent of each other, which is the goal of disentangled representation learning. This factorial distribution is approximated using a permutation method across a minibatch of latent space samples [28]. The VAE and the discriminator are trained jointly, using the above objective and approximation of the total correlation using the discriminator [28]. Altogether, this architecture/framework/method is known as FactorVAE [28].

On both the dSprites dataset [25] and their own 3D shapes dataset (where the factors of variation are the wall color, floor color, shape color, scale, shape, and orientation) [29], they achieve better disentanglement scores than β-VAE, using their own defined metric [28], which yet again is another one to address the limitations of using a linear classifier for the metric in the β-VAE paper [7]. As γ increases, the reconstruction error stays stable according to the experimental graphs, unlike β-VAE, and the disentanglement metric scores (both on the one proposed by β-VAE and their own) are higher [28] This stable reconstruction error is also relatively low compared to the β-VAE ones, matching the reconstruction error associated with lower choices of β, while the losses associated with higher levels of β go up, as discussed before [28].

There are limitations with the FactorVAE paper's total correlation term and its disentanglement metric [28] Their total correlation term is necessary but not sufficient for disentanglement because there are unideal cases when the total correlation could be 0, such as all but one latent dimension collapsing to the prior distribution [28]. Their disentanglement metrics require the generation of samples while fixing one factor, which is not always possible depending on the training dataset [28].

A related paper, by Chen et al. (2019) [30], is able to address those two limitations. Their total correlation

approximation is their most important term for disentanglement and their disentanglement metric does not depend on any generation of samples. After similarly decomposing the KL divergence term in β-VAE:

$$\mathbb{E}_{p(n)}[\mathrm{D_{KL}}(q(\mathbf{z} \mid n)\|p(\mathbf{z}))] = \underbrace{\mathrm{D_{KL}}(q(\mathbf{z}, n)\|q(\mathbf{z})p(n))}_{\text{(i) Mutual Information } I_q(\mathbf{z}; n)} + \underbrace{\mathrm{D_{KL}}\left(q(\mathbf{z})\|\prod_j q\left(z_j\right)\right)}_{\text{(ii) Total Correlation}} + \underbrace{\sum_j \mathrm{D_{KL}}\left(q\left(z_j\right)\|p\left(z_j\right)\right)}_{\text{(iii) Dimension-wise KL}}$$

[9]

like so, they estimate all the components of their objective term,

$$\mathcal{L}_{\beta-\mathrm{TC}} := \mathbb{E}_{q(\mathbf{z}|n)p(n)}[\log p(n \mid \mathbf{z})] - \alpha I_q(\mathbf{z}; n) - \beta\, \mathrm{D_{KL}}\left(q(\mathbf{z})\|\prod_j q\left(z_j\right)\right) - \gamma \sum_j \mathrm{D_{KL}}\left(q\left(z_j\right)\|p\left(z_j\right)\right)$$

,

using minibatch weighted sampling and stochastic estimation [30]. Their method is referred to as β-TCVAE, where "TC" stands for "Total Correlation." This estimation has the advantages of training stability and reduction in the number of training hyperparameters/inner optimization loops [30]. Even though they have various hyperparameters, they obtained the best disentanglement results by tuning β. In fact, setting $\alpha = 0$ did not make a significant difference in performance [30]. On both the dSprites dataset and the 3D faces dataset, the median score on their disentangled metric.

Their disentangled metric is classifier-free and finds, among all the latent variables and a ground truth factor of variation, the highest mutual information value and the second-highest and takes their differences. These differences are summed up across all the ground truth factors, get normalized, and then the final result is known as the mutual information gap (MIG) [30]. Its range lies between 0 and 1, where 1 is the best possible score. In essence, for every dataset point and for every ground-truth factor of variation g, it takes the top two latent factors L_1, L_2 most related to a ground-truth factor of variation (as measured through mutual information), and takes their gap:

$$I(L_1; g) - I(L_2; g)$$

where

$$I(L_1; g) \geq (L_2; g)$$

[9] where $q(\mathbf{z}|n) = q(\mathbf{z}|\mathbf{x}_n)$, $q(\mathbf{z}, n) = q(\mathbf{z}|n)p(n) = \frac{1}{N}q(\mathbf{z}|n)$, and $n \in \{1, \ldots, N\}$. N is the number of training data points [30].

.

These gaps are then averaged across the ground-truth factors of variation to obtain the final MIG. An entire pass through the dataset is needed to obtain the MIG for a trained model [30]. Despite needing a pass through the whole dataset though, their implementation can be computed within a reasonable amount of time, at least for the datasets they used, such as dSprites [30]. The advantages of this metric are that it detects axis alignment, is unbiased to all hyperparameter settings, is broadly applicable to any latent distribution provided the existence of efficient estimations, and can better capture subtle differences in models compared to existing metrics [30]. [10] Axis alignment is the quality that each latent variable contains information regarding just one ground truth factor, and not two or more, which would be entanglement [30]. Limitations of β-TCVAE include rare low outliers in performance [30] and a relatively higher reconstruction error compared to FactorVAE at the cost of better disentanglement.

Controllable VAE by Shao et al. (2020) [31] takes a dynamic approach to the regularization weight term of the β in β-VAE. Inspired by control theory, they design a new non-linear controller to automatically adjust β from feedback during model training and symbolically indicate this as $\beta(t)$. In total, the objective function is:

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

The work we have gone over so far mostly focused on decomposing the ELBO, weighting specific aspects of the KL divergence. A couple of works stand apart from this progression [32, 33] by focusing on regulating the types of latent variables found. [32]'s model is known as Relevant Factor-VAE (RF-VAE), which aims to filter between major and minor factors of variation present in a dataset. [33]'s model, JointVAE, aims to separate out factors of variation that are continuous in nature (e.g. width, height, and orientation) and discrete in nature (e.g. handwritten digits). but not every factor of variation should be described in this manner. Sometimes, factors of variation are discrete in nature instead, such as handwritten digits. These works though do still use a modified form of the original ELBO, even borrowing certain concepts from previous works such as total correlation [28, 30] and gradual increase in KL divergence during training with a channel capacity hyperparameter C [26].

JointVAE's limitations there is additional computation to predict not just the continuous latent variables, but also the discrete latent variables as well in the intermediate layer between the encoder and the decoder. In addition, the discrete latent variables are not parameterized ideally by a set of categorical distributions, since the ability to differentiate for backpropagation is needed. They resort to a differentiable relaxation of

[10] I elaborated on this metric more than the others since we utilize it in our experiments.

discrete random variables based on the Gumbel Max trick [33]. In addition, without the channel capacity parameters for the discrete and continuous variables, the model may ignore discrete latent variables [33], so an extensive hyperparameter search will be needed. With RF-VAE, though it may effectively separate filter our minor/irrelevant latent factors of variation, sometimes, observing those minor factors of variation may be important to observe, especially in the application of highlighting underrepresented factors of variation in datasets to highlight and reduce bias in datasets [34].

GAN-based: Generative Adversarial Networks (GANs) are generator-discriminator pairs, where the generator aims to take a random low-dimensional input vector (i.e., a random latent code) and generate a realistic-looking sample from the training dataset [35]. The discriminator aims to distinguish between an output from the generator and the corresponding training data point the generator was trying to imitate [35]. The generator and discriminator are trained jointly, where the discriminator helps the generator gradually create more and more realistic-looking samples, and eventually, the discriminator will give a 50% probability to both the generator's output and the corresponding training data point [35]. Therefore, the objective is a min-max game and is considered adversarial:

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

.

There is nothing inherent about the setup that ensures a disentangled latent space, as the focus is on purely transforming a simple prior distribution into the complicated, high-dimensional data distribution from the training dataset. Chen et al. (2016) [20] are the first to attempt to make disentanglement more explicit by adding an additional vector $\mathbf{c}$ to the latent space (that can be parameterized as a categorical or any continuous distribution). They also modify the architecture by adding a classifier, whose parameters are shared with the discriminator, to predict $\mathbf{c}$ to help maximize the mutual information between the GAN's generated samples and $\mathbf{c}$. While this has shown to be a fruitful approach for learning disentangled representations, it performs comparably worse to VAE-based models such as β-VAE, as mentioned earlier [7].

Larsen et al. (2016) make a hybrid between a VAE and a GAN by merging the decoder and generator as one [36]. In addition, they point out the flaws with using pixel-to-pixel reconstruction error and replace it with feature-wise errors with learned representations from the discriminator [36]. One flaw with the pixel-to-pixel reconstruction error is that a large error is produced when an object in an image is shifted in any direction by any amount, so this kind of error fails to recognize the concept of similarity despite translation [36]. Their replacement of feature-wise errors is able to overcome flaws like these, resulting in a model that outperforms VAEs in terms of reconstruction error. A limitation is that it relies on attribute vectors

instead of individual latent dimensions to represent semantic attributes, so there is more space required and the overall VAE and GAN combination is more computationally expensive than a VAE model with added regularization.

Zhu et al. (2018) [37] propose a method referred to as Visual Object Networks (VONs), which seeks to address the issue of generative models lacking an understanding of the underlying 3D world. They make an image processing pipeline, involving a shape network, differentiable projection, and texture network to model both 3D shapes and 2D images, taking into account shape, viewpoint, and texture [37]. They propose to end-to-end adversarial learning framework to jointly model 3D shapes and 2D images. The advantages are that this model outperforms other GAN models in terms of Frechet Inception Distances, and the majority of humans in a survey they conducted prefer generated images compared to other models, which indicates a superb reconstruction quality [37]. The limitations of this paper course are that this is a computationally expensive setup and is specialized for image generation as opposed to disentanglement, and it only vaguely provides disentanglement through separated networks to handle the factors of shape, viewpoint, and texture. In addition, there is a relatively complicated objective function setup with two adversarial losses for texture, 3 cycle-consistency losses, and a KL divergence loss [37].

Finally, an interesting paper by Wu et al. (2021) [38] does not propose a new architecture, but rather analyzes a pre-existing GAN model: StyleGAN2. The authors show that the space of channel-wise style parameters, which they refer to as StyleSpace, already exhibits disentanglement, and is more disentangled than other intermediate latent spaces explored by similar prior literature [38]. They propose a method to discover the large collection of style channels and map them to specific attributes by using a pre-trained classifier or a small number of example images [38]. They also show methods to manipulate images in a disentangled fashion [38]. The limitation of course is that because this looks at a pre-existing model, it is limited to that model's disentanglement, and we cannot be certain that every single factor of variation has been disentangled [38]. However, the authors mention developing inversion techniques (e.g., an autoencoder) to have high reconstruction accuracy and manipulability [38].

The challenge with unsupervised methods: There is an overarching limitation to all of the unsupervised methods mentioned above [8], and this limitation is the key point my thesis addresses. Locatello et al. (2019) [8] explain intuitively, from their impossibility result, that since there are many possible estimations of the marginal distribution $p(\mathbf{x})$ with entanglement properties and disentanglement properties, an unsupervised disentanglement method cannot properly distinguish between them. They recommend making explicit inductive biases on both the models themselves and the datasets so that the space of solutions can be cut down and that these solutions can match the true generative model [8]. The authors give three examples of

promising avenues: disentanglement learning with interactions, weak forms of supervision such as grouping information, and temporal structure [8]. In this thesis, we focus on solutions interrelated with supervision.

3.2 Supervised Methods

VAE-based: Kulkarni et al. (2015) [21] developed a VAE-like model and enforce specific latent dimensions (or, chosen neurons in the graphics code layer in their own terminology) to specifically represent active factors of variation by having each training mini-batch have a set of only one active factor of variation, while the rest remain constant. For example, a nodding face actively changes the factor of elevation, but the factors of shape and texture would hold constant [21]. They use an objective very similar to the original VAE one, but now focus on a particular latent variable z_i:

$$-log(P(\mathbf{x}|z_i)) + D_{KL}(Q(z_i|\mathbf{x})||P(z_i))$$

to make it so that that variable explains all the variance in a minibatch. Every minibatch contains images with only one ground truth factor of variation changing while holding all other factors constant [21]. They also perform clamping and modify the gradients in their training algorithm [21]. The advantage of this method is that we can map a specific latent dimension to a specific desired semantic, thereby enforcing a one-to-one mapping.

Weaker levels of supervision, without knowing the ground truth factors of variation, have shown to be successful for disentanglement too, as Bouchacourt et al. (2018) demonstrate [39]. It is generally inexpensive to partition data into groups, where each group shares a common (but unknown) value for a factor of variation [39]. One could group images by shape or by color, for example, while letting the rest of the factors of variations vary [39]. The authors also partition their latent representation in terms of style and content. In a data group, the content is invariant while the style changes [39].

Whenever labels are provided, even if they are imprecise and incomplete, they can be incorporated to reliably learn disentangled representations according to Locatello et al. (2020) [40]. They use a binary cross-entropy to match the factors to their targets, along with a γ_{sup} regularization weight [40]. Only 100 labeled examples and this cross-entropy loss are needed to outperform unsupervised models in both disentanglement and downstream tasks. Unsupervised training with supervised validation and semi-supervised training is robust to label noise and can handle coarse and partial annotations [40].

Another problem with unsupervised methods, raised by Träuble et al. (2021), is that any correlations in a dataset get learned and reflected in latent representations. They resolve these latent correlations with a small number of labels post-hoc or use weak supervision during training [41]. For example, a size factor

may address both foot length and body height. To address this spurious correlation, weak supervision can help [41]. Their weak supervision method is defined by [42], where they consider pairs of images where it is certain that only a small subset of the total factors of variation have changed their values, leaving the rest of the values invariant.

GAN-based: Disentangled Representation learning-GAN (DR-GAN) by Train et al. (2017) is a model proposed for the specific application of pose recognition. They seek to jointly learn face frontalization (where a frontal view of the face is generated, no matter the input pose) and learn a pose-invariant representation from the non-frontal face image, instead of separating the two as past literature did [43]. Labels are explicitly fed into components of DR-GAN: pose code for the decoder and pose estimation input for the discriminator, all of which allow for explicit disentanglement [43]. They achieved superior results over the state-of-the-art in quantitative and qualitative evaluation on both controlled and in-the-wild datasets.

DNA-GAN by Xiao et al. (2017) [44] structure the latent representations as inspired by the DNA double helix structure, where each piece of the encoding is an independent factor of variation [44]. Swapping pieces and annihilating a recessive piece allows for two different representations to be decoded into two kinds of images, differing by the corresponding attribute [44]. They couple this structure with multi-attribute labels along with a discriminator to tell apart the genetically modified reconstructions from their input counterparts for disentangling [44]. Their results show that their method can overcome challenges in unbalanced datasets and helps to disentangle multiple attributes in the latent space [44].

Causal-based: these kinds of models fundamentally change the generative model design with a directed acyclic graph (DAG) that describe relationships in which one factor causes another factor to change [45, 46]. In 2020, Shen et al. introduced "Disentangled gEnerative cAusal Representation" (DEAR), which uses supervised information and a structural causal model (SCM) as a prior for the bidirectional generative model [46]. This model is jointly generated with the generator and encoder with a GAN algorithm [46]. This setup is able to model real-life scenarios where data is not necessarily generated with independent factors of variation. For example, one's shadow is dependent on the source of illumination and one's position relative to it; it does not make sense to have a shadow without a source of illumination [46]. Yang et al. (2021) [45] uses a causal layer to transform externally independent factors to internally causal ones, and uses a DAG-constrained adjacency matrix [45].

The challenge with supervised methods: Even with success with just a relatively small amount of labels compared to the size of the entire training datasets, there are still fundamental challenges to data labeling, such as cost, time consumption, and human error. Moreover, even if we had the ideal labels that captured all the possible factors of variation we cared about, there may be more subtle factors of variation that could be important for our model to capture. Our understanding of the world and its variation is continuing to

grow every day, and our hope with disentangled representation learning is to subtle factors of variation we may have not noticed before.

3.3 Self-Supervised Methods:

This is the direct umbrella my thesis falls under. Self-supervised learning is a subarea of machine learning that deals with unlabeled data as with unsupervised learning but still has some supervised-like objective function to achieve. One example of this is contrastive learning, where similar images are encouraged to have similar representations, and if possible to obtain, different images are encouraged to have representation pulled far apart from each other [47]. This is accomplished through some kind of distance metric/regularization function [47].

A prominent paper in this area is Wang et al. (2021) [11], where self-supervised learning, augmentation, and an iterative partitioning algorithm are used to find entangled factors of variation, disentangle them with a gradient-based parameter updating, and then repeat the process until convergence. However, as the authors note themselves, their limitations are a tricky optimization process and that their algorithm can be time-consuming in practice [11]. At what stage to perform maximization, how many epochs to train the maximization step for, and how to decide when a step achieves convergence are all open questions[11]. With VAEs in contrast, for optimization, all that needs to be adjusted is the number of epochs or training steps.

4 Proposal

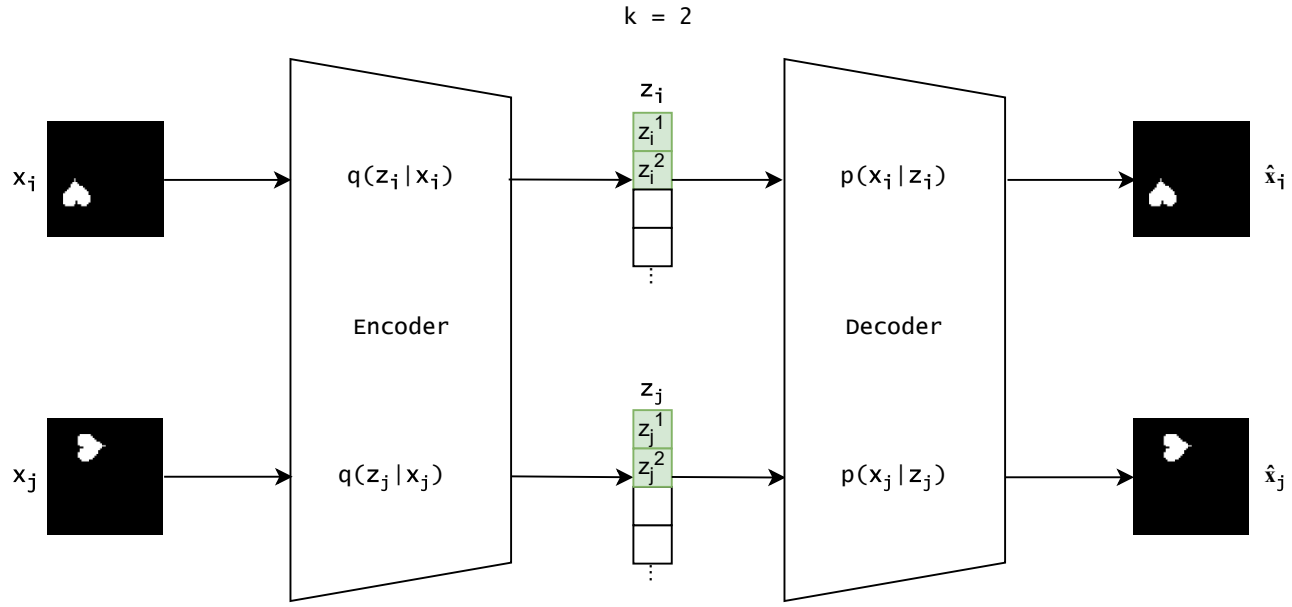


Figure 4.1: Proposed model architecture: the green squares highlight factors of variation to whose values are enforced to be similar through our regularization loss.

kFactorVAE takes advantage of data augmentations that preserve at least one factor of variation through well-known, common augmentation techniques, such as rotation, reflection, and noise injection. For example, in the dSprites example in 4.1, rotating an image of a heart keeps the shape factor constant and the size factor constant. k is a hyperparameter indicating the number of factors of variation whose values are encouraged, by a regularization term, to be brought closer together across a data point and its augmentation.

kFactorVAE adopts FactorVAE's encoder architecture, decoder architecture, and discriminator network noted in appendices A and B, along with its objective function. We chose FactorVAE because it achieves both a low reconstruction error and a relatively high disentanglement score–namely, the mutual information gap [30] as explained in the literature review–compared to other models [28].

Adding in the k-factor regularization term (also known as the k-factor similarity loss) yields our modified objective to maximize:

$$\frac{1}{2N}\sum_{i=1}^{2N}\underbrace{\left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x}^{(\mathbf{i})})}\left[\log p\left(\mathbf{x}^{(\mathbf{i})}\mid\mathbf{z}\right)\right]}_{\text{reconstruction accuracy}}\underbrace{-D_{KL}\left(q\left(\mathbf{z}\mid\mathbf{x}^{(\mathbf{i})}\right)\|p(\mathbf{z})\right)\Big]}_{\text{encoder distrib. dist. from }\mathcal{N}(0,I)}\underbrace{-\gamma D_{KL}(q(\mathbf{z})\|\overline{q}(\mathbf{z}))}_{\text{total correlation}}\underbrace{-\alpha\cdot\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{z_{2i-1}}^{1:k}-\mathbf{z_{2i}}^{1:k}\|_2^2}_{k\text{-factor similiarty loss}},$$

$$D_{KL}(q(\mathbf{z})\|\overline{q}(\mathbf{z}))=\mathbb{E}_{q(\mathbf{z})}\left[\log\frac{q(\mathbf{z})}{\overline{q}(\mathbf{z})}\right]\approx\mathbb{E}_{q(\mathbf{z})}\left[\log\frac{D(\mathbf{z})}{1-D(\mathbf{z})}\right]$$

where p is associated with, and is parametrized by the decoder. q is associated with, and is parametrized by the encoder. $\bar{q}$ is the estimated factorial distribution based on the permutation of a minibatch of latent space samples. N is the batch size. $2N$ indicates an additional N data points from applying an augmentation for each of the original N selected data points for the batch. α is the weight of the newly proposed regularization term. The $2i-1$ and $2i$ denotes an indexing scheme where a data point and its augmentation are placed adjacently. $1:k$ is another indexing scheme indicating the selection of the first k indices of a data point's latent space representation and the augmentation's latent space representation. $\|\mathbf{z_{2i-1}}^{1:k} - \mathbf{z_{2i}}^{1:k}\|_2^2$ is the square of the L_2 norm. $\mathbf{z}$ indicates a sample from the normal distribution outputted by the encoder. k is an integer ranging from 1 to the total number of dimensions of the latent space, inclusive. D is the same discriminator from FactorVAE [28]. $\bar{q}(\mathbf{z})$ is approximated in the same permutation method as FactorVAE [28], using another N randomly selected data points from the training dataset.

We selected the MIG as the benchmark metric given its powerful ability to effectively establish whether changes in the latent space reflect meaningful and expected changes in semantics. Moreover, the authors of the MIG metric have thoroughly laid out the exact interpretation of different ranges of MIG score as shown in figure 4.2. From this figure, we can observe that state-of-the-art methods already push the MIG towards its theoretical limits. Therefore, any incremental improvement in the state-of-the-art MIG is significant.

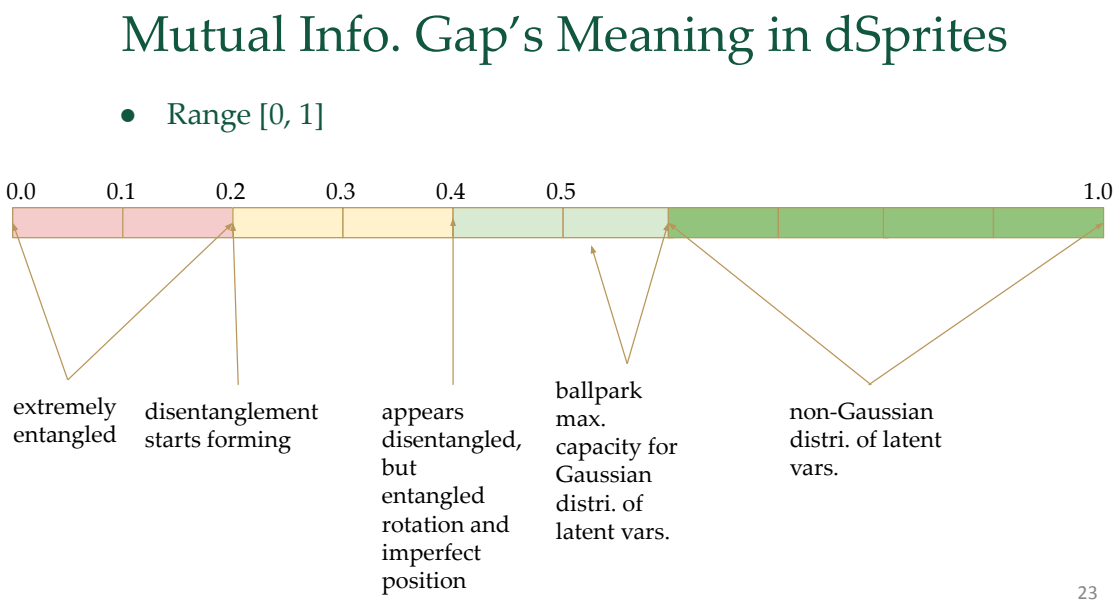


Figure 4.2: (Slide 23 from the oral defense presentation) The interpretation of MIG scores [30] for the dSprites dataset. Scores that are above 0.5 need to go beyond assuming a Gaussian distribution for the latent variables, namely a mixture of Dirac deltas [30]. However, Dirac deltas would have a high dimenwion-wise KL-divergence with a factorized Gaussian setup.

5 Experiments with Discussion

5.1 Setup

We selected the dSprites dataset [25] given that it has been extensively used in the literature, its ground truth factors of variation and can be measured against, and has simplicity in the pixel values. The ground-truth factors of variation are x-position, y-position, scale, orientation, and shape. The shape can vary from an ellipse, heart, or square.

The augmentations explored were the random rotation (90, 180, 270 degrees clockwise), horizontal flip, and random noise injection. These were chosen because they had straightforward implementations C and could be analyzed for specific behavior. For example, say one has a dSprite image of a heart. When this image of a heart gets rotated 90, 180, or 270 degrees counter-clockwise, then the heart still stays as a heart and the size of the heart stays the same. Therefore, since both the shape and size are preserved, $k = 1$ or $k = 2$ are two ideal hyperparameters to try out. Therefore, we would be able to analyze the latent traversals from the first one or two latent dimensions and visually tell if the first one or two dimensions correspond to shape and/or scale. The horizontal flip has a similar situation where both the shape and the scale remain constant.

Another intuition comes from the observation that when the number of chosen dimensions for the latent spaces exceeds the number of ground truth factors of variation, the remaining latent dimensions are constant. For example, there are 5 ground truth factors of variation in the dSprites dataset. When one chooses the number of latent dimensions to be 10, then 5 of the latent dimensions are constant (each with a very minuscule value around 0), while the remainder of the latent dimensions captures factors of variation. Therefore, $k = 5$ and $k = 6$ were additional hyperparameters tested.

Random noise augmentation has a slightly different situation. Instead of reconstructing noisy input, we made the objective to denoise it instead. This has grounding from the literature on denoising autoencoders [48]. In addition, all factors of variation should have the same value between an input and its noisy augmentation, because noise is not considered a factor of variation in the dSprites dataset. Instead of testing our model's generalization reconstruction accuracy, We investigated the effect of random noise on the disentanglement ability with the MIG score.

For the FactorVAE hyperparameters, We selected $\gamma = 10$ since that was the choice used by the FactorVAE authors and their ControlVAE authors [28, 31].

We also plotted training metrics from the standard VAE terms (reconstruction loss and KL divergence) [5] and FactorVAE terms (discriminator accuracy and estimated total correlation) [28], along with the new regularization term we propose (k-factor similarity loss). The x-axis is the current training iteration, and

the y-axis is the minibatch average of a training metric. Discriminator accuracy measures the percentage of correct classifications of latent samples from the VAE and of factorized latent samples through the permutation trick [28]. For the KL divergence, the dimension-wise KL divergences, "mean" KL divergence, and the "total" KL divergences are plotted. A dimension-wise KL divergence means for each latent variable, across a minibatch, what its average KL divergence is. The "total" KL div. means the average sum of all the dimension-wise KL divergences. The "mean" KL div. means the average across the minibatch and dimensions.

The GitHub repo. can be found here `https://github.com/joegenius98/Undergrad_Honors_Thesis` to reproduce experimental results.

5.2 Rotation

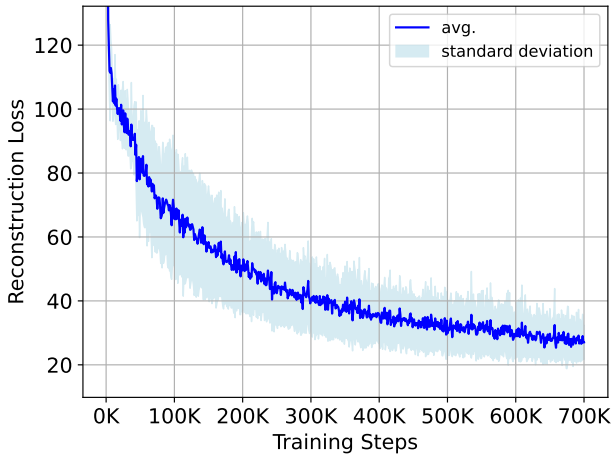
From an initial exploration where $k \in \{1, 2, 5, 6\}$ and $\alpha \in \{1, 2, 5\}$, we have obtained the top two results as described in table 5.1. Graphed training metrics are shown in figures 5.1 and 5.5.

Model	MIG	Training Steps
1FactorVAE ($\alpha = 2$)	$\mathbf{0.5691 \pm 0.0485}$	**700K**
6FactorVAE ($\alpha = 2$)	$\mathbf{0.5664 \pm 0.0341}$	**700K**
FactorVAE ($\gamma = 10$)	0.5340 ± 0.0443	700K
ControlVAE (KL = 16)	0.5628 ± 0.0222	1200K
FactorVAE ($\gamma = 10$)	0.5625 ± 0.0443	1200K
β-VAE ($\beta = 100$)	0.5138 ± 0.0371	1200K

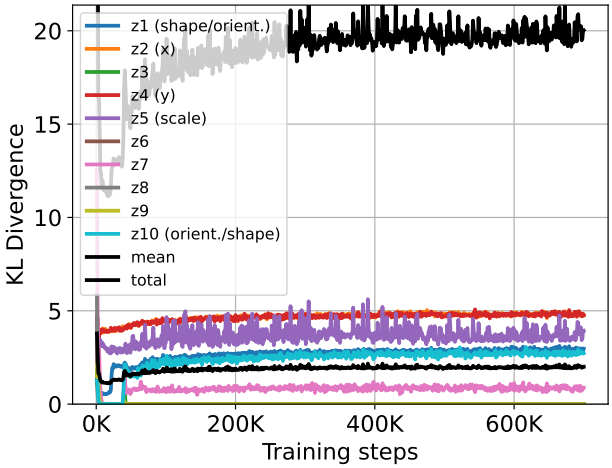
Table 5.1: Performance comparison of baseline methods vs. the rotation-augmented kFactorVAE, using the Mutual Information Gap (MIG) [30] as the disentanglement metric averaged over 5 random seeds. The higher the MIG, the better. kFactorVAE ($k = 1, \alpha = 2$), referred to as 1FactorVAE ($\alpha = 2$) for conciseness, outperforms the other state-of-the-art models with a standard deviation comparable to that of FactorVAE ($\gamma = 10$). 6FactorVAE ($\alpha = 2$) also outperforms the other models with a lower variance. Moreover, these methods take fewer training steps to achieve these results. The 1200K training results were retrieved from [31]. A training session of 700K steps is the default in the FactorVAE [28] implementation.

The training graph results in figures 5.1 and 5.5, which are associated with the best results shown in table 5.1, are consistent with the literature [28, 31] in terms of the average lines for reconstruction loss, KL divergence, and the estimated total correlation. However, the standard deviations for the reconstruction loss, discriminator accuracy, and estimated total correlation are relatively high. This might be explained by the randomness of the rotation augmentation itself, as it is not fixed to solely 90, 180, or 270 degrees. Discriminator accuracy results were not shown in FactorVAE's paper [28], but given its link to the estimated total correlation, discriminator accuracy seems to be consistent too. Our proposed regularization term: the k-factor similarity loss, has a slight downhill trend as expected, given the small α weight.

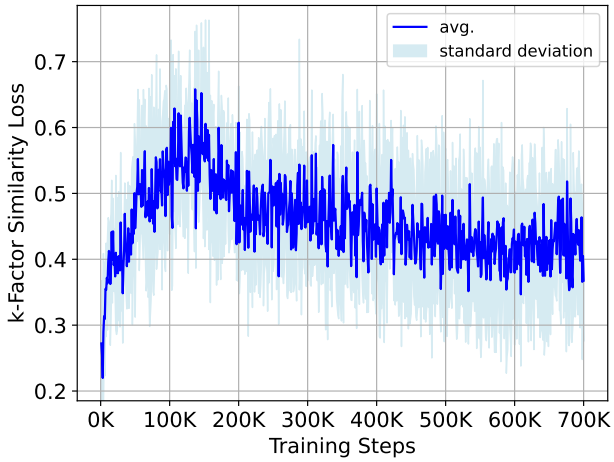
Extending the caption of figure 5.1, the reason that the KL divergence graph associated with the median



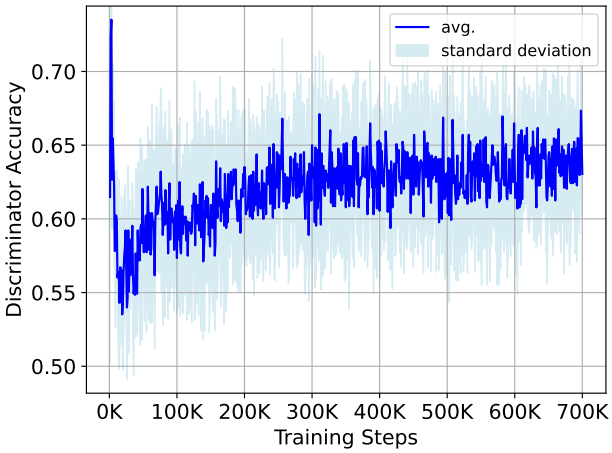
(a) Reconstruction Loss



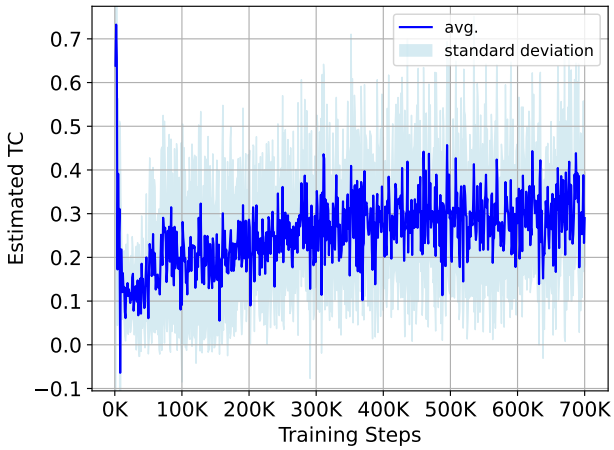
(b) KL Divergence with factors of variation



(c) k-Factor Similarity Loss



(d) Discriminator Accuracy



(e) Total Correlation

Figure 5.1: Rotation-augmented 1FactorVAE ($\alpha = 2$) training result graphs based on the average and one standard deviation from the distribution of the 5 seeds. The KL divergence graph associated with the seed with the median MIG score: 0.5367 is shown.

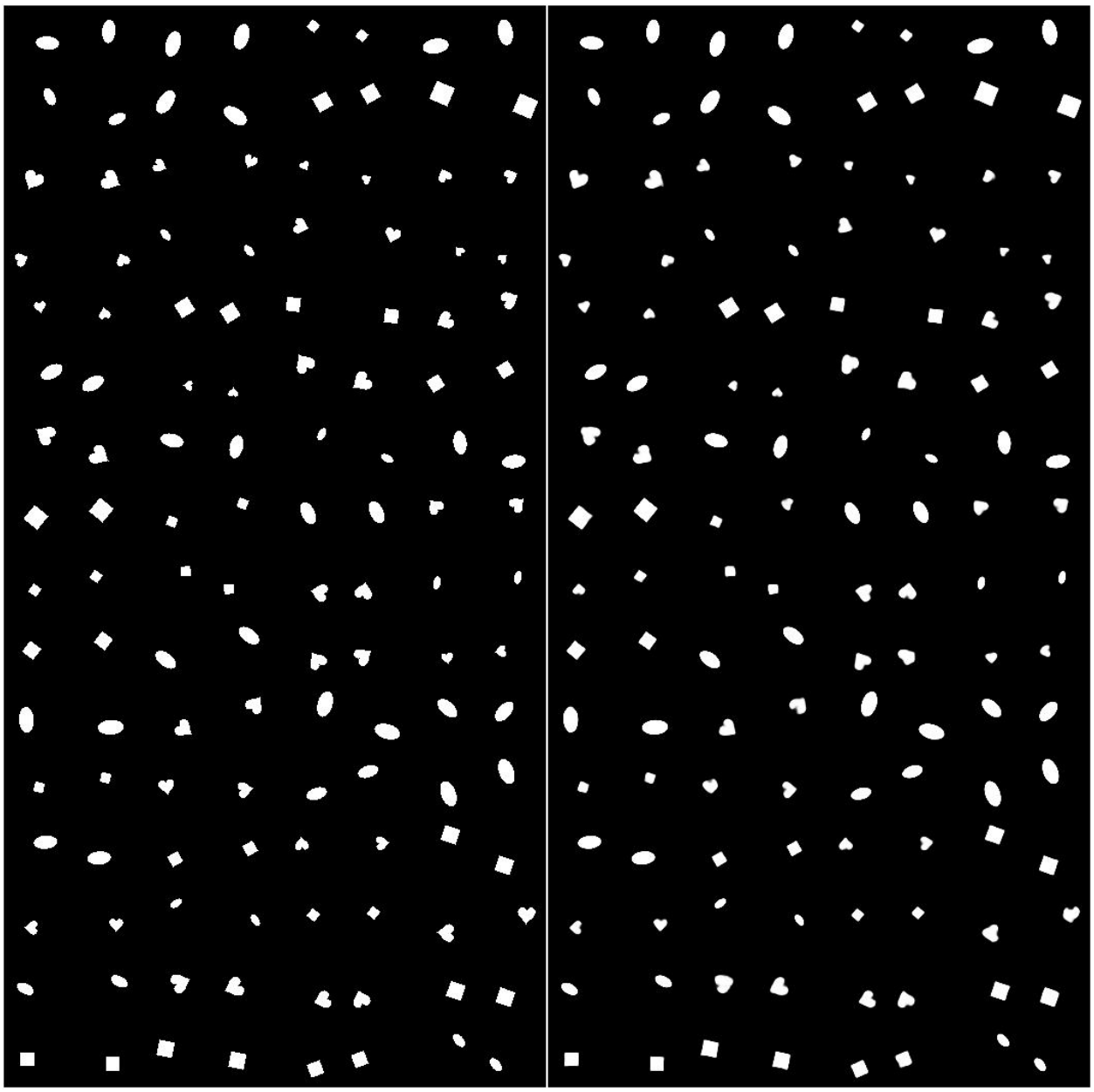
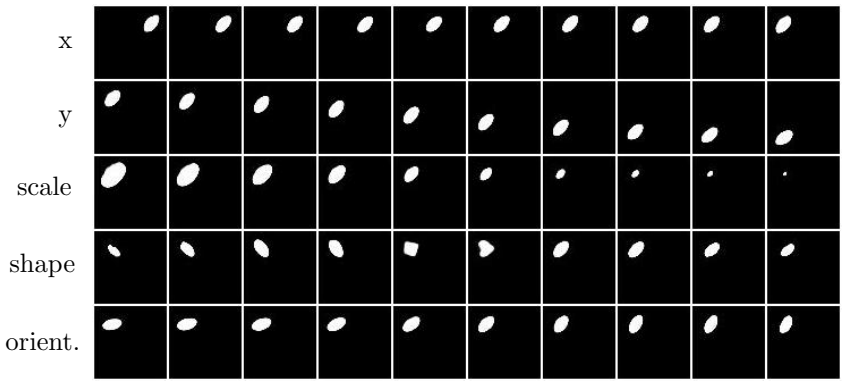
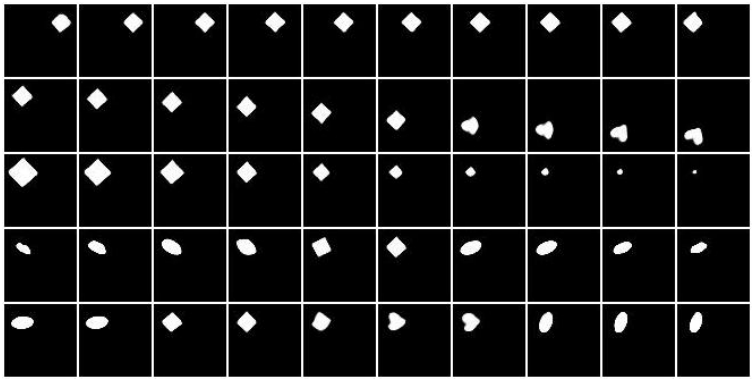


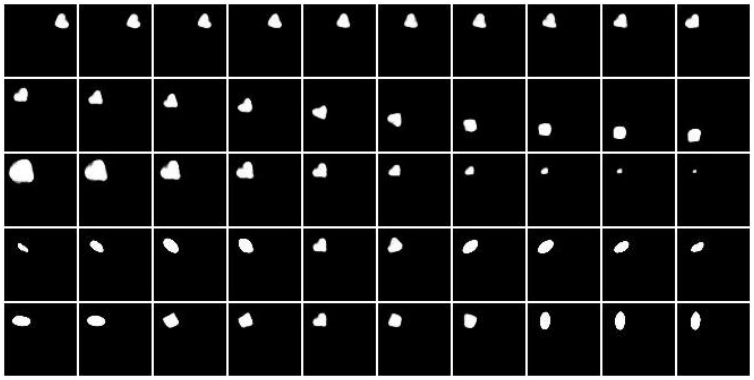
Figure 5.2: Reconstructions for rotation-augmented 1FactorVAE ($\alpha = 2$) at the final iteration 700K on seed 1, which is associated with the median MIG score amongst the 5 seeds. The left side represents a minibatch of 128 dSprites dataset images, and the right side represents their respective reconstructions. Each adjacent pair of images, starting from the top left corner, represents an image and its rotation augmentation. Each image has only one shape and is blended into the other images because of the black pixels.



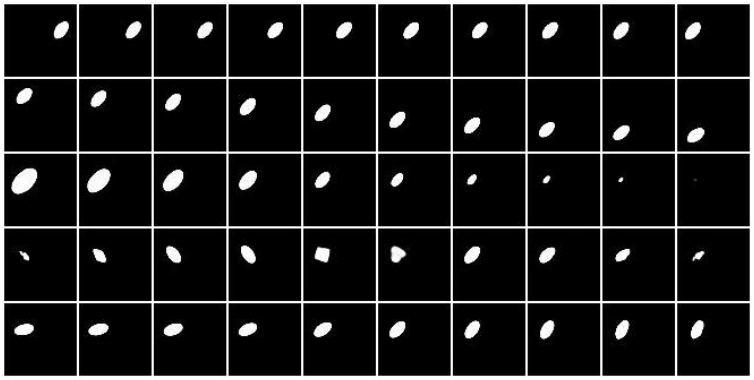
(a) Ellipse dSprite image latent traversal



(b) Square dSprite image latent traversal



(c) Heart dSprite image latent traversal



(d) Random dSprite image latent traversal

Figure 5.3: Latent traversal samples for rotation-augmented 1FactorVAE ($\alpha = 2$). (a) showcases the title of the factor of variation for each row. From the top row to the bottom row per image traversal, the factors of variation are sorted in decreasing dimension-wise KL divergence error. Each row goes in increments of $\frac{2}{3}$, starting from -3 and going up to 3 inclusively. "orient." is short for orientation.

(a) Ellipse dSprite image latent traversal on the 1st dimension

(b) Square dSprite image latent traversal on the 1st dimension

(c) Heart dSprite image latent traversal on the 1st dimension

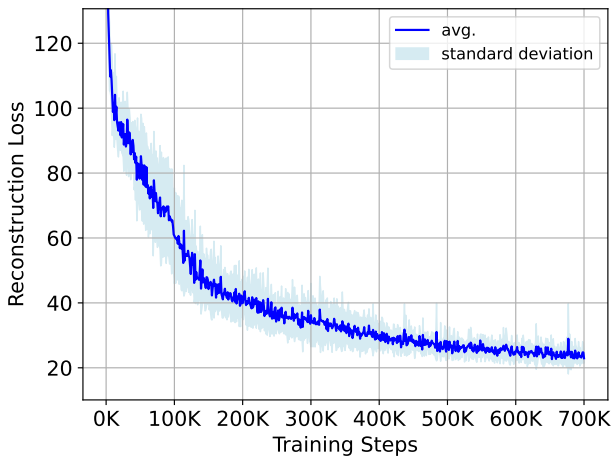
(d) Random dSprite image latent traversal on the 1st dimension

Figure 5.4: Latent traversal samples for rotation-augmented 1FactorVAE ($\alpha = 2$) on the 1st dimension

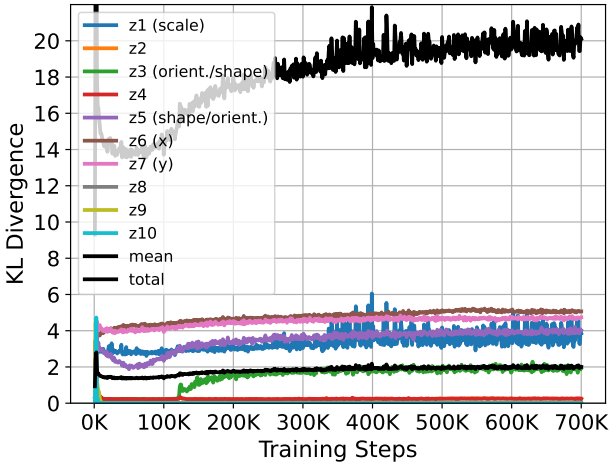
MIG score was plotted as opposed to plotting the average of five seeds as with the other graphs is that positions of the factors of variations tend to switch up per seed run. For example, while the 2nd latent dimension corresponds to the x-position factor, in another seed, the x-position factor corresponds with the 5th latent dimension instead.

Analyzing the rotation-augmented latent traversals in the first k dimensions, for 1FactorVAE ($\alpha = 2$), figure 5.4 demonstrates that between an image and its rotation augmentation, the model enforces the shape factor of variation to be shared in value between an image and its rotation. However, this factor is still entangled with the orientation factor, as FactorVAE faced the same issue [28]. This issue is also observed with the bottom two rows for each image's traversal in figure 5.3. For 6FactorVAE ($\alpha = 2$), figure 5.8 demonstrates the model enforced the factors of scale, shape, and x-position to be constant between an image and its rotation. The rest of the six factors are constant. This is all expected except for the x-position because the x-position of a dSprite is not guaranteed to be constant after a rotation is performed. Perhaps the conflict with enforcing the x-position factor value to be the same between an image and its rotation explains the relatively higher k-factor similarity loss in 6FactorVAE ($\alpha = 2$) 5.5 compared to that of 1FactorVAE ($\alpha = 2$) 5.1. Despite this counter-intuitive result, where ideally the x-position factor would be replaced by another constant factor, kFactorVAE seems to be able to produce robustly high MIG scores even when the k factors expected to be shared between an image and its rotation do not align with the actual results. However, as with 1FactorVAE ($\alpha = 2$), 6FactorVAE ($\alpha = 2$) still struggles with disentangling orientation and shape.

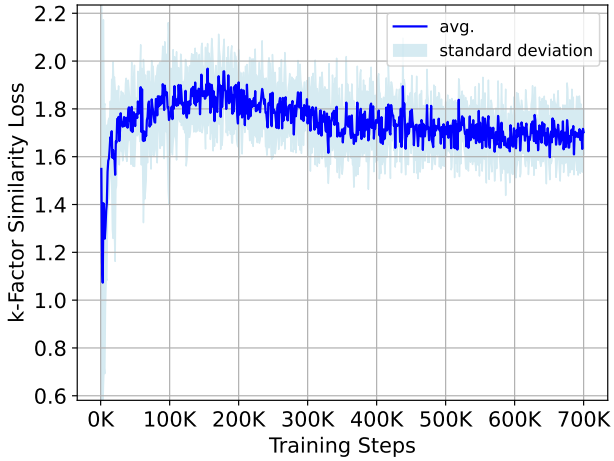
The outlier in table 5.2 is associated with $k = 6$ and $\alpha = 1$. The seed value of 5 is causing the lower average and higher variance. Without that seed, the score would have been 0.542 ± 0.0324. This shows that our regularization term does not overcome potential lower outliers/higher variances in the disentanglement



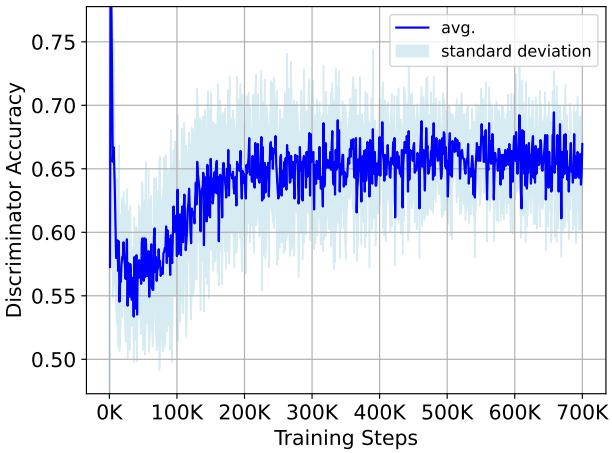
(a) Reconstruction Loss



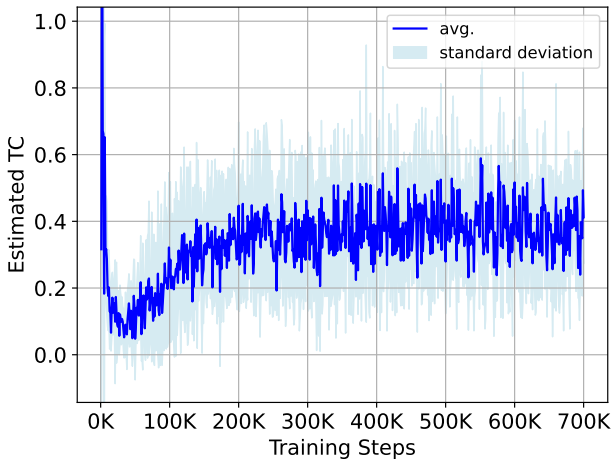
(b) KL Divergence with factors of variation



(c) k-Factor Similarity Loss



(d) Discriminator Accuracy



(e) Total Correlation

Figure 5.5: Training result graphs for rotation-augmented 6FactorVAE ($\alpha = 2$) with the same setup as figure 5.1. The median MIG score associated with the KL divergence graph is 0.5786.

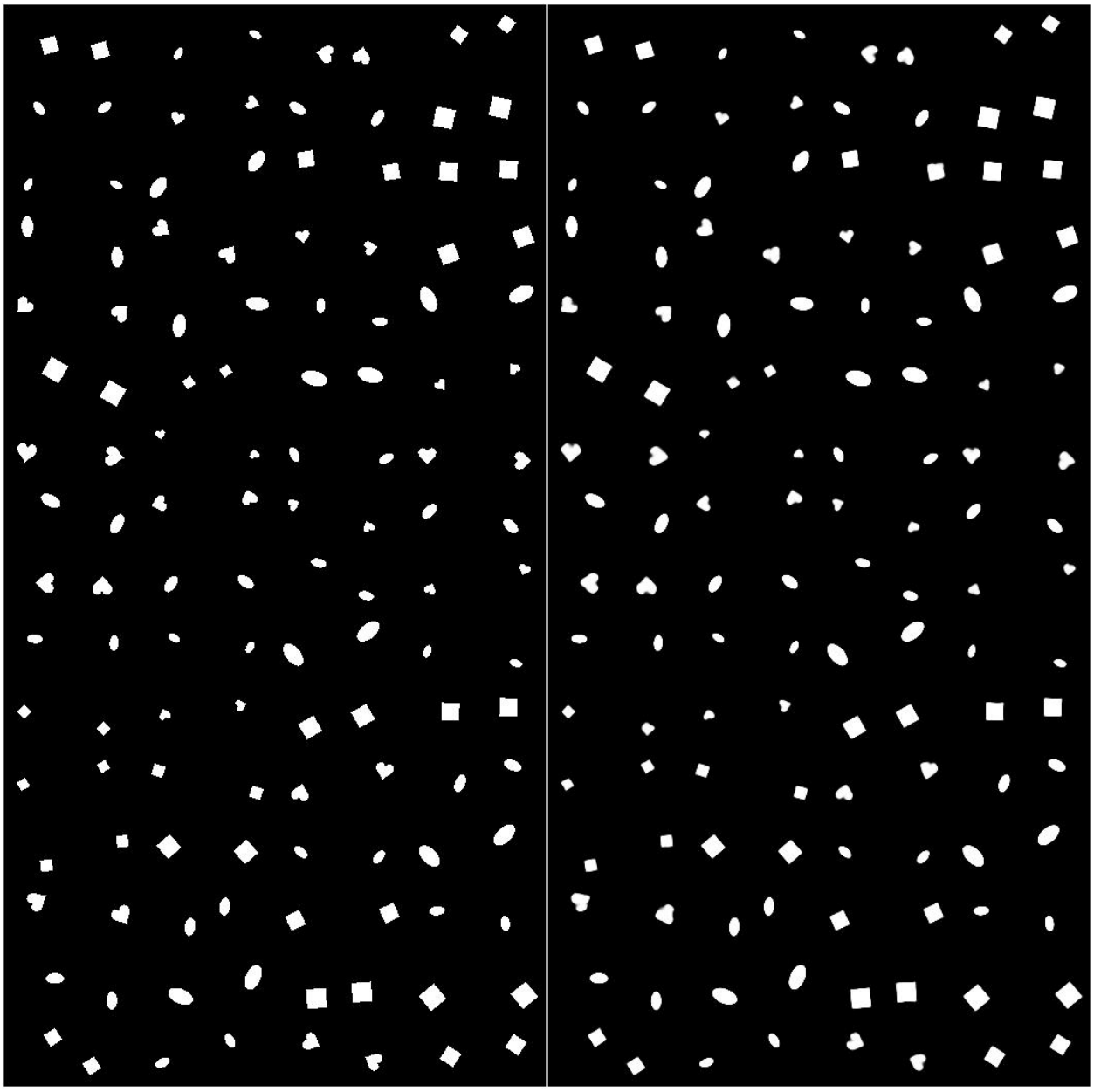
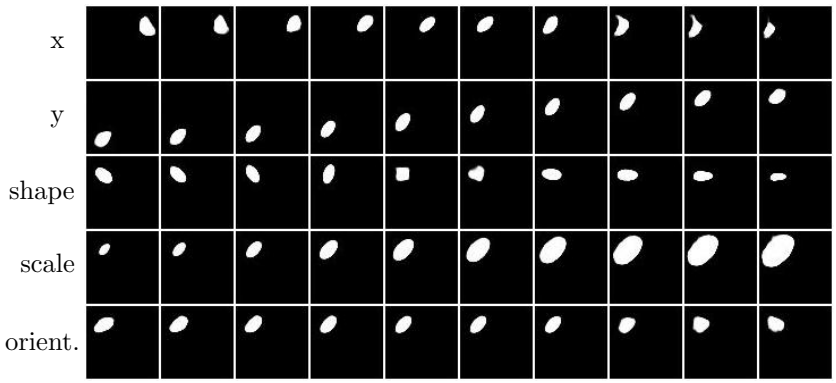


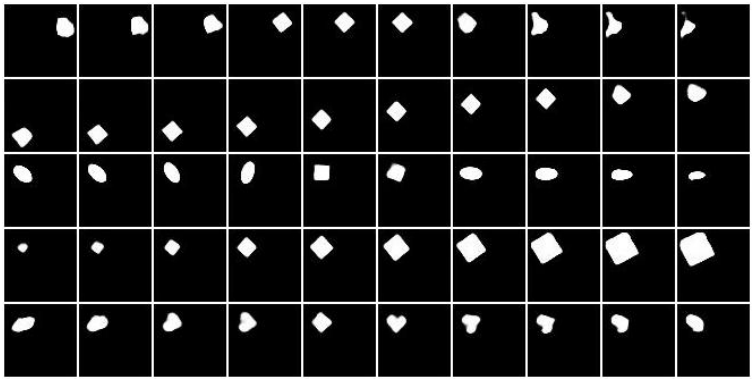
Figure 5.6: Reconstructions for rotation-augmented 6FactorVAE ($\alpha = 2$) with the same setup as figure 5.2.

Rotation	$k = 1$	$k = 2$	$k = 5$	$k = 6$
$\alpha = 1$	0.521 ± 0.0244	0.5249 ± 0.0399	0.5049 ± 0.0443	0.4833 ± 0.1341
$\alpha = 2$	$\mathbf{0.5691} \pm 0.0485$	0.5411 ± 0.0389	0.521 ± 0.0259	$\mathbf{0.5664} \pm 0.0341$
$\alpha = 5$	0.5189 ± 0.0107	0.5319 ± 0.0245	0.5371 ± 0.0352	0.5473 ± 0.0208

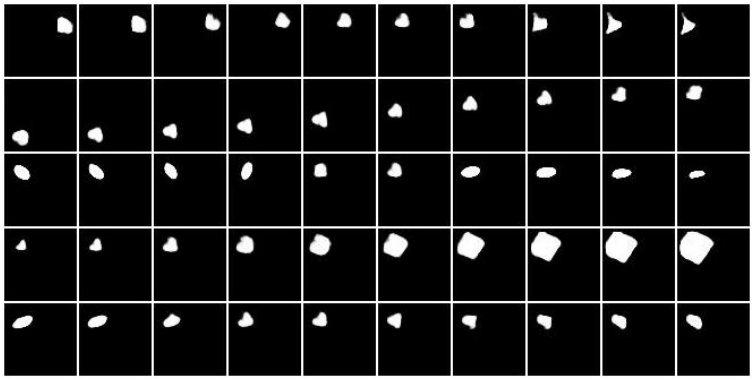
Table 5.2: MIG score results from hyparameter exploration of k and α with rotation.



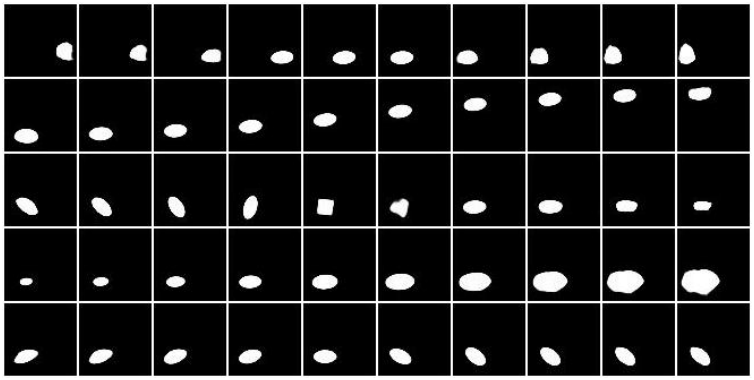
(a) Ellipse dSprite image latent traversal



(b) Square dSprite image latent traversal

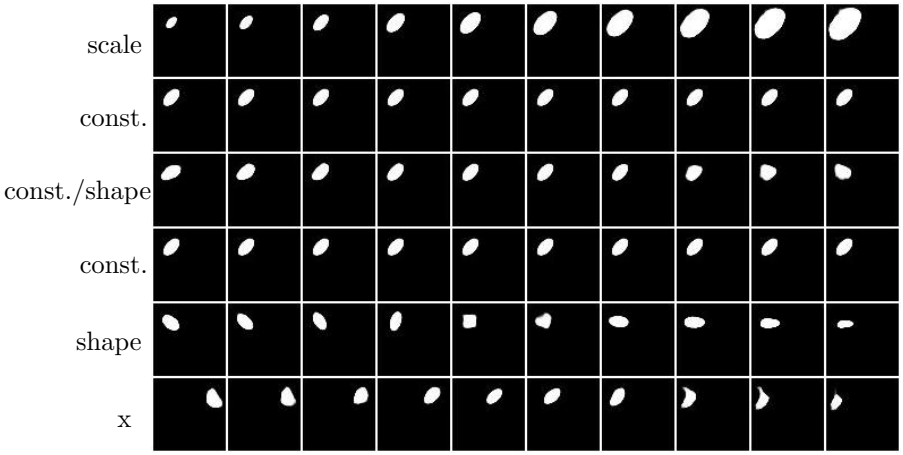


(c) Heart dSprite image latent traversal

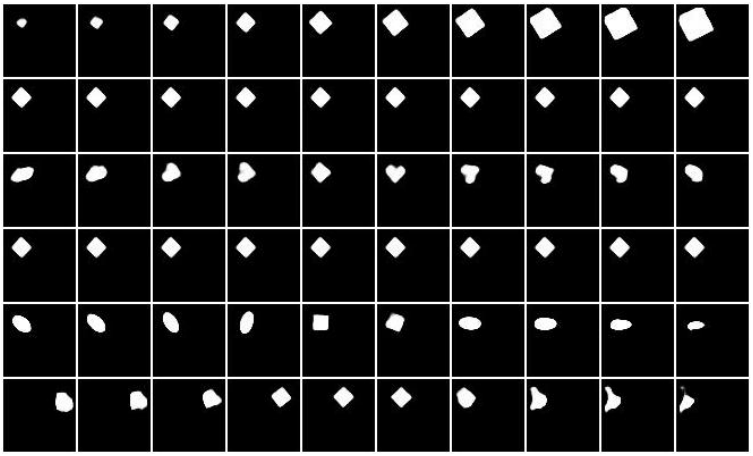


(d) Random dSprite image latent traversal

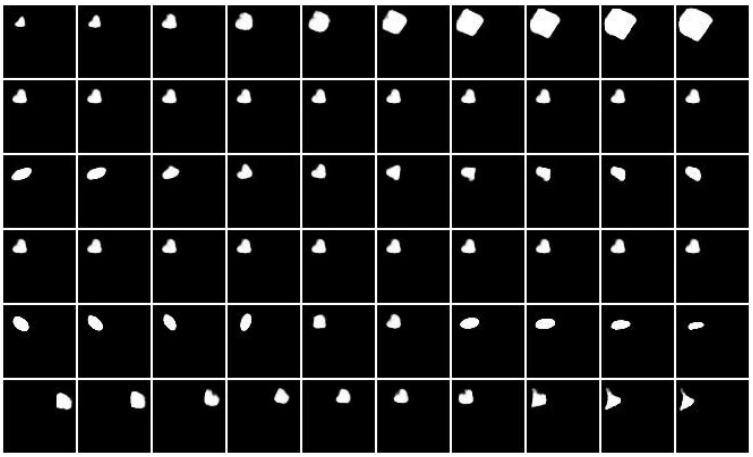
Figure 5.7: Latent traversal samples for rotation-augmented 6FactorVAE ($\alpha = 2$) with the same setup as figure 5.3.



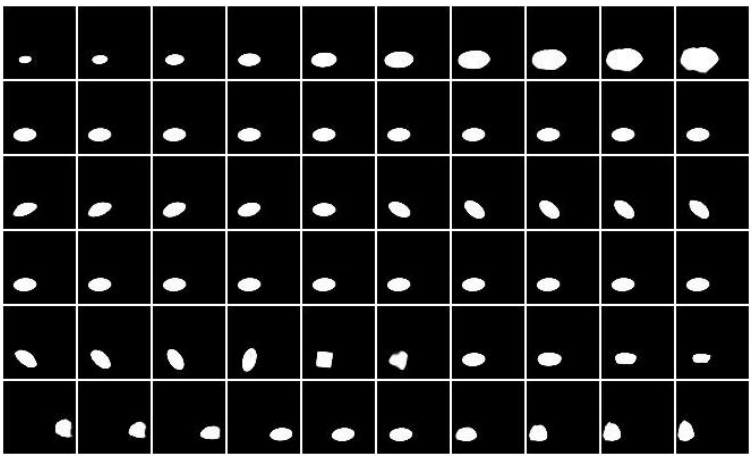
(a) Ellipse dSprite image latent traversal on the first 6 dimensions



(b) Square dSprite image latent traversal on the first 6 dimensions.



(c) Heart dSprite image latent traversal on the first 6 dimensions



(d) Random dSprite image latent traversal on the first 6 dimensions

Figure 5.8: Latent traversal samples for rotation-augmented 6FactorVAE ($\alpha = 2$) on the first 6 dimensions. "const." is short for "constant", and it indicates a latent factor that collapsed to the prior. It is not a legitimate factor of variation.

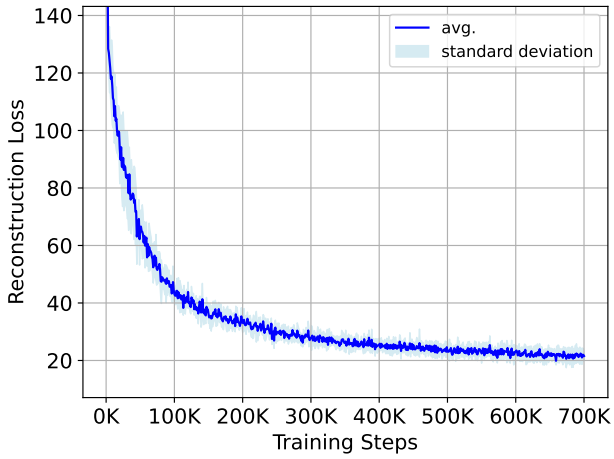
metric, even with an inductive bias shift from unsupervised learning to self-supervised learning. Locatello et al. (2019) [8] have shown that disentanglement metric scores for unsupervised disentanglement methods, in general, are heavily influenced by the regularization strength hyperparameter and the seed, and the objective function has less impact. Therefore, our method does not completely overcome the same issues that unsupervised learning approaches face.

Higher values of α were tested, but their results are not shown here due to poor results. For example, $\alpha = 100$ had a reconstruction error sitting around 500 to 600 and every latent variable collapsed to the prior, given that their traversals were constant. In that instance, the model probably prioritized the regularization term over the reconstruction loss and the KL divergence because of the very high α weight, while the k-factor similarity loss was some multiple of 10^{-6}.

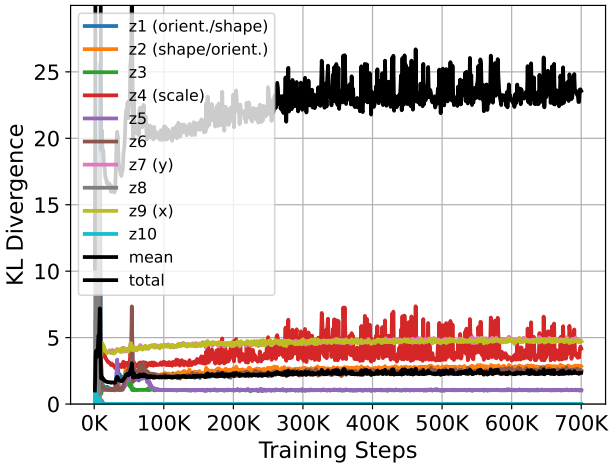
5.3 Horizontal Flip

The hyperparameter exploration was $k \in \{1, 2, 5, 6\}$ and $\alpha \in \{1, 2, 5\}$, just as for the rotation augmentation. In this case, from tables 5.3 and 5.4, none of the scores outperformed ControlVAE and only one set of hyperparameters ($k = 6$, $\alpha = 5$) that has a comparable MIG metric to that of ControlVAE. Despite this lack of success, on average across all the 12 hyperparameter combinations searched, the horizontal flip augmentation has a higher average MIG metric and a lower standard deviation. For the rotation augmentation, those quantities are 0.5306 and 0.0401 respectively. For the horizontal flip augmentation, they are 0.5365 and 0.0271 respectively.

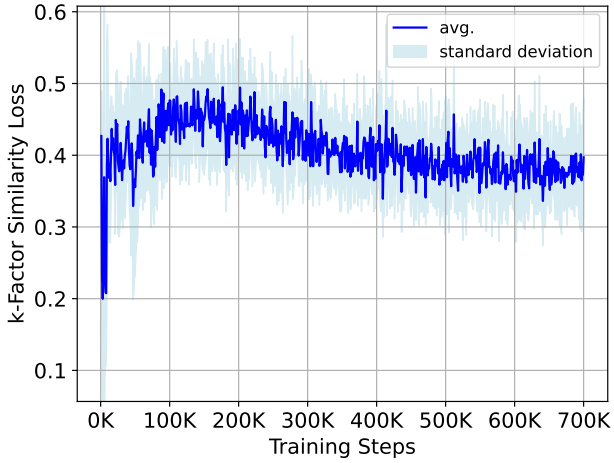
Discussing the results, once again, the training graph results in figure 5.9, which are associated with the



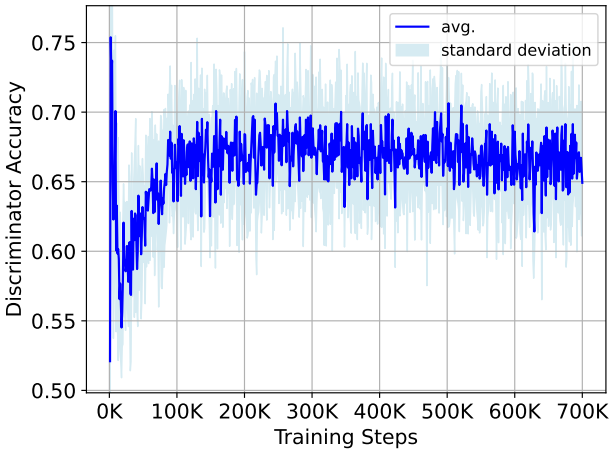
(a) Reconstruction Loss



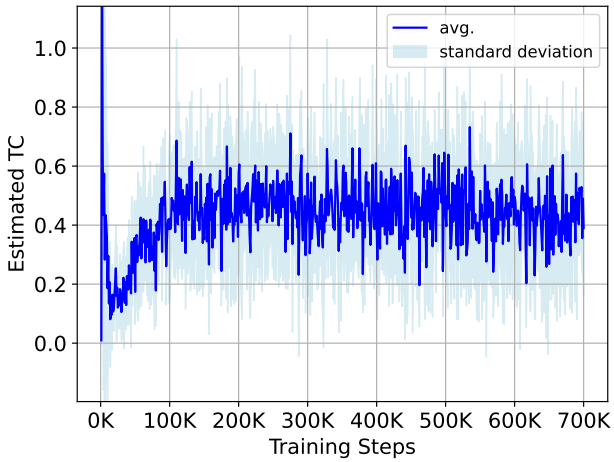
(b) KL Divergence with factors of variation



(c) k-Factor Similarity Loss



(d) Discriminator Accuracy



(e) Total Correlation

Figure 5.9: Training result graphs for horiz.-flip-augmented 6FactorVAE ($\alpha = 5$) with the same setup as figure 5.1. The median MIG score associated with the KL divergence graph is 0.5671.

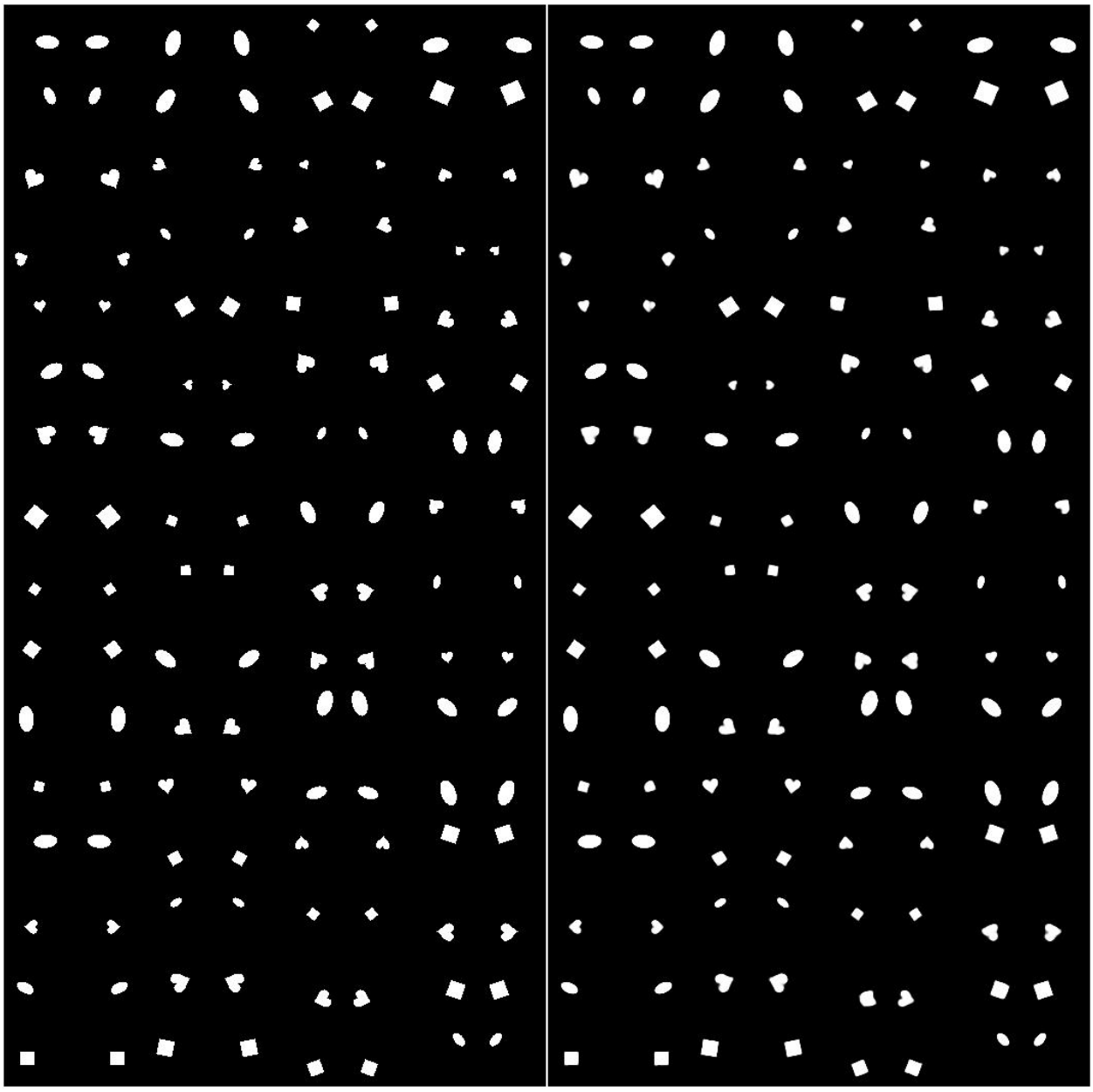
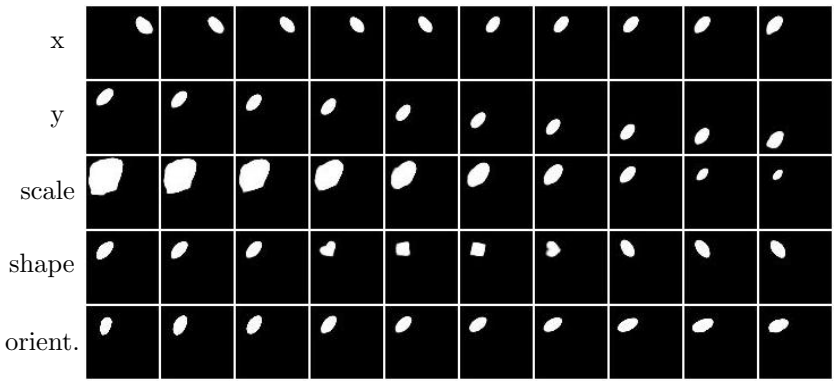
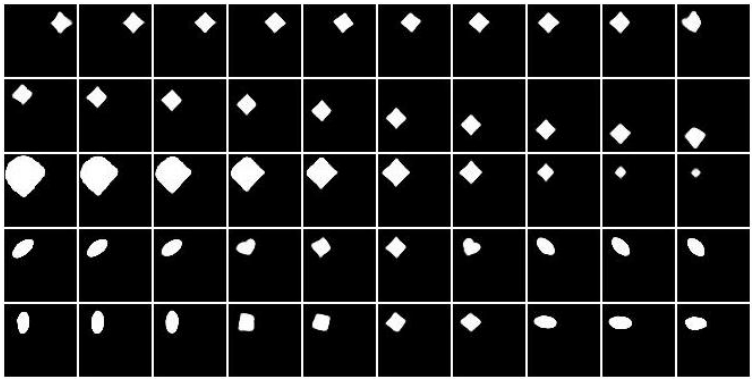


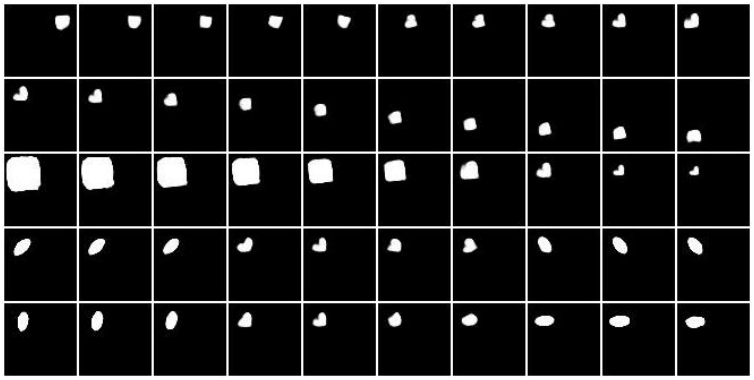
Figure 5.10: Reconstructions for horiz.-flip-augmented 6FactorVAE ($\alpha = 5$) with the same setup as figure 5.2.



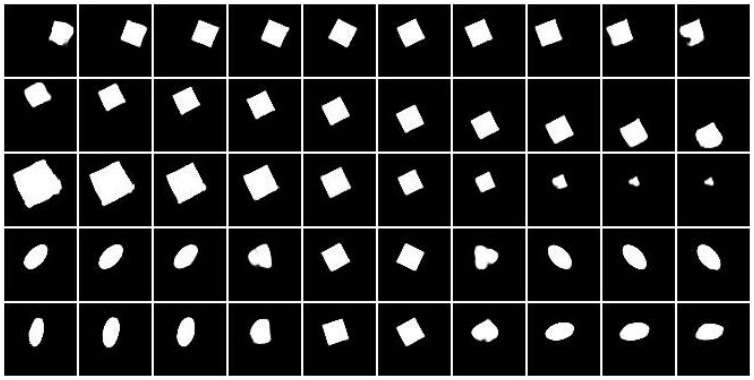
(a) Ellipse dSprite image latent traversal



(b) Square dSprite image latent traversal

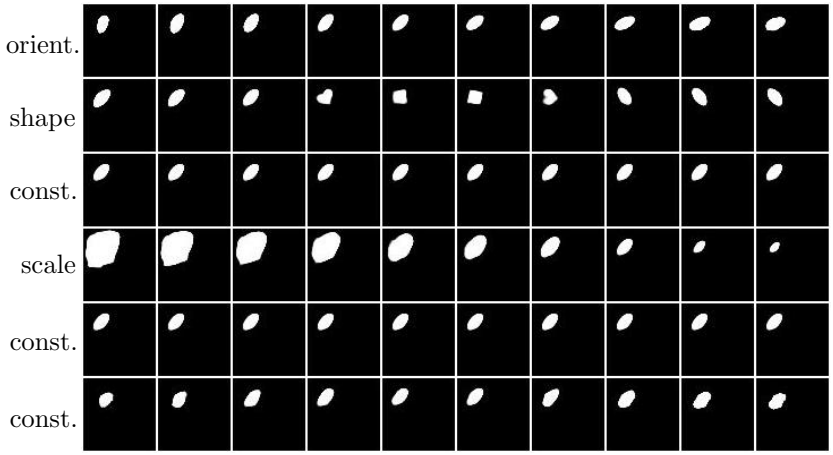


(c) Heart dSprite image latent traversal

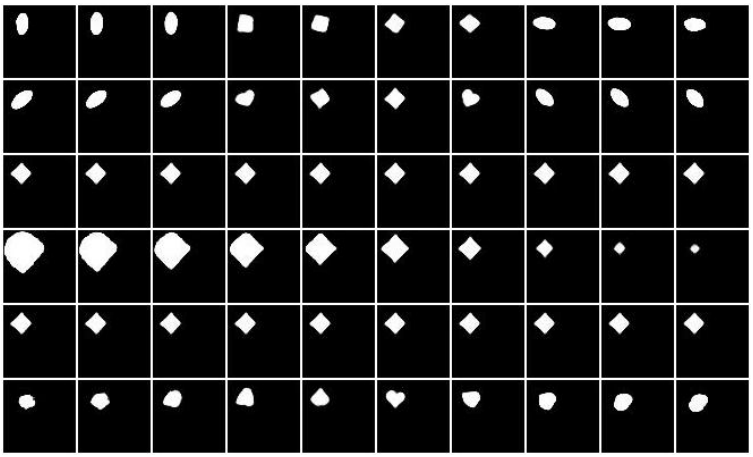


(d) Random dSprite image latent traversal

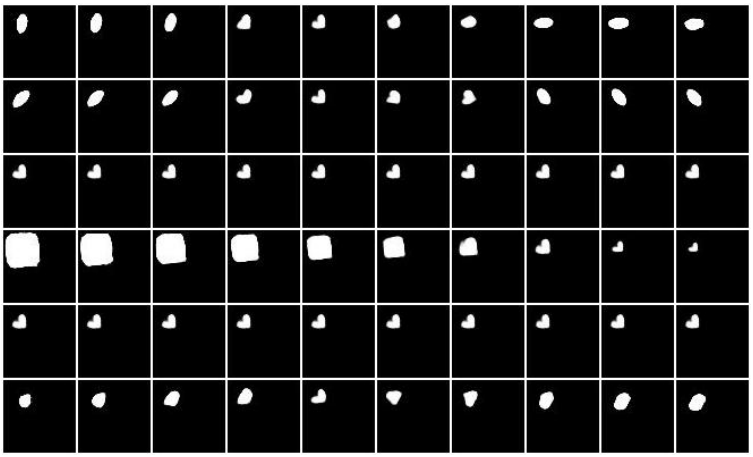
Figure 5.11: Latent traversal samples for horiz.-flip-augmented 6FactorVAE ($\alpha = 5$) with the same setup as figure 5.3.



(a) Ellipse dSprite image latent traversal on the first 6 dimensions



(b) Square dSprite image latent traversal on the first 6 dimensions



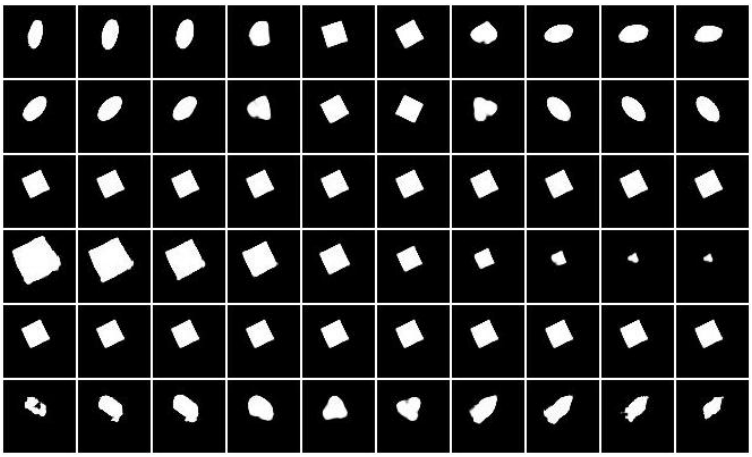
(c) Heart dSprite image latent traversal on the first 6 dimensions

Model	MIG	Training Steps
6FactorVAE ($\alpha = 5$)	$\mathbf{0.5617 \pm 0.0186}$	**700K**
FactorVAE ($\gamma = 10$)	0.5340 ± 0.0443	700K
ControlVAE (KL = 16)	0.5628 ± 0.0222	1200K
FactorVAE ($\gamma = 10$)	0.5625 ± 0.0443	1200K
β-VAE ($\beta = 100$)	0.5138 ± 0.0371	1200K

Table 5.3: Performance comparison of baseline methods vs. best horiz.-flip-augmented kFactorVAE using the Mutual Information Gap (MIG) [30] as the disentanglement metric, averaged over 5 random seeds. The higher the MIG, the better. 6FactorVAE ($\alpha = 5$) has a comparable MIG and a lower variance to ControlVAE.

Horizontal Flip	$k = 1$	$k = 2$	$k = 5$	$k = 6$
$\alpha = 1$	0.5341 ± 0.0232	0.538 ± 0.0255	0.5138 ± 0.0576	0.5272 ± 0.0185
$\alpha = 2$	0.5379 ± 0.0251	0.5247 ± 0.0205	0.5492 ± 0.0236	0.5199 ± 0.0618
$\alpha = 5$	0.5461 ± 0.0183	0.5354 ± 0.0153	0.5502 ± 0.0173	$\mathbf{0.5617 \pm 0.0186}$

Table 5.4: MIG score results for exploration of k and α with the horizontal flip augmentation. None of these results outperform ControlVAE, but the choice of $k = 6$ with $\alpha = 5$ comes the closest and has a less standard deviation.



(d) Random dSprite image latent traversal on the first 6 dimensions

Figure 5.12: Latent traversal samples for horiz.-flip-augmented 6FactorVAE ($\alpha = 5$) on the first 6 dimensions.

model with the best results in table 5.3, are similar to the results from the rotation augmentation, but this time, the reconstruction loss has a much lower standard deviation. This is probably because a horizontal flip has no randomness as the rotation implementation did. On the first six latent-dimension traversals on the best model 5.12, once again, every factor of variation is expected except for one. This time, it is orientation. A horizontal flip switches the orientation of a dSprite shape across the vertical axis. Therefore, when our regularization term involves an augmentation in which the orientation is not invariant, and our model enforces the orientation factor to be the same between an image and its augmentation anyway, that can have a negative effect on disentanglement.

5.4 Random Noise

Random noise had no noticeable effect, visually, on the latent traversals and on the training result graphs, compared to what we have seen already with rotation and horizontal flip. Additionally, as seen in figure 5.13, its ability to denoise is effective on the dSprites training dataset. However, from the same hyperparameter exploration range as mentioned the past couple of times, none of the MIG scores outperformed or were comparable to the ControlVAE baseline. The closest score to that baseline is 0.5502 ± 0.0198 from $k = 5$ and $\alpha = 5$. This demonstrates that, in terms of disentanglement, while the disentangled representations formed by kFactorVAE are nearly robust to noise, noise has a negative effect.

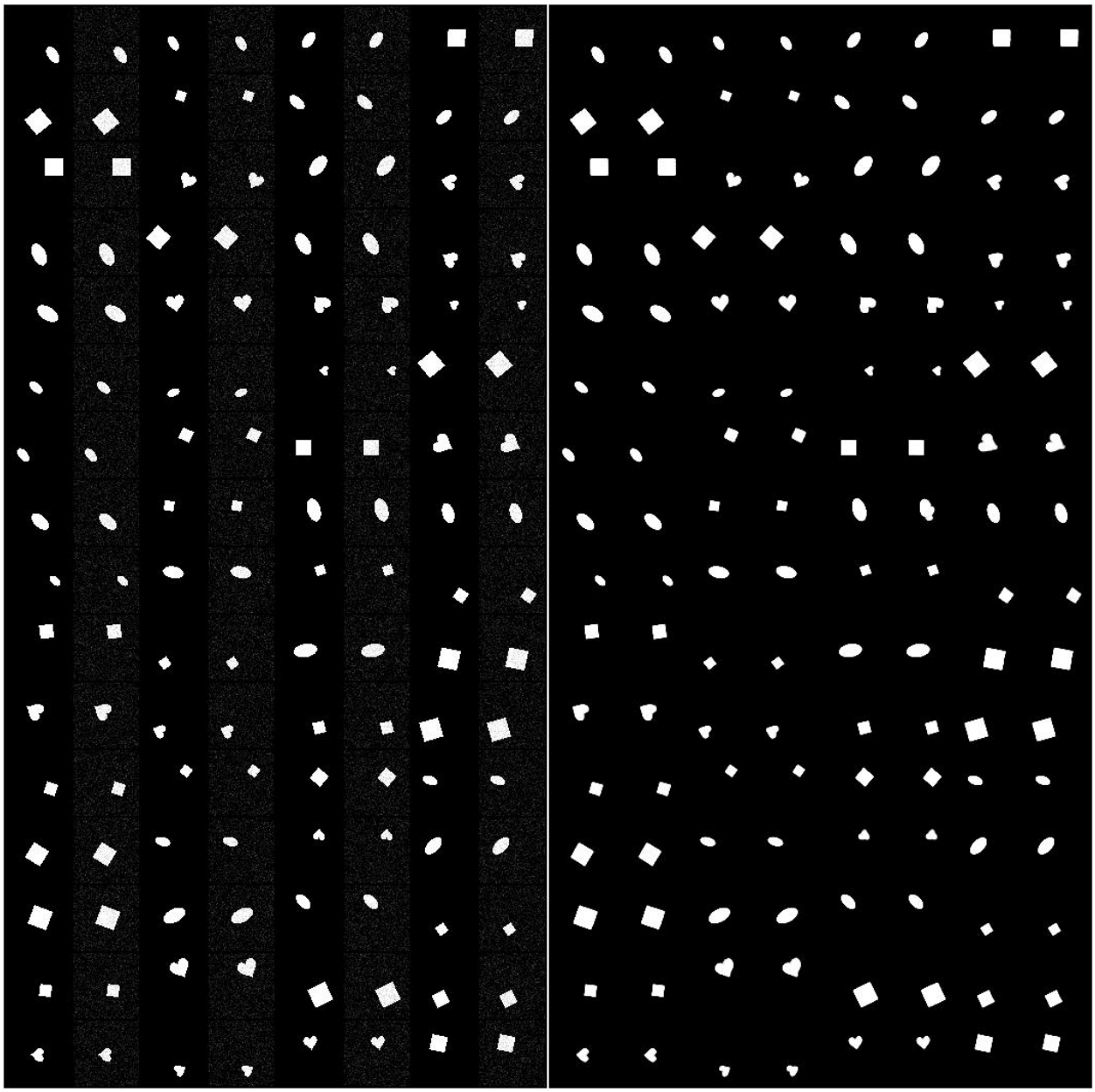


Figure 5.13: Reconstructions and denoising for noise-augmented 5FactorVAE ($\alpha = 5$) with the same setup as 5.2. The median score is 0.5518.

6 Conclusion

In this thesis, we proposed a self-supervised-based regularization term as an extension to the FactorVAE model, kFactorVAE, to encourage invariance between the values of shared factors of variation between an image and its augmentation. This extended model was then tested on three different augmentations: random rotation, horizontal flip, and Gaussian noise injection. For certain hyperparameter choices, in fewer training steps, kFactorVAE slightly outperformed its unsupervised counterparts in the state-of-the-art mutual information gap (MIG) metric or had a comparable performance with lower variance. When the intended invariant k factors do not match up to the first k factors seen in a latent traversal, the effect seems to be either neutral or negative. Therefore, an open avenue to explore is how to achieve better alignment to mitigate the chance of a negative effect on disentanglement. Qualitatively, the latent traversals are at least as consistent with the literature.

One limitation with the latent traversals is that the shape factor is still entangled with the orientation factor. Therefore, a critical extension to my proposed regularization term includes disentangling discrete factors from continuous factors (e.g. orientation). Several studies have been making progress on this challenge [33, 49, 50]. Merging our regularization term and/or an equivalent term for discrete factors with this branch of work may be promising.

Extensive testing in the hyperparameter spaces of k, α, and the augmentation type may also be explored, given higher computational resources, to perform a more thorough ablation study. The trend seems to be that higher values $k \geq 5$ and $\alpha \geq 2$ produce higher MIG scores, but there have been exceptions to that trend too. More sophisticated augmentation types could be explored, such as neural style transfer and other GAN-based augmentations, especially on datasets such as ImageNet [51] or CIFAR-10 [52]. Augmentation types such as that may allow invariance of an object's factors of variation to be robust against the background.

Another challenge is that there is a significant amount of variance in both training results and MIG scores amongst the distribution of 5 seeds, as is the case with unsupervised VAE-based models [8]. This makes sense with different augmentation types. Therefore, it will take more than just potentially augmentation-based learning with regularization to address this issue. Weakly-supervised models, based on similar ideas on accounting for invariant/variant factors, show promise towards this issue [42].

References

[1] J. Rocca, "Understanding variational autoencoders (vaes)," Mar. 2021.

[2] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut Learning in Deep Neural Networks," *Nature Machine Intelligence*, vol. 2, pp. 665–673, Nov. 2020. arXiv:2004.07780 [cs, q-bio].

[3] X. Wang, H. Chen, S. Tang, Z. Wu, and W. Zhu, "Disentangled Representation Learning," Nov. 2022. arXiv:2211.11695 [cs] version: 1.

[4] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," Apr. 2014. arXiv:1206.5538 [cs].

[5] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013 (revised 2022).

[6] A. Soleimany and A. Amini, "Mit 6.s191 (2022): Deep generative modeling."

[7] 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*, 2017.

[8] F. Locatello, S. Bauer, M. Lučić, G. Rätsch, S. Gelly, B. Schölkopf, and O. F. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning*, 2019. Best Paper Award.

[9] D. H. Wolpert, "What is important about the no free lunch theorems?," *CoRR*, vol. abs/2007.10928, 2020.

[10] X. Ren, T. Yang, Y. Wang, and W. Zeng, "Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view," in *International Conference on Learning Representations*, 2022.

[11] T. Wang, Z. Yue, J. Huang, Q. Sun, and H. Zhang, "Self-supervised learning disentangled group representation as feature," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 18225–18240, Curran Associates, Inc., 2021.

[12] J. Kahana and Y. Hoshen, "A contrastive objective for learning disentangled representations," 2022.

[13] S. Mo, Z. Sun, and C. Li, "Representation disentanglement in generative models with contrastive learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1531–1540, January 2023.

[14] R. S. Zimmermann, Y. Sharma, S. Schneider, M. Bethge, and W. Brendel, "Contrastive learning inverts the data generating process," 2021.

[15] F. J. Moreno-Barea, F. Strazzera, J. M. Jerez, D. Urda, and L. Franco, "Forward noise adjustment scheme for data augmentation," *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 728–734, 2018.

[16] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *ArXiv*, vol. abs/1712.04621, 2017.

[17] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2017.

[18] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 3964–3979, nov 2021.

[19] I. Shafkat, "Intuively understanding variational autoencoders," Feb. 2018.

[20] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," 2016.

[21] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum, "Deep convolutional inverse graphics network," 2015.

[22] 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.

[23] 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," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3762–3769, 2014.

[24] IEEE, *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, (Genova, Italy), 2009.

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

[26] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," 2018.

[27] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," 2018.

[28] H. Kim and A. Mnih, "Disentangling by factorising," 2019.

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

[30] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," 2019.

[31] 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*, pp. 8655–8664, PMLR, 2020.

[32] M. Kim, Y. Wang, P. Sahu, and V. Pavlovic, "Relevance factor vae: Learning and identifying disentangled factors," 2019.

[33] E. Dupont, "Learning disentangled joint continuous and discrete representations," 2018.

[34] A. Amini, A. P. Soleimany, W. Schwarting, S. N. Bhatia, and D. Rus, "Uncovering and mitigating algorithmic bias through learned latent structure," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 289–295, 2019.

[35] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[36] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," 2016.

[37] J.-Y. Zhu, Z. Zhang, C. Zhang, J. Wu, A. Torralba, J. B. Tenenbaum, and W. T. Freeman, "Visual object networks: Image generation with disentangled 3d representation," 2018.

[38] Z. Wu, D. Lischinski, and E. Shechtman, "Stylespace analysis: Disentangled controls for stylegan image generation," 2020.

[39] D. Bouchacourt, R. Tomioka, and S. Nowozin, "Multi-level variational autoencoder: Learning disentangled representations from grouped observations," 2017.

[40] F. Locatello, M. Tschannen, S. Bauer, G. Rätsch, B. Schölkopf, and O. Bachem, "Disentangling factors of variation using few labels," 2020.

[41] F. Träuble, E. Creager, N. Kilbertus, F. Locatello, A. Dittadi, A. Goyal, B. Schölkopf, and S. Bauer, "On disentangled representations learned from correlated data," 2021.

[42] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen, "Weakly-supervised disentanglement without compromises," 2020.

[43] 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 (CVPR)*, July 2017.

[44] T. Xiao, J. Hong, and J. Ma, "Dna-gan: Learning disentangled representations from multi-attribute images," 2018.

[45] 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*, pp. 9593–9602, 2021.

[46] X. Shen, F. Liu, H. Dong, Q. Lian, Z. Chen, and T. Zhang, "Weakly supervised disentangled generative causal representation learning," 2022.

[47] X. Liu, F. Zhang, Z. Hou, Z. Wang, L. Mian, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *CoRR*, vol. abs/2006.08218, 2020.

[48] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," ICML '08, (New York, NY, USA), p. 1096–1103, Association for Computing Machinery, 2008.

[49] Y. Jeong and H. O. Song, "Learning discrete and continuous factors of data via alternating disentanglement," 2019.

[50] B. Esmaeili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. Dy, and J.-W. van de Meent, "Structured disentangled representations," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics* (K. Chaudhuri and M. Sugiyama, eds.), vol. 89 of *Proceedings of Machine Learning Research*, pp. 2525–2534, PMLR, 16–18 Apr 2019.

[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[52] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[53] L. Weng, "From autoencoder to beta-vae," *lilianweng.github.io*, 2018.

A Encoder and Decoder Configurations

Encoder	**Decoder**
Input $1 \times 64 \times 64$ binary image	Input latent sample $\in \mathbb{R}^{10}$
32 out. chs., 4×4 conv., stride 2, pad 1, ReLU	128 out. chs., 1×1 conv. (equiv. to FC), ReLU
32 out. chs., 4×4 conv., stride 2, pad 1, ReLU	64 out. chs., 4×4 upconv, ReLU.
64 out. chs., 4×4 conv., stride 2, pad 1, ReLU	64 out. chs., 4×4 upconv., stride 2, pad 1, ReLU
64 out. chs., 4×4 conv., stride 2, pad 1, ReLU	32 out. chs., 4×4 upconv., stride 2, pad 1, ReLU
128 out. chs., 4×4 conv., stride 1, ReLU	32 out. chs., 4×4 upconv., stride 2, pad 1, ReLU
$2^*\|\mathbf{z}\|$ out. chs., 1×1 conv. (equivalent to FC layer)	1 out. ch., 4×4 upconv., stride 2, pad 1

B Discriminator Configuration

Layer	**Output Shape**	**Activation Function**
Input	(batch size, z_{dim})	None
Fully Connected	(batch size, 1000)	LeakyReLU 0.2
Fully Connected	(batch size, 1000)	LeakyReLU 0.2
Fully Connected	(batch size, 1000)	LeakyReLU 0.2
Fully Connected	(batch size, 1000)	LeakyReLU 0.2
Fully Connected	(batch size, 1000)	LeakyReLU 0.2
Fully Connected	(batch size, 2)	None

C Augmentation Implementations

Imports

```
import torch
from torchvision.transforms.functional import rotate, hflip
import random
```

Discrete random rotation:

```
def discrete_random_rotate(image):
    return rotate(image, random.choice([90, 180, 270]))
```

Horizontal flip:

```
def horizontal_flip(image):
    return hflip(image)
```

Random noise:

```
def gaussian_noise(image: torch.Tensor):
    std_dev = 0.1 * (image.max() - image.min())

    noisy_img = image + torch.randn(image.shape) * std_dev
    noisy_img = torch.clamp(noisy_img, 0, 1)
    return noisy_img
```

D k-Factor Similarity Loss Implementation

```
def k_factor_sim_loss(latent_samples: torch.Tensor, k):
    if k is None or k == 0:
        return 0

    assert latent_samples.size(0) % 2 == 0

    image_reprs_k, aug_reprs_k = None, None

    image_reprs_k = latent_samples[::2, :k]
    aug_reprs_k = latent_samples[1::2, :k]

    # k-factor similarity loss
    repr_diffs_k = image_reprs_k - aug_reprs_k

    # squared error
    repr_diff_norms_k = torch.sum(repr_diffs_k ** 2, dim = 1)

    # mean squared error
    return torch.mean(repr_diff_norms_k)
```