Orthogonality-Enforced Latent Space in Autoencoders: An Approach to Learning Disentangled Representations

Jaehoon Cha[1] Jeyan Thiyagalingam[1]

Abstract

Noting the importance of factorizing (or disentangling) the latent space, we propose a novel, non-probabilistic disentangling framework for autoencoders, based on the principles of symmetry transformations that are independent of one another. To the best of our knowledge, this is the first deterministic model that is aiming to achieve disentanglement based on autoencoders using only a reconstruction loss without pairs of images or labels, by explicitly introducing inductive biases into a model architecture through Euler encoding. The proposed model is then compared with a number of state-of-the-art models, relevant to disentanglement, including symmetry-based models and generative models. Our evaluation using six different disentanglement metrics, including the unsupervised disentanglement metric we propose here in this paper, shows that the proposed model can offer better disentanglement, especially when variances of the features are different, where other methods may struggle. We believe that this model opens several opportunities for linear disentangled representation learning based on deterministic autoencoders.

1. Introduction

Learning generalizable representations of data is one of the fundamental aspects of modern machine learning (Rudin et al., 2022). In fact, better representations are more than a luxury now, and is a key to achieving generalization, interpretability, and robustness of machine learning models (Bengio et al., 2013; Brakel & Bengio, 2017; Spurek et al., 2020). One of the primary and desired characteristics of the learned representation is factorizability or disentanglement, so that latent representation can be composed of multiple, independent generative factors of variations. The disentanglement process renders the latent space features to become independent of one another, providing a basis for a set of novel applications, including scene rendering, interpretability, and unsupervised deep learning (Eslami et al., 2018; Iten et al., 2020; Higgins et al., 2021).

Deep generative models, particularly that build on variational autoencoders (VAEs) (Kingma & Welling, 2013; Kumar et al., 2017; Higgins et al., 2017; Tolstikhin et al., 2018; Burgess et al., 2018; Chen et al., 2018; Burgess et al., 2018; Kim & Mnih, 2018; Zhao et al., 2019), have shown to be effective in learning factored representations. Although these approaches have advanced the disentangled representation learning by regularizing the latent spaces, there are a number of issues that limit their full potential: (a) VAE-based models consist of two loss components, and balancing these loss components is a well known issue (Asperti & Trentin, 2020) (b) it is almost impossible to honor the idealized notion of having a known prior distribution for VAEs in practical settings (Takahashi et al., 2019; Asperti & Trentin, 2020; Zhang et al., 2020; Aneja et al., 2021) and, (c) factorizing the aggregated posterior in the latent space does not guarantee corresponding uncorrelated representations (Locatello et al., 2019). An alternative approach for achieving disentangled representations is through seeking irreducible representations of the symmetry groups (Cohen & Welling, 2014; Higgins et al., 2018; Painter et al., 2020; Wang et al., 2021; Tonnaer et al., 2022), where the aim is to find latent space transformations that are independent of one another, underpinned by well-defined mathematical frameworks based on group theory. As this group of methods exploits the notion of transitions between samples, they require pairs of images representing the transitions (Cohen & Welling, 2014; Worrall et al., 2017; Painter et al., 2020) or equivalent labels (Tonnaer et al., 2022). Finally, there are attempts to use Generative Adversarial Networks (GANs) to learn factored representations by maximizing the mutual information between a subset of latent variables and the observations. Regardless of the approach, as shown in Locatello et al. (2019), it is fundamentally impossible to learn disentangled representations without having inductive biases on either the model or the dataset, and both VAE- and symmetry-based

[1] Scientific Computing, Rutherford Appleton Laboratory, Science and Technology Facilities Council, United Kingdom. Correspondence to: Jaehoon Cha <jaehoon.cha@stfc.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

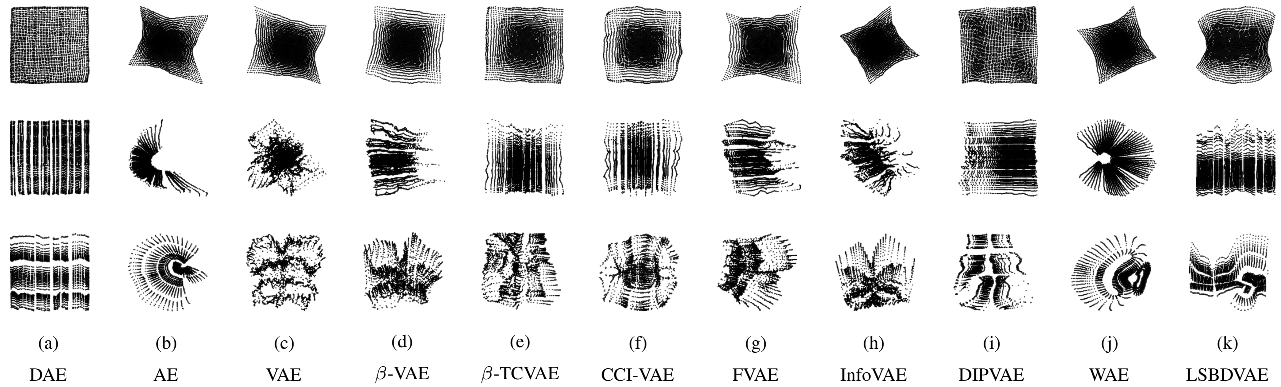


Figure 1. Latent spaces learned by different models. Ideal latent space should cover a two-dimensional grid (Higgins et al., 2018). The first, second and third rows show the latent spaces learned from three datasets, namely, XY, 2D Arrow, and 3D Airplane datasets, respectively. Columns correspond to models at the bottom. It can be seen that the proposed model, DAE, achieves the best disentanglement.

approaches exemplify implicitly embedding inductive bias.

Despite the advances, we note a number of issues in the existing approaches about how they address disentanglement. Firstly, the majority of the VAE-based approaches are probabilistic, and as such, the quality of the disentanglement depends on ideal or near-ideal priors, and on the process of learning the correct posteriors for a given data. Secondly, the majority of symmetry-based disentangling approaches need pairs of images or labels, even in the unsupervised setting, owing to the requirements around inductive bias. Thirdly, none of these models conform to the formal definition of a linear disentangled representation proposed by Higgins et al. (2018). Finally, and most importantly, none of the existing approaches have the unsupervised approach for introducing inductive biases (required for disentanglement) both on the models and on the datasets, essentially demanding labels or image pairs. Motivated by these shortcomings, in this paper, we propose a novel approach for deriving disentangled representation learning, with the following key contributions: We,

- propose a totally unsupervised approach for introducing inductive bias into the model and data, without requiring pairs of images or labels,
- propose a non-probabilistic approach that does not involve any priors or learning posteriors,
- as a consequence, train the proposed model by minimizing only a reconstruction loss between the inputs and the outputs of the model,
- introduce orthogonality in the latent space to enable the changes of output from changes of different latent dimensions to be orthogonal, and
- propose a new unsupervised metric, namely, Grid Fitting Score (GF-Score), to quantify the disentanglement, echoing the aspiration of an ideal disentanglement measure outlined in Higgins et al. (2018).

As such, the proposed approach, which we name Disentangling Auto-Encoder (DAE), offers a theoretically sound framework for learning independent multi-dimensional vector subspaces, and hence towards learning disentangled representations. To the best of our knowledge, this is the first attempt to actually implement a disentanglement approach using deterministic autoencoders by using only a reconstruction loss, especially without pairs of images or labels, and hence in a truly unsupervised manner. We provide a glimpse into the capability of the proposed model for disentanglement using three datasets compared against ten other models, which are either autoencoder-based probabilistic models or symmetry-based disentangled models, that do not require any labels or pairs of inputs in Figure 1.

The rest of this paper is organized as follows. In Section 2 we review the related work, focusing on VAE-based and symmetry-based approaches. This is then followed by a derivation of AE-based non-probabilistic approach for deriving disentangled representations in Section 3. In Section 4, we perform a detailed evaluation to decide the overall performance of the proposed model, using twelve baseline models, eight datasets, and six disentanglement metrics, and discuss our findings. We then conclude the paper in Section 5 with directions for further research. Given the space constraints, we highlight the prominent results in the main part of the paper, while providing the remaining set of results and relevant material as part of the Appendix.

2. Related Work

2.1. Disentanglement

Disentangled representation learning focuses on learning a set of independent factors containing useful but minimal information for a given task, such that their variations are orthogonal to each other while accounting for the entire dataset (Bengio et al., 2013; Higgins et al., 2018). This essentially entails a method or a set of methods for decoupling correlations between latent variables. A large body of work around disentanglement, and the ideal properties of a disentangled representation can be found in Ridgeway (2016); Eastwood & Williams (2018); Ridgeway & Mozer (2018); Zaidi et al. (2020). Among a number of desirable properties of a disentangled representation, modularity, compactness and explicitness are three critically important properties. A number of metrics have been proposed in the literature to quantify these properties (Higgins et al., 2017; Kim & Mnih, 2018; Eastwood & Williams, 2018; Chen et al., 2018; Do & Tran, 2019; Sepliarskaia et al., 2019). In our work, we use the notions outlined in Zaidi et al. (2020), where the metrics are divided into three classes, namely, Intervention-based, Predictor-based, and Information-based metrics. These metrics are all used in a supervised manner and can be of indicators for the robustness of the representation to noise, and for the non-linearity of the relationships between learnt representations and ground truth factors, in addition to the three properties outlined above.

2.2. Autoencoder-based Probabilistic Models

AE-based probabilistic generative models are realized by replacing the conventional encoder $E_{\boldsymbol{\phi}}$ and decoder $D_{\boldsymbol{\theta}}$ with probabilistic counterparts. The probabilistic encoder, denoted by $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, is used to approximate the intractable true posterior, and the probabilistic decoder, denoted by $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, is used to reconstruct the $\boldsymbol{x}$ from $\boldsymbol{z}$ (Kingma & Welling, 2013). The majority of the previous work on disentangled representation learning are based on probabilistic models, particularly building on VAE. They enforce regularization in the latent space that either regularizes the approximate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ or the aggregate posterior $q(\boldsymbol{z}) = \frac{1}{N}\sum_{i=1}^{N} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})$, as summarized in Tschannen et al. (2018). The overall objective of the majority of the VAE-based methods can be expressed as:

$$L_{recon}(\boldsymbol{\phi}, \boldsymbol{\theta}) + L_{reg}(\boldsymbol{\phi}) \tag{1}$$

where $L_{reg}(\boldsymbol{\phi})$ is a regularizer of the concerned generative model. A carefully designed regularizer should enable the model to achieve better disentanglement, either by controlling the capacity of the latent space, or by measuring the total correlation between latent variables. However, it is worth noting that factorizing aggregated posterior using regularizers does not guarantee linear disentangled representations (Locatello et al., 2019). Recently, a deterministic approach is proposed in Berman et al. (2023) based on Koopman operators with structured spectra using an additional spectral loss function. Although the motivation in that paper is different from the other probabilistic models in that the proposed model in Berman et al. (2023) is a deterministic model, the model still requires an additional regularizer to achieve disentanglement. We summarize the regularization terms of nine state-of-the-art generative models in Appendix A (See Columns 2 and 3 of Table 3).

2.3. Symmetry-Based Disentangling Model

While Higgins et al. (2018) proposed a formal definition of linear disentangled representations, it was generic, so that no specific architecture, model or technique were defined. As such, it does not provide an actual mechanism for learning such disentangled representations, albeit providing a formal definition, which is essential for this purpose. From the definitions in Higgins et al. (2018), a symmetry group can be decomposed as a product of multiple subgroups, if suitable subgroups can be identified. This can provide an intuitive method for disentangling the latent space, if subgroups that independently act on subspaces of a latent space can be found. If actions applied on each of the subgroups affect only the corresponding subspace, these actions are called disentangled group actions. In other words, disentangled group actions only change a specific property of the state of an object, and leaves the other properties invariant. If there is a transformation in a vector space of representations, corresponding to a disentangled group action, the representation is called a disentangled representation.

The concept and implementation of symmetry-based disentangled representations were proposed using pairs of images in Cohen & Welling (2014). However, owing to the limitation around commutative Lie groups, upon which this model is built upon, the real world applicability of the technique from Cohen & Welling (2014), especially across a range of diverse datasets, are limited. Following a formal definition for linear disentangled representations in Higgins et al. (2018), there has been a considerable amount of effort to learn the transitions between images (Caselles-Dupré et al., 2019; Quessard et al., 2020; Painter et al., 2020). The transitions between images are learned by treating each transition as a sequence of transitions until the base transition relies on pairs of images and by using additional networks. An alternative approach is to rely on labels, for example, as in Tonnaer et al. (2022), where they propose two Diffusion VAE-based methods (Rey et al., 2019), namely semi-supervised and unsupervised, along with a new metric called LSBD (Linear Symmetry-Based Disentanglement metric). The former model relies on labels, while the latter does not. As such, the latter model is directly relevant to our work, and, we use this as one of the baselines for our evaluation (See Section 4).

3. Framework for DAE

The deterministic, and hence, non-probabilistic, approach we propose here, builds on the autoencoder architecture (rather than variational autoencoders). We provide the relevant background on the disentangled representations from Higgins et al. (2018) in the Appendix A.2. In this section, we define a necessary mathematical framework and a corresponding neural network architecture implementing the proposed disentangling autoencoder.

3.1. Association between the Disentangled Representation and Autoencoder

The definition of disentangled group actions from Higgins et al. (2018) assumes that a group G can be decomposes a direct product $G = G_1 \times \cdots \times G_n$. Here we formalize disentangled group action by a relation to relax the condition, and connect the group action to a generative process. Let us consider a group G, which is generated by a set $S = \{s_1, s_2, ..., s_n\}$ subject to a set R of relations among elements in S.

Definition 3.1. Let W be a set of world-states and suppose we have a group action $\cdot : G \times W \rightarrow W$. Then, we say that the action is disentangled by the relation R if there is a decomposition $W = W_1 \times \cdots \times W_n$ and actions $\cdot_i :< s_i > \times W_i \rightarrow W_i, i \in \{1, ..., n\}$ such that:

1. $(s_1^{\epsilon_1}, ..., s_n^{\epsilon_n}) \cdot (w_1, ..., w_n) = (s_1^{\epsilon_1} \cdot w_1, ..., s_n^{\epsilon_n} \cdot w_n)$ and,
2. if any elements $g \in G$ can be uniquely expressed in the form of $g = s_1^{\epsilon_1} \cdots s_n^{\epsilon_n}$ for some $\epsilon_i \in \mathbb{Z}$ by the relation R.

With the definition of an equivariant map in place (A.2), disentangling a latent space relies on finding a corresponding group action $\cdot : G \times Z \rightarrow Z$ so that the symmetry structure of W is reflected in an agent's representations, Z. This can be achieved if the following condition is satisfied:

$$g \cdot f(\boldsymbol{w}) = f(g \cdot \boldsymbol{w}) \quad \forall g \in G, \boldsymbol{w} \in W. \tag{2}$$

where $f : W \rightarrow Z$ is a mapping from world-states to an agent's representations. However, in general, one cannot control the nature of the generative process $b : W \rightarrow O$ leading from world-states to observations, O. In addition, without loss of generality, we can easily assume that the generative process b is an equivariant map.

Theorem 3.2. *Suppose a generative process b is an equivariant map satisfying , $g \cdot b(\mathbf{w}) = b(g \cdot \mathbf{w}) \quad \forall g \in G, \mathbf{w} \in W$. Then, there exists a function f that satisfies (2) if an inference process $h : O \rightarrow Z$ is an equivariant map satisfying,*

$$g \cdot h(\mathbf{o}) = h(g \cdot \mathbf{o}) \quad \forall g \in G, \mathbf{o} \in O. \tag{3}$$

Proof. Proof in the Appendix A.3. □

Following the Theorem 3.2, this assumption leads to the fact that the goal of disentangling is the same as finding an inference process $h : O \rightarrow Z$ satisfying,

$$g \cdot h(\boldsymbol{o}) = h(g \cdot \boldsymbol{o}) \quad \forall g \in G, \boldsymbol{o} \in O. \tag{4}$$

Although there is no guarantee that one can find a compatible action $\cdot : G \times Z \rightarrow Z$ satisfying (4), if h is bijective then (4) can be expressed as follows,

$$h^{-1} \cdot g \cdot h(\boldsymbol{o}) = g \cdot h^{-1} \cdot h(\boldsymbol{o}) \tag{5}$$

However, if h is a bijective function, simple neural network-based models cannot learn the overall equivariant map. Yet, the equivariant map, such as one outlined in equation 5 can be learned by autoencoders with inductive biases both on the model and the datasets, which is the central contribution of this paper. To show this mapping, let h and h^{-1} be an encoder, $E_{\boldsymbol{\phi}}$, and a decoder, $D_{\boldsymbol{\theta}}$, of an autoencoder. Then, the group action $\cdot : G \times Z \rightarrow Z$ can be defined as follows:

$$\begin{tikzcd}
G \times Z \arrow[r, "\cdot_{\mathbf{Z}}"] \arrow[d, "id_G \times D_{\boldsymbol{\theta}}"'] & Z \arrow[d, "D_{\boldsymbol{\theta}}"] \\
G \times O \arrow[r, dashed, "\cdot_{\mathbf{O}}"] & O
\end{tikzcd}$$

where $E_{\boldsymbol{\phi}} : O \rightarrow Z$, and $\cdot_{\mathbf{Z}}$ and $\cdot_{\mathbf{O}}$ are group actions on Z and O, respectively. This shows that the equivariant map can, indeed, be learned by an autoencoder. However, this is not without a number of challenges, which we discuss in Section 3.2 below.

3.2. Introducing Disentangled Representations into AutoEncoders

To connect the Definition 3.1 to disentangled representation in the latent space, it is worth noting that n-th root of unity is a cyclic group, and a vector addition in the latent space enables natural transition between latent variables. We achieve a disentangled group action using a set of relations R in the latent space, that maps $(s_1^{\epsilon_1}, ..., s_n^{\epsilon_n})$ to $(e^{i\alpha_1\epsilon_1}, ..., e^{i\alpha_n\epsilon_n})$, where $e^{ix} = \cos x + i \sin x$. However complex numbers are undesirable in machine learning. We overcome this issue by introducing the Euler encoding, E, which is defined as,

$$E(\boldsymbol{z}) = (cos(2\pi z_1), sin(2\pi z_1), ..., cos(2\pi z_n), sin(2\pi z_n)) \tag{6}$$

where n is the the number of dimensions of the latent space.

Theorem 3.3. *Let E be a Euler encoding and $\boldsymbol{A} : \mathbb{R}^{2n} \rightarrow \mathbb{R}^m$ be an injective linear transformation where $m > 2n$. For $\alpha \in (0, 1)$ and $i \in \{1, ..., n\}$, let $T_i^\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^n$ by $T_i^\alpha(\boldsymbol{x}) = (x_1, ..., x_i + \alpha, ..., x_n)$. Then $\boldsymbol{A} \cdot E(T_i^\alpha(\boldsymbol{z})) = \boldsymbol{A} \cdot E(T_j^\beta(\boldsymbol{z}))$ if and only if $i = j$ and $\alpha = \beta$.*

Proof. For $\boldsymbol{z} \in \mathbb{R}^n$, let $\boldsymbol{A} \cdot E(T_i^\alpha(\boldsymbol{z})) - \boldsymbol{A} \cdot E(T_j^\beta(\boldsymbol{z})) = \boldsymbol{0}$ and define

$$S_i^\alpha = \begin{bmatrix} \boldsymbol{I}_{2(i-1)} & & & \\ & cos(2\pi\alpha) & -sin(2\pi\alpha) & \\ & sin(2\pi\alpha) & cos(2\pi\alpha) & \\ & & & \boldsymbol{I}_{2(n-i)} \end{bmatrix}$$

Since $E(T_i^\alpha(\boldsymbol{z})) = S_i^\alpha \cdot E(\boldsymbol{z})$, $\boldsymbol{A} \cdot (S_i^\alpha - S_j^\beta) \cdot E(\boldsymbol{z}) = \boldsymbol{0}$.

(a) If $i \neq j$, then $\boldsymbol{A}$ is a zero transformation, which is a contradiction.

(b) If $i = j$, then $\alpha = \beta \pm k$, where $k \in \mathbb{Z}$, hence, $\alpha = \beta$.

□

Since $E(T_i^\alpha(\boldsymbol{z})) = S_i^\alpha \cdot E(\boldsymbol{z})$ and S_i^α is an orthogonal transformation, the Euler encoding after translation on Z can be considered as an orthogonal transformation of $E(\boldsymbol{z})$. As $\boldsymbol{A}$ can be seen as a linear layer to the decoder in Figure 2, the Euler encoding enables the changes of output from changes of different latent dimensions to be orthogonal. In other words, if $D_{\boldsymbol{\theta}}(E(T_i^\alpha(E_{\boldsymbol{\phi}}(O)))) \neq D_{\boldsymbol{\theta}}(E(T_j^\alpha(E_{\boldsymbol{\phi}}(O))))$, then $i \neq j$. Nevertheless, there are still a number of practical challenges to overcome. These are: **(a) Number of Elements in a Subgroup**: The number of possible elements in the subgroups N_j ($j = 1, \ldots, n$), or at least the relative ratio of the number of elements between the subgroups are not known a priori. This has a crucial role in introducing inductive biases on datasets, **(b) Spatial Distribution of Features**: An ideal factorized latent space must have the features spatially distributed in an equally likely manner. However, the equivariant map we discussed above alone may not be sufficient to address this issue, and **(c) Robustness to Small Perturbations**: Because the proposed approach for disentanglement is deterministic, the model is not resilient to small perturbations (e.g., noise) (Camuto et al., 2021), which is essential for the model to behave in a robust manner when presented with unseen examples. Although it is possible to address some of these concerns from a theoretical stand point, nearly all of these are addressable by carefully designing an architecture that exploits both the AE and the equivariant map principle discussed above, achieving the best possible disentanglement. We discuss these details in Section 3.3 below.

3.3. Architecture of the DAE

In mapping our theory to an architecture, we build on the AE model, which constitutes an encoder, that maps the observation space O to a latent space Z, followed by the disentangling process that factorizes/disentangles the latent space Z to a factorized latent space Z', and finally the decoding layer, that maps the factorized latent Z' to the regenerated observation space O through the Euler encoding. Each of the concerns that were discussed in Section 3.2 (a) through (c) are handled by a network of layers in our architecture. We show this model in Figure 2 and describe how this model addresses the relevant concerns next.

(a) Number of Elements in a Subgroup: Although the number of elements in a subgroup is not known a priori, the number or the relative ratio of the possible number of elements across subgroups can be estimated using techniques that can extract the variance information from compressed information, such as principal component analysis (PCA) (Jolliffe, 2002), independent component analysis (ICA) (Hyvärinen & Oja, 2000), or even through neural networks (Kingma & Welling, 2013; Burgess et al., 2018; Mondal et al., 2021). In this paper, for the reasons of simplification, we will be using the PCA technique. Since the singular values from PCA are proportional to the variances of the principal components of compressed data, these values are used to obtain a relative ratio of the number of possible element in the subgroups (Wall et al., 2003). Then, all singular values are divided by the maximum values, and rounded to one decimal places, and then values smaller than 1 are replaced with hyperparameter α. The relevant algorithm is shown in Algorithm 1 in the Appendix. Let Λ be the relative ratio of the possible number of elements across subgroups obtained from Algorithm 1.

(b) Uniform Spatial Distribution of Features: To ensure that each feature is equally/likely distributed across the latent space and falls within the $(0, 1)$ (which lets $\boldsymbol{A} \cdot E(T_i^\alpha(\boldsymbol{z})) = \boldsymbol{A} \cdot E(T_j^\beta(\boldsymbol{z}))$ if and only if $i = j$ and $\alpha = \beta$ in Theorem 3.3), we introduce a normalization layer, where we apply batch min-max normalization to the outputs of the encoder. As minimum and maximum values vary from batch (mini-batch) to batch (mini-batch), we update the moving minimum and maximum values during the training process, and use them during the test phase, akin to a batch normalization layer (Ioffe & Szegedy, 2015). The minimum and maximum values are also initialized close to the middle point of $[0, 1)$ to facilitate learning. This is then followed by scaling by Λ to account for different number of possible elements for different features.

(c) Robustness to Small Perturbations: Robustness is achieved by introducing an Interpolation layer that performs Gaussian interpolation on the output of the normalized latent space, following Vincent et al. (2010); Berthelot et al.

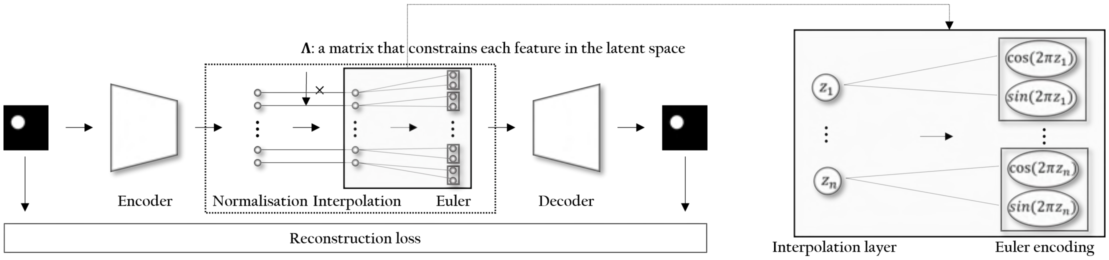


Figure 2. Illustration of the DAE architecture. The model includes the Euler encoding, and the outputs from the interpolation layer are mapped to cosine and sine values. Here Λ is one dimensional vector that constraints each latent dimension with respect to the relative ratio of the possible number of elements across latent space and $\times$ is column-wise product, i.e., $A \times B = [a_1 \cdot B_{:,1}, a_2 \cdot B_{:,2}, \cdots, a_n \cdot B_{:,n}]$ where $A = [a_1, a_2, \cdots, a_n]$, $B = [B_{:,1}, B_{:,2}, \cdots, B_{:,n}]$ and $\cdot$ is scalar multiplication.

(2018). Gaussian interpolation is used to map unseen examples to known examples, and to make the latent space locally smooth. Since the proposed model is deterministic, it is important to map a number of unseen examples to the learned representations. This is achieved by adding a weight-sensitive Gaussian noise to the outputs of the previous layer during training, which is obtained based on the closest proximal distance of each dimension of the representations. The relevant algorithm is shown in Algorithm 2 in the Appendix. It is worth noting that this layer will not be used during the inference phase.

3.4. A Novel Metric for Quantifying Disentanglement: GF-Score

Nearly all of the existing set of metrics outlined in the literature for quantifying linear symmetry-based disentanglement require ground truth labels. Here, we propose a new metric, namely, Grid Fitting Score (GF-Score), to achieve the same purpose without the need for labels. Our hypothesis is that performing independent disentangled actions on a symmetry group causes the corresponding subspace to form a grid-shape latent space. This can be exploited by generating a square grid to include latent variables, and by measuring the mean of the minimum distances from the square grid to latent variables to signify the quality of disentanglement. If the latent variables fit perfectly into the square grid, it would imply that the model achieves perfect linear disentanglement, and we can mark this as zero score. Therefore, the lower the GF-Score is, the better the disentanglement is. The relevant algorithm is shown in Appendix A 3.

4. Evaluation and Results

4.1. Evaluation Method

Our evaluation involves comparing the performance of the proposed approach against twelve baseline models across eight datasets using six disentanglement metrics. We outline these details below.[1]

Datasets: One of the critical challenges around evaluating disentanglement is identifying suitable datasets. It is difficult to identify common datasets to study this problem. In the literature, different datasets have been used for different purposes. For example, dSprite (Matthey et al., 2017), 3D Chair (Burgess & Kim, 2018) and CelebA (Liu et al., 2015) datasets have been used in β-VAE, β-TCVAE, and FVAE. Although these datasets are useful to understand the traversal order of the latent space, they lack a clear underlying group structure. As such, here, we utilize the datasets that have been first utilized in Higgins et al. (2018), with relevant enhancements, which we describe in the Appendix (See A.10). In addition to this dataset, we use six more datasets containing clear underlying group structure, namely, 2D Arrow (Tonnaer et al., 2022), 3D Airplane (Tonnaer et al., 2022), 3D Teapots (Eastwood & Williams, 2018), 3D Shape (Burgess & Kim, 2018), 3D Face Model (Paysan et al., 2009) and Sprites (Reed et al., 2015) datasets. Finally, to demonstrate the performance on complex datasets, we use the Blood Cell dataset (Acevedo et al., 2020), which does not have the ground truth factors but is useful to show the reconstructions of latent traversals with respect to size and shape of blood cells (See the results in the Appendix A.11).

Baseline Models: We considered ten different baselines models for our evaluation, namely, plain AE, vanilla VAE, β-VAE, β-TCVAE, CCI-VAE, FVAE, InfoVAE, DIPVAE, WAE, LSBDVAE, InfoGAN and IB-GAN. For DIPVAE, we only test DIPVAE-I, owing to the reasons of that DIPVAE-II model works better only for cases where the dimension of the latent space is larger than the ground truth factors, which is not the case for us. To render a fair evaluation mechanism, we used the same encoder and decoder architectures, and same latent space dimensions (for each baseline model), which are used in Higgins et al. (2017); Kim & Mnih (2018); Quessard et al. (2020); Tonnaer et al. (2022) throughout the evaluation. Please see the Appendix (See A.8) for additional

[1] Code available at https://github.com/stfc-sciml/dae

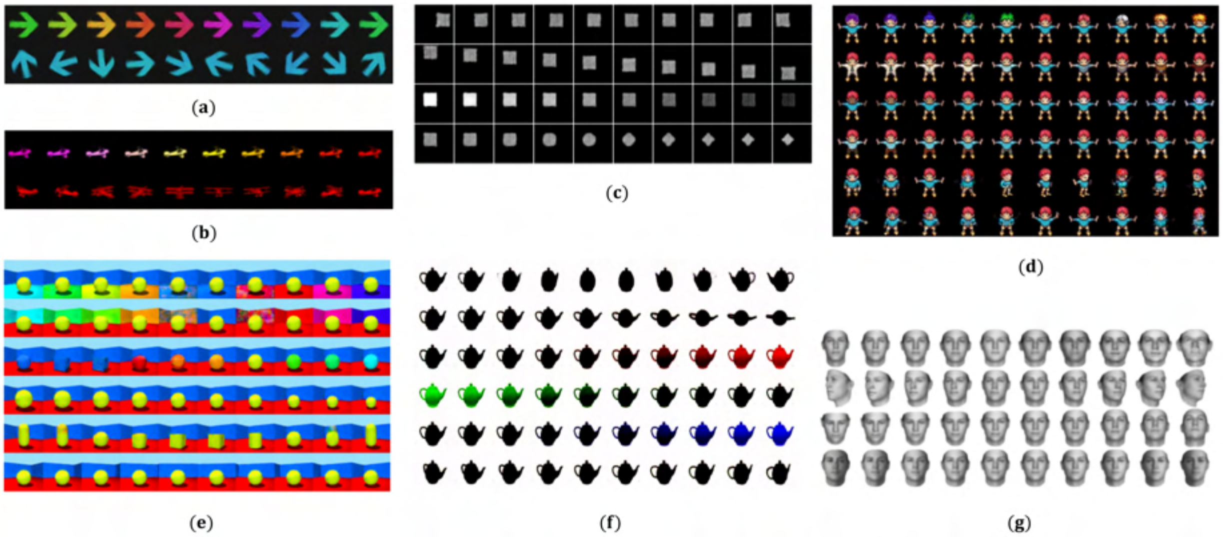


Figure 3. Reconstructions of latent traversals across each latent dimension obtained by the DAE for the (a) 2D Arrow (color and shape), (b) 3D Airplane (color and shape), (c) $XYCS$ (x position, y position, color and shape), (d) Sprites (hair color, top color, body color, bottom color and actions), (e) 3D Shape (floor hue, wall hue, object hue, scale, shape and orientation), (f) 3D Teapots (azimuth, elevation, red, green, blue and extra) and (g) 3D Face Model datasets (face id, azimuth, elevation and lighting).

details.

Performance Metrics: As discussed in Section 2.1, a large number of metrics can be used to study the performance of disentanglement, depending on the nature of the dataset, access to ground truth, availability of latent factors, and the number of dimensions in the latent space. We use two types of metrics: **(a) Visualization of the latent space**, and **(b) Numerical disentanglement score**. The former permits one to visualize the orthogonality between features, and can be used to demonstrate how the latent traversal is achieved by the model and grid structures in the latent space. The second approach provides a quantifiable method for assessing the disentanglement. Collectively, we have used six metrics, including five supervised metrics accounting for each of the disentanglement metric classes (see Section 2.1), namely, **z-diff** and **z-min** from the intervention-based, **dci-rf** from the predictor-based, and **jemmig** and **dcimig** from the information-based metric classes in order to measure disentanglement, completeness and informativeness, along with GF-Score (See 3.4). In Locatello et al. (2019), it was shown that variances of all metrics are large with random seeds and it disturbs the comparison between different models. Hence, we run all the models on each data set for 20 different random seeds and select the random seed with the highest total score over these metrics.

4.2. Results and Discussions

Our evaluation has produced a considerable volume of results, and for the reasons of brevity, we present two sets of results here, namely, (i) we show the reconstructions of latent traversals for the 2D Arrow, 3D Airplane, $XYCS$, 3D shape, 3D Teapots, 3D Face Model and Sprites datasets in Figure 3, and (ii) we present the percentage of changes in disentanglement scores when color and shape features are added to XY dataset in Table 1, along with the disentanglement scores of top two performing models across all datasets for all metrics in Tables 2. We provide the remaining set of results (reconstructions of latent traversals and disentanglement scores of all models), and other relevant details (such as hyper-parameters, and network architectures A.7 and A.8) as part of the Appendix.

4.2.1. Latent Space Visualization

We show the disentangled the latent spaces for the XY, 2D Arrow and 3D Airplane datasets, which have two underlying factors, in Figure 1 (also see Table 17 in Appendix A for details of relevant hyperparameters). As can be seen in Figure 1, the proposed model, in general, provides the ideal grid-shape outlined in Higgins et al. (2018). The plain AE, vanilla VAE, Info-VAE and WAE models offer the worst performance. Other models, such as β-VAE, β-TCVAE, CCI-VAE, and DIPVAE models also come closer to the ideal pattern in the three datasets, and thus most models are able to disentangle x and y positions in XY datasets, and rotation and color factors in 2D Arrow and 3D Airplane datasets. However, the Table 1 shows that when color or shape feature is added to the XY dataset (i.e., for XYC, XYS and $XYCS$ datasets), the disentanglement can become a significant challenge, other than for the proposed

Table 1. Differences in scores between $XY/XYCS$ datasets. Absolute and percentage of change from XY to $XYCS$ are shown. Percentage changes closer to the zero are desirable.

	Disentanglement scores				
Models	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑
DAE	**$^{1.00}/1.00$ (0.0%)**	**$^{1.00}/1.00$ (0.0%)**	**$^{0.99}/0.94$ (-5.0%)**	**$^{0.81}/0.67$ (-17.2%)**	**$^{0.80}/0.75$ (-6.2%)**
AE	$^{1.00}/0.95$ (-5.0%)	$^{1.00}/0.55$ (-45.0%)	$^{0.79}/0.24$ (-69.6%)	$^{0.55}/0.22$ (-60.0%)	$^{0.50}/0.13$ (-74.0%)
VAE	$^{1.00}/0.82$ (-18.0%)	$^{1.00}/0.30$ (-70.0%)	$^{0.84}/0.09$ (-89.2%)	$^{0.58}/0.21$ (-63.7%)	$^{0.54}/0.08$ (-85.1%)
β-VAE	$^{1.00}/0.79$ (-21.0%)	$^{1.00}/0.46$ (-54.0%)	$^{0.91}/0.08$ (-91.2%)	$^{0.65}/0.18$ (-72.3%)	$^{0.63}/0.13$ (-79.3%)
β-TCVAE	$^{1.00}/0.72$ (-28.0%)	$^{1.00}/0.41$ (-59.0%)	$^{0.93}/0.15$ (-83.8%)	$^{0.70}/0.24$ (-65.7%)	$^{0.69}/0.14$ (-79.7%)
CCI-VAE	**$^{1.00}/1.00$ (0.0%)**	**$^{1.00}/1.00$ (0.0%)**	$^{0.98}/0.63$ (-35.0%)	$^{0.78}/0.47$ (-39.7%)	$^{0.76}/0.46$ (-39.4%)
FVAE	**$^{1.00}/1.00$ (0.0%)**	$^{1.00}/0.91$ (-9.0%)	$^{0.96}/0.18$ (-80.8%)	$^{0.73}/0.21$ (-69.5%)	$^{0.70}/0.13$ (-80.5%)
InfoVAE	$^{1.00}/0.92$ (-8.0%)	$^{1.00}/0.54$ (-46.0%)	$^{0.90}/0.21$ (-76.6%)	$^{0.64}/0.27$ (-57.8%)	$^{0.67}/0.13$ (-76.6%)
DIPVAE	**$^{1.00}/1.00$ (0.0%)**	$^{1.00}/0.44$ (-56.0%)	$^{0.98}/0.32$ (-67.3%)	$^{0.78}/0.28$ (-64.1%)	$^{0.78}/0.11$ (-85.8%)
WAE	$^{1.00}/0.93$ (-7.0%)	$^{1.00}/0.38$ (-62.0%)	$^{0.44}/0.06$ (-86.3%)	$^{0.35}/0.15$ (-57.1%)	$^{0.27}/0.2$ (-92.5%)
LSBDVAE	**$^{1.00}/1.00$ (0.0%)**	$^{1.00}/0.76$ (-24.0%)	$^{0.96}/0.38$ (-60.4%)	$^{0.72}/0.30$ (-58.3%)	$^{0.70}/0.28$ (-60.0%)
InfoGAN	$^{1.00}/0.79$ (-21.0%)	$^{1.00}/0.41$ (-59.0%)	$^{0.85}/0.23$ (-72.9%)	$^{0.54}/0.15$ (-72.2%)	$^{0.51}/0.17$ (-66.6%)
IB-GAN	$^{1.00}/0.82$ (-18.0%)	$^{1.00}/0.60$ (-40.0%)	$^{0.53}/0.20$ (-40.0%)	$^{0.35}/0.22$ (-37.1%)	$^{0.31}/0.20$ (-35.4%)

Table 2. Disentanglement scores for the 2D Arrow, 3D Airplane, 3D Teapots, 3D Shape, 3D Face Model and Sprites datasets

Datasets	2D Arrow		3D Airplane		3D Teapots		3D Shape		3D Face Model		Sprites	
Metrics/Models	**DAE**	DIPVAE	**DAE**	DIPVAE	**IB-GAN**	DAE	**FVAE**	DAE	**InfoGAN**	DAE	**DAE**	β-VAE
z-diff ↑	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
z-var ↑	0.85	0.96	1.00	0.96	1.00	1.00	1.00	0.93	0.95	1.00	0.90	0.77
dci-rf ↑	0.88	0.85	0.80	0.54	0.92	0.89	0.99	0.99	0.62	0.65	0.70	0.54
jemmig ↑	0.80	0.75	0.79	0.51	0.60	0.54	0.86	0.87	0.53	0.48	0.59	0.51
dcimig ↑	0.79	0.72	0.75	0.43	0.60	0.53	0.88	0.90	0.54	0.47	0.55	0.43
GF ($\times \frac{1}{100}$) ↓	0.30	2.55	0.19	7.66	0.10	0.002	0.20	0.0009	0.16	0.02	0.005	0.08

model (See Figure 4 in Appendix). Furthermore, the proposed model is the only model that can successfully disentangle the 3D Teapots dataset as shown in Figure 7. In the Sprites dataset, only the proposed model and β-VAE learn hair, top, body and bottom colors while the other models fail to disentangle the features as shown in Figure 10. In addition to these, reconstructions of latent traversals (across each latent dimension) of seven datasets for the DAE are shown in Figure 3 (Also the Appendix A.11 for more results).

4.2.2. Disentanglement Scores

We present the supervised disentanglement scores and their percentage changes when color and shape features are added to the XY dataset in Table 1, with the changes measured relative to the XY dataset, and it is worth noting that the added features have smaller variances than x and y positions. As can be seen, in general, nearly, all models suffer from the performance drop except a few. The CCI-VAE is the only model that performs as good as the proposed model for the **z-diff** and **z-min** metrics. The proposed model shows the smallest drop across three remaining metrics, namely, **dci-rf**, **jemmig** and **dcimig**. While the largest drop in the proposed model is 17.2%, the scores fall by between 39.7% and 92.5% for the other models.

The disentanglement scores for the top two performing models for all datasets (except the 2D Toy dataset) are shown in Tables 2, with the best performing model highlighted in boldface. From these results (including those in the Appendix A.6), the following observations can be drawn: Firstly, the proposed model outperforms all models across all metrics for the 2D Toy, 3D Airplane and Sprites datasets. Secondly, DAE is the only model placed in the top two performing models across all datasets. DAE offers the best score for 2D Arrow and the second best performance for the 3D Teapots, 3D Shape and 3D Face Model dataset. Thirdly, except for the two datasets (3D Teapots and 3D Face Model datasets), DAE shows superior performance compared to the other datasets for the two GAN-based models while the largest drop of DAE compared to two GAN-based models in 3D Teapots and 3D Face Model datasets is 12.9%. Furthermore, training GAN-based models for achieving better or acceptable level of disentanglement was considerably difficult owing to the reason of their instability. The GAN-based models required different learning rates, and different number of epoch for achieving stable and better disentangle per dataset. However, autoencoder-based models did not require such adjustments. Fourthly, for the 2D Toy dataset, the proposed model maintains the reconstruction loss as small as possible whilst offering improved disentanglement scores (See Figures 12-15). On the other hand, the reconstruction losses for the β-VAE, β-TCVAE and CCI-VAE models increase along with their disentanglement scores. Fifthly, the results in Table 14 show that the disentanglement scores drop significantly without the Euler encoding or the normalization layer. Finally, GF-Score shows that the proposed model perfectly fits the latent space into a grid structure across all datasets and baseline models. Based on the GF-Score from Table 4 to 12, a model without regularizer, such as AE, fails to form a grid structure in the latent space.

5. Conclusions

In the context of representation learning, being able to factorize or disentangle the latent space dimensions is crucial for obtaining latent representations that are composed of multiple, independent factors of variations. Literature around disentanglement methods are either predominantly supervised or semi-supervised, and as such, either labels or pairs of images are required or achieved via factorizing the aggregated posterior in the latent space.

In this paper, we presented a non-probabilistic, deterministic model, namely, disentangling autoencoder or DAE, addressing a number of issues found in the literature. Our approach exploits the Euler encoding and the normalization layer that make the subspaces of the latent space independent of one another. Along with the architectural details, we also presented a novel metric for quantifying disentanglement, namely, GF-Score. Our detailed evaluations, performed against a large number of AE-based models, using considerably a large number of metrics show that our model can easily offer superior disentanglement performance when compared against a number of existing methods across a number of datasets.

Although the results are encouraging, a number of aspects remain to investigated, including, evaluation of the proposed model on datasets that lack underlying group structure, understanding the effect of the choice of the latent dimension on the outcomes, and to evaluate different latent space smoothing algorithms, to mention a few.

Acknowledgements

This work was supported by the EPSRC grants EP/T001569/1 (“AI for Science” theme through the Alan Turing Institute), EP/X019918/1 (Blueprinting for AI for Science at Exascale or BASE-II), and by the International Science Partnerships Fund (ISPF) for the AI for Realistic Science (AIRS) programme with the Department of Energy (DOE) laboratories, more specifically with the Argonne and Oak Ridge National Laboratories. We also would like to thank the Scientific Computing Department, Rutherford Appleton Laboratory, Science and Technology Facilities Council for providing the necessary computing resources.

References

Acevedo, A., Merino, A., Alférez, S., Molina, Á., Boldú, L., and Rodellar, J. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in brief*, 30, 2020.

Aneja, J., Schwing, A., Kautz, J., and Vahdat, A. A contrastive learning approach for training variational autoencoder priors. *Advances in Neural Information Processing Systems*, 34, 2021.

Asperti, A. and Trentin, M. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *IEEE Access*, 8:199440–199448, 2020.

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

Berman, N., Naiman, I., and Azencot, O. Multifactor sequential disentanglement via structured koopman autoencoders. *arXiv preprint arXiv:2303.17264*, 2023.

Berthelot, D., Raffel, C., Roy, A., and Goodfellow, I. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018.

Brakel, P. and Bengio, Y. Learning independent features with adversarial nets for non-linear ica. *arXiv preprint arXiv:1710.05050*, 2017.

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

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

Camuto, A., Willetts, M., Roberts, S., Holmes, C., and Rainforth, T. Towards a theoretical understanding of the robustness of variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pp. 3565–3573. PMLR, 2021.

Caselles-Dupré, H., Garcia Ortiz, M., and Filliat, D. Symmetry-based disentangled representation learning requires interaction with environments. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Chen, R. T., Li, X., Grosse, R., and Duvenaud, D. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

Cohen, T. and Welling, M. Learning the irreducible representations of commutative lie groups. In *International Conference on Machine Learning*, pp. 1755–1763. PMLR, 2014.

Do, K. and Tran, T. Theory and evaluation metrics for learning disentangled representations. *arXiv preprint arXiv:1908.09961*, 2019.

Dummit, D. S. and Foote, R. M. *Abstract algebra*, volume 3. Wiley Hoboken, 2004.

Eastwood, C. and Williams, C. K. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2(5):6, 2017.

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

Higgins, I., Chang, L., Langston, V., Hassabis, D., Summerfield, C., Tsao, D., and Botvinick, M. Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. *Nature communications*, 12(1):1–14, 2021.

Hyvärinen, A. and Oja, E. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5): 411–430, 2000.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Iten, R., Metger, T., Wilming, H., Del Rio, L., and Renner, R. Discovering physical concepts with neural networks. *Physical review letters*, 124(1):010508, 2020.

Jolliffe, I. T. *Principal component analysis for special types of data*. Springer, 2002.

Kim, H. and Mnih, A. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018.

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

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

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

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

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

Mondal, A. K., Asnani, H., Singla, P., and Prathosh, A. Flexae: flexibly learning latent priors for wasserstein auto-encoders. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 525–535. PMLR, 27–30 Jul 2021.

Painter, M., Prugel-Bennett, A., and Hare, J. Linear disentangled representations and unsupervised action estimation. *Advances in Neural Information Processing Systems*, 33:13297–13307, 2020.

Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pp. 296–301. Ieee, 2009.

Quessard, R., Barrett, T., and Clements, W. Learning disentangled representations and group structure of dynamical environments. *Advances in Neural Information Processing Systems*, 33:19727–19737, 2020.

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

Rey, L. A. P., Menkovski, V., and Portegies, J. W. Diffusion variational autoencoders. *arXiv preprint arXiv:1901.08991*, 2019.

Ridgeway, K. A survey of inductive biases for factorial representation-learning. *arXiv preprint arXiv:1612.05299*, 2016.

Ridgeway, K. and Mozer, M. C. Learning deep disentangled embeddings with the f-statistic loss. *arXiv preprint arXiv:1802.05312*, 2018.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., and Zhong, C. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022.

Sepliarskaia, A., Kiseleva, J., de Rijke, M., et al. Evaluating disentangled representations. *arXiv preprint arXiv:1910.05587*, 2019.

Spurek, P., Nowak, A., Tabor, J., Maziarka, L., and Jastrzebski, S. Non-linear ica based on cramer-wold metric. In *International Conference on Neural Information Processing*, pp. 294–305. Springer, 2020.

Takahashi, H., Iwata, T., Yamanaka, Y., Yamada, M., and Yagi, S. Variational autoencoder with implicit optimal priors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5066–5073, 2019.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schölkopf, B. Wasserstein auto-encoders. In *International Conference on Learning Representations (ICLR 2018)*. OpenReview. net, 2018.

Tonnaer, L., Rey, L. A. P., Menkovski, V., Holenderski, M., and Portegies, J. Quantifying and learning linear symmetry-based disentanglement. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21584–21608. PMLR, 17–23 Jul 2022.

Tschannen, M., Bachem, O., and Lucic, M. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

Wall, M. E., Rechtsteiner, A., and Rocha, L. M. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pp. 91–109. Springer, 2003.

Wang, T., Yue, Z., Huang, J., Sun, Q., and Zhang, H. Self-supervised learning disentangled group representation as feature. *Advances in Neural Information Processing Systems*, 34:18225–18240, 2021.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. Interpretable transformations with encoder-decoder networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5726–5735, 2017.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. *arXiv preprint arXiv:2110.14795*, 2021.

Zaidi, J., Boilard, J., Gagnon, G., and Carbonneau, M.-A. Measuring disentanglement: A review of metrics. *arXiv preprint arXiv:2012.09276*, 2020.

Zhang, Z., Zhang, R., Li, Z., Bengio, Y., and Paull, L. Perceptual generative autoencoders. In *International Conference on Machine Learning*, pp. 11298–11306. PMLR, 2020.

Zhao, S., Song, J., and Ermon, S. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5885–5892, 2019.

A. Appendix

A.1. A review of group theory

More details of definitions and theorems can be found in Dummit & Foote (2004).

Definition A.1. A group is an ordered pair $(G, \star)$ where G is a set and $\star$ is a binary operation on G satisfying the following axioms:

1. Associativity: $(a \star b) \star c = a \star (b \star c)$, for all $a, b, c \in G$,
2. Identity: there exists an element e in G, called an identity of G, such that for all $a \in G$, $a \star e = e \star a$,
3. Inverses: for each $a \in G$ there is an element a^{-1} of G, called an inverse of a, such that $a \star a^{-1} = a^{-1} \star a = e$.

We shall write the operation $a \star b$ as ab.

Definition A.2. A group action of a group G on a set A is a map $\cdot : G \times A \to A$ by $\cdot(g, a) = g \cdot a$ satisfying the following properties:

1. $g_1 \cdot (g_2 \cdot a) = (g_1 g_2) \cdot a$, for all $g_1, g_2 \in G$, $a \in A$, and
2. $e \cdot a = a$, for all $a \in A$.

Definition A.3. Let G be a group. The subset H of G is a subgroup of G if H is nonempty and $x, y \in H$ implies $x^{-1} \in H$ and $xy \in H$.

Definition A.4. A group H is cylic if H can be generated by a single element, i.e., there is some element $x \in H$ such that $H = \{x^n | n \in Z\}$, and will be denoted by $< x >$.

Definition A.5. Assume that G is a group generated by a subset S and there is some collection of relations, say $R_1, R_2, \cdots, R_m$. Here R_i is an equation in the elements from $S \cup \{e\}$ such that any relation among the elements of S can be deduced from these. Then, we say G has presentation $< S|R >$ and the elements of S are called the generators of $< S|R >$ and the elements of R are called the relators.

A.2. Disentangled representation

The notion of a disentangled representation is mathematically defined using the concept of symmetry in Higgins et al. (2018). For example, horizontal and vertical translations are symmetry transformations in two-dimensional grid, and, hence, such transformations change the location of an object in this two-dimensional grid. From the definitions of a symmetry group in Higgins et al. (2018), a symmetry group can be decomposed as a product of multiple subgroups, if suitable subgroups can be identified. This can render an intuitive method to disentangle the latent space, if subgroups that independently act on subspaces of a latent space, can be found. If actions by transformations of each subgroup only affect the corresponding subspace, the actions are called *disentangled group actions*. In other words, disentangled group actions only change a specific property of the state of an object, and leaves the other properties invariant. If there is a transformation in a vector space of representations, corresponding to a disentangled group action, the representation is called a *disentangled representation*. We reproduce the formal definitions of disentangled group action and disentangled representation from Higgins et al. (2018), as Definitions A.6 and A.7, respectively.

Definition A.6. Suppose that we have a group action $\cdot : G \times X \to X$, and the group G decomposes as a direct product $G = G_1 \times \cdots \times G_n$. Let the action of the full group, and the actions of each subgroups be referred to as $\cdot$ and $\cdot_i$, respectively. Then, the action is *disentangled* if there is a decomposition $X = X_1 \times \cdots \times X_n$, and actions $\cdot_i : G_i \times X_i \to X_i$, $i \in \{1, \cdots, n\}$ such that:

$$(g_1, \cdots, g_n) \cdot (\mathbf{x_1}, \cdots, \mathbf{x_n}) = (g_1 \cdot \mathbf{x_1}, \cdots, g_n \cdot \mathbf{x_n}) \tag{7}$$

for all $g_i \in G_i$ and $\mathbf{x_i} \in X_i$.

Now, to derive the definition of a disentangled representation from the definition of disentangled group action, consider a set of world-states, denoted by W. Furthermore, assume that: (a) there is a generative process $b : W \to O$ leading from world-states to observations, O, (b) and an inference process $h : O \to Z$ leading from observations to an agent's representations, Z. With these, consider the composition $f : W \to Z$, $f = h \circ b$. In terms of transformation, assume that these transformations are represented by a group G of symmetries acting on W via an action $\cdot : G \times W \to W$.

The overarching goal of disentangling the latent space now relies on finding a corresponding action $\cdot : G \times Z \to Z$ so that the symmetry structure of W is reflected in Z. In other words, an action on Z corresponding to the action on W is desirable. This can be achieved if the following condition is satisfied:

$$g \cdot f(\mathbf{w}) = f(g \cdot \mathbf{w}) \quad \forall g \in G, \mathbf{w} \in W. \tag{8}$$

In other words, the action, $\cdot$, should commute with f, which adheres to the definition of the equivariant map, and thus, f is an equivariant map, as shown below.

$$\begin{tikzcd}
G \times W \arrow[r, "\cdot \mathbf{w}"] \arrow[d, "id_G \times f"'] & W \arrow[d, "f"] \\
G \times Z \arrow[r, dashed, "\cdot \mathbf{z}"] & Z
\end{tikzcd}$$

From Higgins et al. (2018), a disentangled representation can be defined as follows:

Definition A.7. The representation Z is disentangled with respect to $G = G_1 \times \cdots \times G_n$ if

1. There is an action $\cdot : G \times Z \to Z$,
2. The map $f : W \to Z$ is equivariant between the actions on W and Z, and
3. There is a decomposition $Z = Z_1 \times \cdots \times Z_n$ or $Z = Z_1 \oplus \cdots \oplus Z_n$ such that each Z_i is fixed by the action of all G_j, $j \neq i$ and affected only by G_i.

A.3. Proof

Proof of Theorem 3.1

Proof. Suppose that there b is an equivariant map defined in Theorem 3.2 and h is an equivariant map. Then

$$\begin{aligned}
g \cdot f(\mathbf{w}) &= g \cdot h(b(\mathbf{w})) \tag{9}\\
&= h(g \cdot b(\mathbf{w})) \tag{10}\\
&= h(b(g \cdot \mathbf{w})) \tag{11}\\
&= f(g \cdot \mathbf{w})) \tag{12}\\
& \tag{13}
\end{aligned}$$

$\forall g \in G, \mathbf{w} \in W$. □

A.4. Comparison of different VAE-based models

In our evaluation, we compare the proposed model against nine other VAE-based derivatives, namely, vanilla VAE, β-VAE, β-TCVAE, CCI-VAE, FVAE, InfoVAE DIPVAE, LSBDVAE and WAE. All these models vary based on the underlying regularizer $L_{reg}(\boldsymbol{\phi})$. For example, the β-VAE model constraints on the latent space using β to limit the capacity of the latent space, which encourages the model to learn the most efficient representation of the data. The regularization term of these different models (Column 2) are summarized in Table 3 along with relevant notes (Column 3).

Table 3. Comparison of different VAE-based models w.r.t the regularizers they employ.

Model	$L_{reg}(\boldsymbol{\phi})$	**Notes**
VAE	$D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\boldsymbol{x})\Vert p(\boldsymbol{z}))$	–
β-VAE	$\beta D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\boldsymbol{x})\Vert p(\boldsymbol{z}))$	Usually, β is greater than 1
β-TCVAE	$I(\boldsymbol{z},\boldsymbol{x}) + \beta D_{\mathrm{KL}}(q(\boldsymbol{z})\Vert \prod_j q(\boldsymbol{z}_j)) + \sum_j D_{\mathrm{KL}}(q(\boldsymbol{z}_j)\Vert p(\boldsymbol{z}_j))$	$I(\cdot,\cdot)$ is a mutual information
CCI-VAE	$\beta\vert D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\boldsymbol{x})\Vert p(\boldsymbol{z})) - C\vert$	C is a capacity
FVAE	$D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\boldsymbol{x})\Vert p(\boldsymbol{z})) + \gamma D_{\mathrm{KL}}(q(\boldsymbol{z})\Vert \prod_j q(\boldsymbol{z}_j)))$	-
InfoVAE	$D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\boldsymbol{x})\Vert p(\boldsymbol{z})) + \lambda MMD(q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\boldsymbol{x}), p(\boldsymbol{z}))$	$MMD(\cdot,\cdot)$ is Maximum Mean Discrepancy
DIPVAE	$D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\boldsymbol{x})\Vert p(\boldsymbol{z})) + \lambda_{od}\sum_{i\neq j}[Cov_{p(\boldsymbol{x})}[\mu_{\boldsymbol{\phi}}(\boldsymbol{x})]]^2_{i,j}$	Cov is a covariance and $\mu_{\boldsymbol{\phi}}(\boldsymbol{x})$ is the output of an encoder.
	$+\lambda_d\sum_i([Cov_{p(\boldsymbol{x})}[\mu_{\boldsymbol{\phi}}(\boldsymbol{x})]]_{i,i} - 1)^2$	We set $\lambda_d = 10\lambda_{od}$ as suggested in Kumar et al. (2017)
WAE	$\lambda MMD(q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\boldsymbol{x}), p(\boldsymbol{z}))$	λ is a regularization coefficient
LSBDVAE	$\Delta D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}\vert\boldsymbol{x})\Vert p(\boldsymbol{z})) + \lambda D_{LSBD}$	ΔD_{KL} is a KL divergence used in Diffusion VAE

A.5. Algorithms

Algorithm 1 Obtaining Λ using PCA

```
  Input: $X$: the entire dataset and $\alpha$: hyperparameter less than $1$
  Output: $\Lambda = [w_1, w_2, \cdots, w_n]$ or $[1, \alpha]$
  if $n > 2$ then
    $S = [s_1, s_2, \cdots, s_n]$: singular values from PCA$(X)$
    $\bar{S} = [\bar{s_1}, \bar{s_2}, \cdots, \bar{s_n}] = S/max(S)$
    $\Lambda = [w_1, w_2, \cdots, w_n]$: round to one decimal place of $\bar{S}$ ($\bar{S}$ of each dataset is shown in Table 15.)
    if there exists $i$ such that $w_i < 1$ then
      $w_i = \alpha$
    end if
  end if
```

Algorithm 2 Interpolation layer

```
  Input: A mini-batch: $B = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ where $\boldsymbol{x}_i \in \mathrm{R}^n$
  Output: $\{\boldsymbol{y}_1, ..., \boldsymbol{y}_m\}$
  Let $\boldsymbol{x}_i = (x_i^k)_{k=1,\cdots,n}$
  for $i = 1$ to $m$ do
    for $k = 1$ to $n$ do
      $w_i^k = min_{j \in \{1, i-1, ..., i+1, m\}} d(x_i^k, x_j^k)$
      $y_i^k = x_i^k + w_i^k * \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, 1)$
    end for
  end for
```

Algorithm 3 Grid fitting score method

```
  Input: $\boldsymbol{Z} = [\boldsymbol{Z}_{:,1}, ..., \boldsymbol{Z}_{:,n}]$: a matrix consisting of all latent variables obtained by a model
  Output: $S$
  for $i = 1$ to $n - 1$ do
    for $j = i$ to $n$ do
      $\boldsymbol{Z}^{i,j} = [\boldsymbol{Z}_{:,i}, \boldsymbol{Z}_{:,j}]$
      $\boldsymbol{G}^{i,j}$: a set of variables from a square grid that fits $\boldsymbol{Z}^{i,j}$
      Let $d^{i,j} = 0$
      for $k = 1$ to len($\boldsymbol{G}^{i,j}$) do
        $d^{i,j} += distance(\boldsymbol{G}^{i,j}_{k,:}, \boldsymbol{Z}^{i,j}_{k,:})$
        $S^{i,j} = d^{i,j}/k$
        $S$: the average of $S^{i,j}$
      end for
    end for
  end for
```

A.6. Disentanglement scores

Table 4. Disentanglement scores for the XY dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times \frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	**1.00**	**0.99**	**0.81**	**0.80**	**0.21**	0.006
AE	**1.00**	**1.00**	0.79	0.55	0.50	8.67	0.003
VAE	**1.00**	**1.00**	0.84	0.58	0.54	8.52	0.002
β-VAE	**1.00**	**1.00**	0.91	0.65	0.63	3.82	0.003
β-TCVAE	**1.00**	**1.00**	0.93	0.70	0.69	4.81	0.004
CCI-VAE	**1.00**	**1.00**	0.98	0.78	0.76	3.25	0.009
FVAE	**1.00**	**1.00**	0.96	0.73	0.70	5.58	0.003
InfoVAE	**1.00**	**1.00**	0.90	0.64	0.60	8.06	0.002
DIPVAE	**1.00**	**1.00**	0.98	0.78	0.78	2.64	0.003
WAE	**1.00**	**1.00**	0.44	0.35	0.27	5.82	0.003
LSBDVAE	**1.00**	**1.00**	0.96	0.72	0.70	5.47	0.003
InfoGAN	**1.00**	**1.00**	0.85	0.54	0.51	2.01	–
IB-GAN	**1.00**	**1.00**	0.53	0.35	0.31	5.46	–

Table 5. Disentanglement scores for the XYC dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times \frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	**1.00**	**0.99**	**0.91**	**0.91**	**0.01**	0.001
AE	**1.00**	0.82	0.28	0.38	0.22	2.53	0.001
VAE	**1.00**	0.68	0.24	0.29	0.21	1.37	0.001
β-VAE	**1.00**	**1.00**	0.90	0.72	0.70	0.64	0.001
β-TCVAE	**1.00**	**1.00**	0.95	0.81	0.81	0.82	0.003
CCI-VAE	**1.00**	**1.00**	0.95	0.67	0.64	1.04	0.002
FVAE	**1.00**	0.80	0.36	0.35	0.31	0.20	0.001
InfoVAE	**1.00**	0.78	0.31	0.33	0.25	0.97	0.001
DIPVAE	**1.00**	0.92	0.37	0.36	0.30	0.54	0.001
WAE	**1.00**	0.69	0.30	0.32	0.18	0.93	0.002
LSBDVAE	**1.00**	0.64	0.30	0.33	0.25	1.09	0.001
InfoGAN	**1.00**	0.53	0.36	0.27	0.27	0.37	–
IB-GAN	**1.00**	0.78	0.18	0.18	0.11	0.71	–

Table 6. Disentanglement scores for the XYS dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times \frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	**1.00**	**0.99**	**0.86**	**0.87**	**0.007**	0.005
AE	**1.00**	0.95	0.47	0.43	0.34	2.92	0.001
VAE	**1.00**	0.80	0.34	0.39	0.30	2.51	0.002
β-VAE	**1.00**	**1.00**	0.97	0.75	0.73	1.14	0.005
β-TCVAE	**1.00**	**1.00**	**0.99**	0.84	0.82	0.77	0.007
CCI-VAE	**1.00**	**1.00**	0.97	0.71	0.69	0.71	0.009
FVAE	**1.00**	0.83	0.42	0.41	0.37	2.02	0.001
InfoVAE	**1.00**	0.94	0.38	0.40	0.34	1.38	0.001
DIPVAE	**1.00**	0.82	0.35	0.37	0.30	0.92	0.001
WAE	**1.00**	0.80	0.31	0.34	0.23	1.26	0.008
LSBDVAE	**1.00**	0.71	0.34	0.41	0.35	1.96	0.002
InfoGAN	**1.00**	0.48	0.50	0.33	0.33	0.38	–
IB-GAN	0.99	0.19	0.05	0.19	0.06	0.87	–

Table 7. Disentanglement scores for the $XYCS$ dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times \frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	**1.00**	**0.94**	**0.67**	**0.75**	**0.01**	0.002
AE	0.95	0.55	0.24	0.22	0.13	18.37	0.001
VAE	0.82	0.30	0.09	0.21	0.08	0.58	0.001
β-VAE	0.79	0.46	0.08	0.18	0.13	0.44	0.001
β-TCVAE	0.72	0.41	0.15	0.24	0.14	0.61	0.002
CCI-VAE	**1.00**	**1.00**	0.63	0.47	0.46	0.25	0.002
FVAE	**1.00**	0.91	0.18	0.21	0.13	0.10	0.004
InfoVAE	0.92	0.54	0.21	0.27	0.14	0.44	0.001
DIPVAE	**1.00**	0.44	0.32	0.28	0.11	0.26	0.001
WAE	0.93	0.38	0.06	0.15	0.05	0.42	0.003
LSBDVAE	**1.00**	0.76	0.38	0.30	0.28	0.54	0.001
InfoGAN	0.79	0.41	0.23	0.15	0.17	0.17	–
IB-GAN	0.82	0.60	0.20	0.22	0.20	0.37	–

Table 8. Disentanglement scores for the 2D Arrow dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times \frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	0.85	**0.88**	**0.80**	**0.79**	**0.30**	0.014
AE	**1.00**	0.30	0.00	0.29	0.17	4480.97	0.012
VAE	0.99	0.85	0.01	0.19	0.05	16.36	0.059
β-VAE	**1.00**	0.88	0.34	0.62	0.55	7.71	0.034
β-TCVAE	**1.00**	0.87	0.83	0.77	0.73	5.48	0.011
CCI-VAE	**1.00**	0.84	0.79	0.75	0.70	3.07	0.017
FVAE	**1.00**	0.90	0.46	0.58	0.52	17.09	0.022
InfoVAE	**1.00**	0.90	0.05	0.33	0.22	16.75	0.029
DIPVAE	**1.00**	0.96	0.85	0.75	0.72	2.55	0.010
WAE	**1.00**	0.62	0.07	0.24	0.10	1510.34	0.006
LSBDVAE	**1.00**	0.83	0.83	**0.80**	0.76	4.07	0.008
InfoGAN	**1.00**	**1.00**	0.04	0.40	0.35	2.39	–
IB-GAN	**1.00**	**1.00**	0.17	0.23	0.11	5.74	–

Table 9. Disentanglement scores for the 3D Airplane dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times \frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	**1.00**	**0.80**	**0.79**	**0.75**	**0.28**	0.011
AE	**1.00**	0.38	0.00	0.22	0.14	2505.92	0.003
VAE	**1.00**	**1.00**	0.01	0.36	0.28	14.96	0.013
β-VAE	**1.00**	**1.00**	0.09	0.44	0.35	7.92	0.015
β-TCVAE	**1.00**	0.96	0.28	0.52	0.44	15.11	0.009
CCI-VAE	**1.00**	0.95	0.13	0.49	0.41	5.60	0.013
FVAE	**1.00**	**1.00**	0.01	0.39	0.29	14.06	0.011
InfoVAE	**1.00**	0.98	0.17	0.48	0.37	11.86	0.014
DIPVAE	**1.00**	0.96	0.54	0.51	0.43	7.66	0.004
WAE	**1.00**	0.82	0.06	0.32	0.17	1392.81	0.002
LSBDVAE	**1.00**	0.94	0.18	0.49	0.41	11.90	0.002
InfoGAN	**1.00**	**1.00**	0.16	0.46	0.40	2.19	–
IB-GAN	**1.00**	0.89	0.21	0.35	0.24	7.28	–

Table 10. Disentanglement scores for the 3D Teapots dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times\frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	**1.00**	0.89	0.54	0.53	**0.0031**	0.002
AE	0.89	0.45	0.16	0.16	0.05	23.22	0.001
VAE	**1.00**	0.77	0.43	0.29	0.20	0.24	0.001
β-VAE	0.93	0.80	0.47	0.29	0.20	0.14	0.001
β-TCVAE	**1.00**	0.83	0.67	0.35	0.30	0.16	0.001
CCI-VAE	0.91	0.65	0.42	0.35	0.16	0.02	0.003
FVAE	**1.00**	0.79	0.50	0.32	0.25	0.17	0.001
InfoVAE	**1.00**	0.73	0.47	0.31	0.23	0.15	0.001
DIPVAE	**1.00**	**1.00**	0.84	0.52	0.50	0.13	0.001
WAE	0.78	0.52	0.15	0.15	0.04	0.20	0.001
LSBDVAE	**1.00**	0.68	0.51	0.33	0.25	0.15	0.001
InfoGAN	**1.00**	**1.00**	0.74	0.50	0.51	0.07	–
IB-GAN	**1.00**	**1.00**	**0.92**	**0.60**	**0.60**	0.10	–

Table 11. Disentanglement scores for the 3D Shape dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times\frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	0.93	**0.99**	**0.87**	**0.90**	**0.0013**	0.0008
AE	0.95	0.70	0.26	0.19	0.13	0.33	0.0006
VAE	0.96	0.81	0.31	0.27	0.22	0.14	0.0008
β-VAE	**1.00**	0.94	0.93	0.80	0.82	0.04	0.0022
β-TCVAE	0.89	0.75	0.79	0.77	0.74	0.02	0.0012
CCI-VAE	0.96	0.94	0.72	0.56	0.59	0.08	0.0008
FVAE	**1.00**	**1.00**	**0.99**	0.86	0.88	0.04	0.0007
InfoVAE	0.90	0.80	0.35	0.22	0.16	0.20	0.0007
DIPVAE	0.93	0.79	0.68	0.49	0.52	0.20	0.0007
WAE	0.74	0.45	0.09	0.17	0.08	0.14	0.0017
LSBDVAE	**1.00**	0.92	0.51	0.42	0.40	0.15	0.0006
InfoGAN	**1.00**	0.80	0.80	0.59	0.66	0.12	–
IB-GAN	0.62	0.32	0.02	0.18	0.03	0.04	–

Table 12. Disentanglement scores for the 3D Face Model dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times\frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	**1.00**	**0.65**	0.48	0.47	**0.02**	0.004
AE	0.69	0.47	0.10	0.19	0.05	100.60	0.003
VAE	**1.00**	0.69	0.47	0.34	0.20	0.47	0.002
β-VAE	**1.00**	0.92	0.60	0.52	0.44	0.34	0.005
β-TCVAE	**1.00**	0.92	0.59	0.53	0.47	0.37	0.005
CCI-VAE	**1.00**	0.90	0.61	0.53	0.44	0.31	0.002
FVAE	**1.00**	0.64	0.49	0.35	0.22	0.39	0.003
InfoVAE	**1.00**	0.90	0.57	0.50	0.38	0.44	0.002
DIPVAE	**1.00**	0.69	0.54	0.36	0.29	0.37	0.003
WAE	**1.00**	0.78	0.20	0.20	0.13	0.39	0.006
LSBDVAE	**1.00**	0.88	0.61	**0.54**	0.43	0.46	0.002
InfoGAN	**1.00**	0.95	0.62	0.53	**0.54**	0.16	–
IB-GAN	**1.00**	0.82	0.52	0.41	0.35	0.31	–

Table 13. Disentanglement scores for the Sprites dataset

Models/Metrics	z-diff ↑	z-var ↑	dci-rf ↑	jemmig ↑	dcimig ↑	GF ($\times\frac{1}{100}$) ↓	MSE ↓
DAE	**1.00**	**0.90**	**0.70**	**0.59**	**0.55**	**0.005**	0.010
AE	0.86	0.57	0.11	0.15	0.10	30.98	0.008
VAE	0.79	0.47	0.18	0.20	0.13	0.18	0.008
β-VAE	0.99	0.77	0.54	0.51	0.43	0.08	0.011
β-TCVAE	0.98	0.75	0.53	0.56	0.42	0.02	0.013
CCI-VAE	0.95	0.71	0.35	0.33	0.28	1.09	0.012
FVAE	0.93	0.64	0.48	0.40	0.37	0.05	0.008
InfoVAE	0.85	0.55	0.34	0.27	0.25	0.20	0.008
DIPVAE	0.81	0.53	0.46	0.35	0.33	0.19	0.007
WAE	0.68	0.35	0.08	0.11	0.04	0.17	0.007
LSBDVAE	0.85	0.57	0.36	0.31	0.31	0.22	0.007
InfoGAN	0.73	0.33	0.07	0.17	0.07	0.11	–
IB-GAN	0.67	0.38	0.09	0.13	0.07	0.11	–

Table 14. Performance effects when removing the Euler encoding (E) or the normalization layer (N)

Datasets	XY			XYC			XYS			$XYCS$		
Metrics/Models	DAE	w/o E	w/o N	DAE	w/o E	w/o N	DAE	w/o E	w/o N	DAE	w/o E	w/o N
z-diff ↑	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.82
z-var ↑	1.00	1.00	1.00	1.00	0.68	0.58	1.00	0.80	0.36	1.00	0.82	0.17
dci-rf ↑	0.99	0.59	0.95	0.99	0.44	0.26	0.99	0.54	0.11	0.96	0.57	0.07
jemmig ↑	0.81	0.38	0.71	0.91	0.50	0.27	0.86	0.48	0.20	0.81	0.47	0.15
dcimig ↑	0.80	0.33	0.70	0.91	0.63	0.27	0.87	0.61	0.19	0.83	0.59	0.06
GF ($\times\frac{1}{100}$) ↓	0.21	1.00	0.86	0.01	0.07	0.48	0.07	0.21	2.07	0.01	0.21	0.59

Datasets	2D Arrow			3D Airplnae			3D Teapots		
Metrics/Models	DAE	w/o E	w/o N	DAE	w/o E	w/o N	DAE	w/o E	w/o N
z-diff ↑	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00
z-var ↑	0.85	0.65	0.57	1.00	1.00	0.40	1.00	0.81	0.92
dci-rf ↑	0.88	0.04	0.00	0.80	0.00	0.00	0.89	0.43	0.62
jemmig ↑	0.80	0.19	0.25	0.79	0.35	0.13	0.54	0.22	0.40
dcimig ↑	0.79	0.53	0.19	0.75	0.27	0.12	0.53	0.16	0.31
GF ($\times\frac{1}{100}$) ↓	0.30	6.73	1213204.08	0.28	1.97	143594.36	0.0031	0.16	0.32

Datasets	3D Shape			3D Face Model			Sprites		
Metrics/Models	DAE	w/o E	w/o N	DAE	w/o E	w/o N	DAE	w/o E	w/o N
z-diff ↑	1.00	0.99	1.00	1.00	0.87	0.90	1.00	0.92	0.85
z-var ↑	0.93	0.81	0.92	1.00	0.70	0.74	0.90	0.56	0.57
dci-rf ↑	0.99	0.43	0.62	0.65	0.65	0.62	0.70	0.22	0.46
jemmig ↑	0.87	0.22	0.40	0.48	0.53	0.50	0.59	0.25	0.41
dcimig ↑	0.90	0.16	0.31	0.47	0.61	0.47	0.55	0.14	0.29
GF ($\times\frac{1}{100}$) ↓	0.0013	0.16	0.32	0.02	0.47	0.44	0.005	0.01	0.12

A.7. Hyperparameters

Table 15. $\bar{S}$ values for different datasets

Dataset	$\bar{S}$
XY	[1.0, 1.0]
XYC	[1.0, 1.0, 0.8]
XYS	[1.0, 1.0, 0.8]
$XYCS$	[1.0, 1.0, 0.8, 0.8]
2D Arrow	[1.0, 1.0]
3D Airplane	[1.0, 1.0]
3D Teapots	[1.0, 0.8, 0.8, 0.4, 0.3, 0.3]
3D Shape	[1.0, 1.0, 1.0, 1.0, 0.5, 0.5]
3D Face Model	[1.0, 0.4, 0.4, 0.3]
Sprites	[1.0, 0.8, 0.7, 0.7, 0.7, 0.6]
Blood Cell	[1.0, 0.6, 0.4]

Table 16. All hyperparameters for models.

Model	Values for $XYCS$ dataset	Extra values for the other datasets
DAE (α)	$[1.0, 0.5, 0.1, 0.05, 0.01]$	–
β-VAE (β)	$[2, 4, 8, 16, 32, 64, 128]$	–
β-TCVAE (β)	$[2, 4, 8, 16, 32, 64, 128]$	–
CCI-VAE (C)	$[100, 500, 1000]$	$[50.0]$
FVAE (γ)	$[1, 5, 10, 20, 30, 40, 50]$	–
InfoVAE (λ)	$[[100, 500, 1000]$	$[50.0, 2000.0]$
DIPVAE (λ)	$[1, 2, 5, 10, 20, 50, 100]$	–
WAE (λ)	$[1, 5, 10, 20, 30, 40, 50]$	–
LSBD (λ)	$[1, 10, 100]$	–

Table 17. Best hyperparameters for models for the 2D datasets.

Model / Dataset	XY	XYC	XYS	$XYCS$	2D Arrow	Sprites	Blood Cell
DAE (α)	1.0	0.001	0.001	0.01	0.5	0.05	0.005
β-VAE (β)	16	32	64	16	16	16	2
β-TCVAE (β)	32	128	128	32	8	16	16
CCI-VAE (C)	500	100	100	100	50	50	20
FVAE (γ)	1	40	1	500	40	10	5
InfoVAE (λ)	100	20	1	500	50	500	50
DIPVAE (λ)	50	100	2	100	20	5	2
WAE (λ)	1	20	40	50	10	20	1
LSBD (λ)	100	1	1	1	10	1	10

Table 18. Best hyperparameters for models for the 3D datasets.

Model / Dataset	3D Airplane	3D Teapots	3D Shape	3D Face Model
DAE (α)	0.05	0.01	0.01	0.05
β-VAE (β)	16	6	64	16
β-TCVAE (β)	16	6	32	32
CCI-VAE (C)	500	50	100	100
FVAE (γ)	20	1	5	1
InfoVAE (λ)	50	50	100	2000
DIPVAE (λ)	100	20	1	2
WAE (λ)	5	10	50	50
LSBD (λ)	10	100	10	10

A.8. Encoder and Decoder architectures

Table 19. Architecture for the 2D Toy Dataset

Encoder	Decoder
Input 84×84×1 image	3×3 1 Conv ↓, Sigmoid
10×10 8 Conv ↓, BN, LReLU	10×10 1 Conv ↑, BN, LReLU
10×10 16 Conv ↓, BN, LReLU	10×10 8 Conv ↑, BN, LReLU
FC 64	FC 256, LReLU
FC The number of features	FC 64, LReLU

Table 20. Architecture for the 2D Arrow and 3D Airplane Datasets

Encoder	Decoder
Input 64×64×3 image	3×3 1 Conv ↓
4×4 32 Conv ↓, BN, LReLU	4×4 1 Conv ↑, BN, LReLU
4×4 32 Conv ↓, BN, LReLU	4×4 32 Conv ↑, BN, LReLU
4×4 64 Conv ↓, BN, LReLU	4×4 32 Conv ↑, BN, LReLU
4×4 64 Conv ↓, BN, LReLU	4×4 64 Conv ↑, BN, LReLU
FC 256	FC 1024, LReLU
FC 2	FC 256, LReLU

Table 21. Architecture for the 3D Teapots and Sprites Dataset

Encoder	Decoder
Input 64×64×3 image	3×3 1 Conv ↓
4×4 32 Conv ↓, BN, ReLU	4×4 3 Conv ↑, BN, ReLU
4×4 32 Conv ↓, BN, ReLU	4×4 32 Conv ↑, BN, ReLU
4×4 64 Conv ↓, BN, ReLU	4×4 32 Conv ↑, BN, ReLU
4×4 64 Conv ↓, BN, ReLU	4×4 64 Conv ↑, BN, ReLU
FC 128	FC 1024, LReLU
FC 6	FC 128, LReLU

Table 22. Architecture for the 3D Shape Dataset

Encoder	Decoder
Input 64×64×3 image	3×3 1 Conv ↓, Sigmoid
4×4 32 Conv ↓, BN, LReLU	4×4 3 Conv ↑, BN, LReLU
4×4 32 Conv ↓, BN, LReLU	4×4 32 Conv ↑, BN, LReLU
4×4 64 Conv ↓, BN, LReLU	4×4 32 Conv ↑, BN, LReLU
4×4 64 Conv ↓, BN, LReLU	4×4 64 Conv ↑, BN, LReLU
FC 256	FC 1024, LReLU
FC 6	FC 256, LReLU

Table 23. Architecture for the 3D Face Model Dataset

Encoder	Decoder
Input 64×64×1 image	3×3 1 Conv ↓, Sigmoid
4×4 32 Conv ↓, BN, LReLU	4×4 1 Conv ↑, BN, LReLU
4×4 32 Conv ↓, BN, LReLU	4×4 32 Conv ↑, BN, LReLU
4×4 64 Conv ↓, BN, LReLU	4×4 32 Conv ↑, BN, LReLU
4×4 64 Conv ↓, BN, LReLU	4×4 64 Conv ↑, BN, LReLU
FC 128	FC 1024, LReLU
FC 4	FC 128, LReLU

A.9. System and Model Configurations

All of our experiments were run on a single hardware consisting two DGX2 nodes, collectively consisting of 32-V100 GPUs, 1.5GB GPU RAM, and 3TB System RAM. Encoder and decoder architecture are the same in all experiments. Encoder has two convolutional layers followed by Batch Normalization layer and LeakyReLU activation. After convolutional layers, there is one fully-connected layer with 64 nodes and another layer which maps to the latent space. The decode part is symmetric to the encoder part. C for CCI-VAE is set as 25 for 2D Toy dataset and as 50 for all other datasets.

A.10. Dataset

1. 2D Toy Dataset: This dataset has objects with three shapes (S) (a circles, a rectangles and a diamonds), and variations to their x and y positions and color information (more specifically, the brightness). This is a rather small, but very effective, dataset. There are 53 unique x positions (X), 53 unique y positions (Y) and 5 colors (C). We create XY, XYC, XYS and $XYCS$ sub-datasets to show the differences of the latent space when the combination of categorical and continuous factors are presented.

2. 2D Arrow Dataset (Tonnaer et al., 2022): This dataset has 4096, three-channel RGB, $64 \times 64 \times 3$ images of a 2D object (arrow) with ground truth factors of 64 colors and 64 fixed in-plane rotations.

3. 3D Airplane Dataset (Tonnaer et al., 2022): This dataset also has 4096, three-channel RGB, $64 \times 64 \times 3$ images of a 3D object (airplane within the ModelNet40 dataset (Wu et al., 2015)) with ground truth factors of 64 colors and 64 rotations with respect to a vertical axis.

4. 3D Teapots Dataset (Eastwood & Williams, 2018): This dataset has $200,000$, three-channel RGB, $64 \times 64 \times 3$ images of a 3D object (teapot) with ground truth factors of independently sampled from its respective distribution: azimuth $\sim U[0, 2\pi]$, elevation $\sim U[0, \pi/2]$, and three colors, namely, red (R), green (G) and blue (B), sampled with $R \sim U[0, 1]$, $G \sim U[0, 1]$, and $B \sim U[0, 1]$. This dataset is very effective to evaluate model when all factors are independently from the uniform distributions.

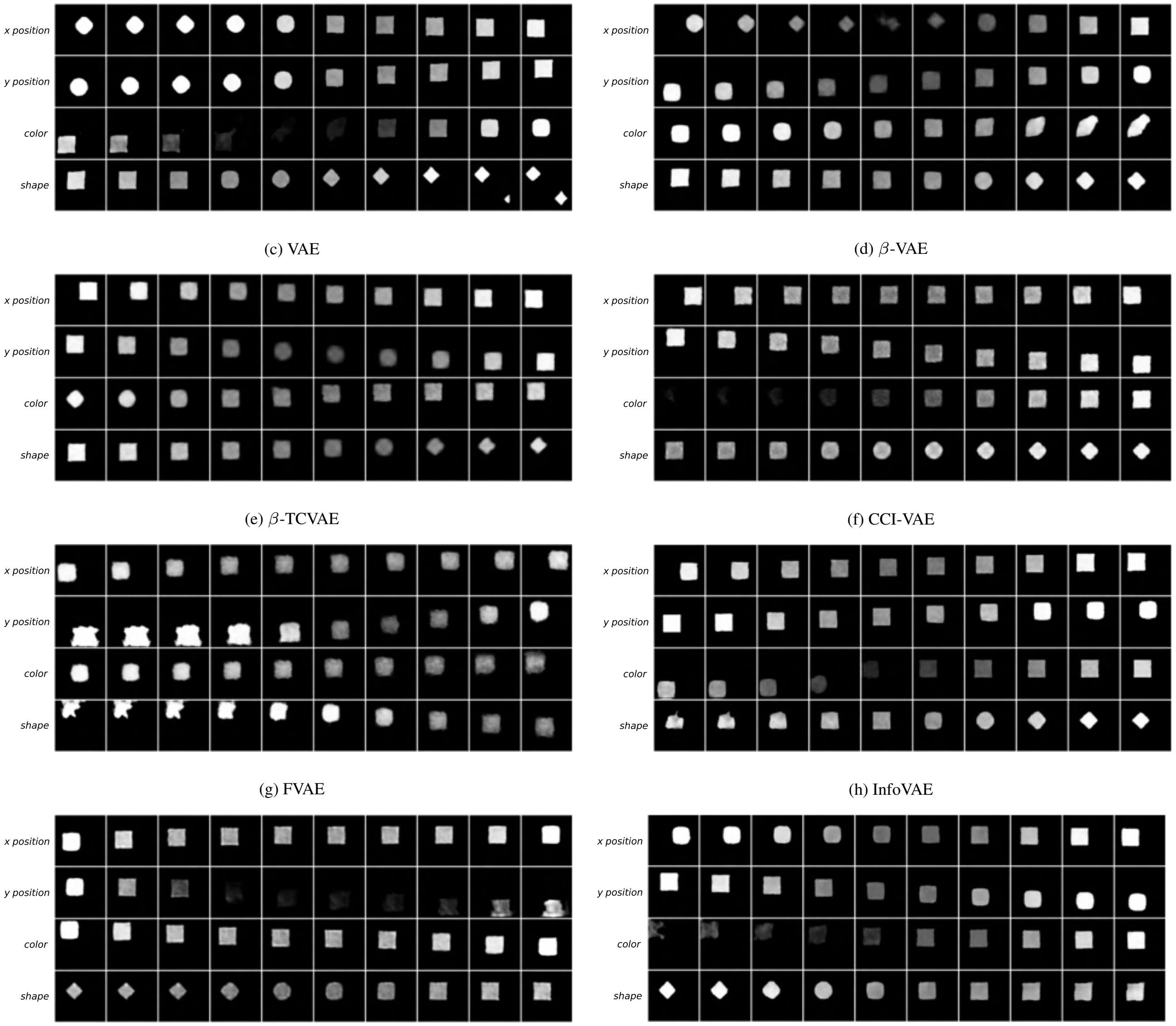
5. 3D Shape Dataset (Burgess & Kim, 2018): This dataset has $480,000$, three-channel RGB, $64 \times 64 \times 3$ images of 3D objects with ground truth factors of four shapes, eight scales, 15 orientations, 10 floor color, 10 wall colors, and 10 object colors.

6. 3D Face Model Dataset (Paysan et al., 2009): This dataset has $127,050$, gray-scale, 64×64 images of 3D faces with ground truth factors of 50 different face ids, 21 azimuth, 11 elevation and 11 lighting conditions.

7. Sprites Dataset (Reed et al., 2015): This dataset has $93,312$, $64 \times 64 \times 3$ images from 1296 unique characters with ground truth factors of the colors of hair, tops, body, bottom, directions and actions.

8. Blood Cell Dataset (Acevedo et al., 2020): This dataset has $17,092$, $360 \times 364 \times 3$ images of eight types of normal peripheral blood cells. We used the $28 \times 28 \times 3$ resized dataset followed by Yang et al. (2021).

A.11. Reconstructions of latent traversals across each latent dimension



(c) VAE

(d) β-VAE

(e) β-TCVAE

(f) CCI-VAE

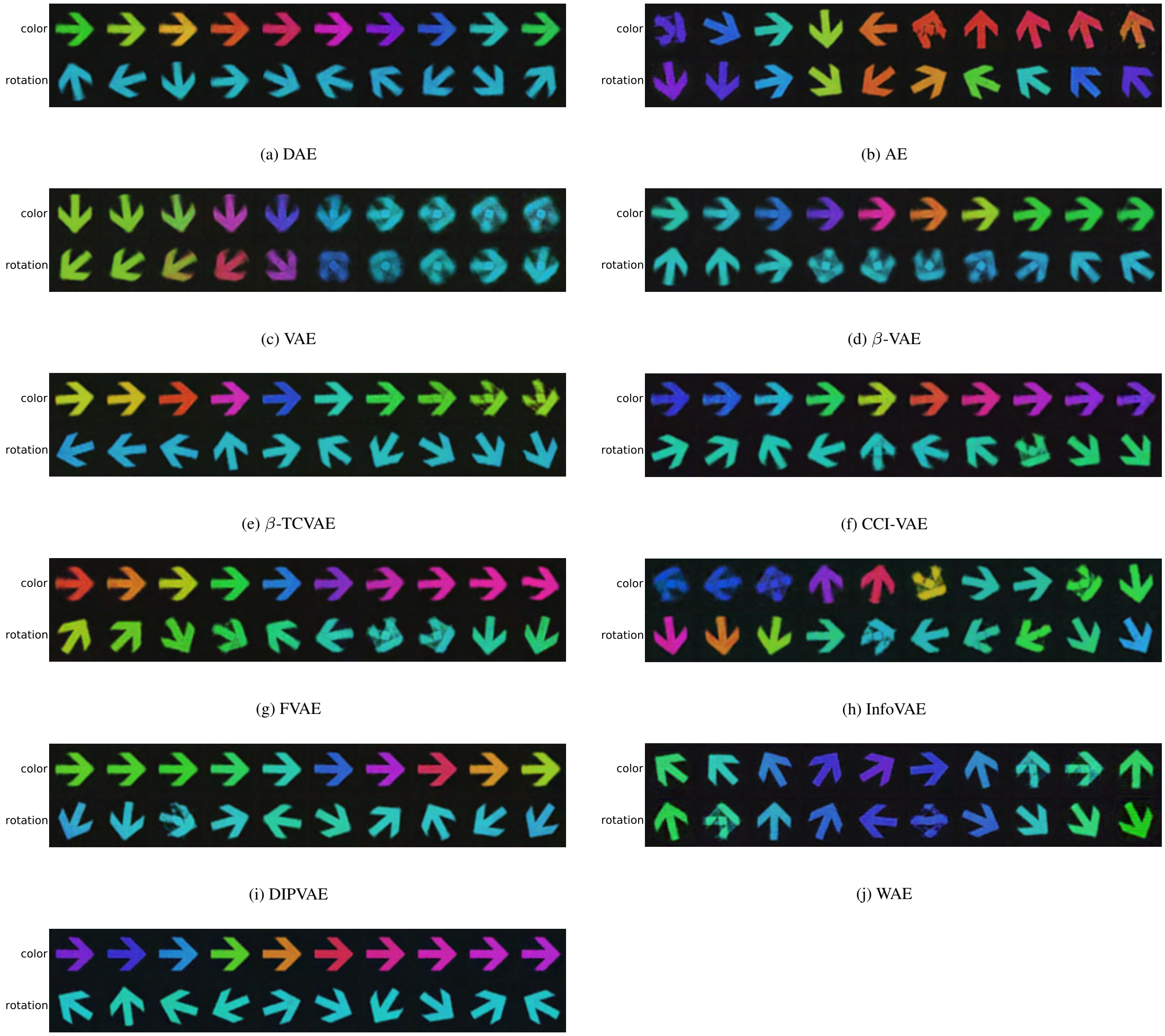
(g) FVAE

(h) InfoVAE

(i) DIPVAE

(j) LSBDVAE

Figure 4. Reconstructions of latent traversals across each latent dimension in the $XYCS$ dataset.



(a) DAE

(b) AE

(c) VAE

(d) β-VAE

(e) β-TCVAE

(f) CCI-VAE

(g) FVAE

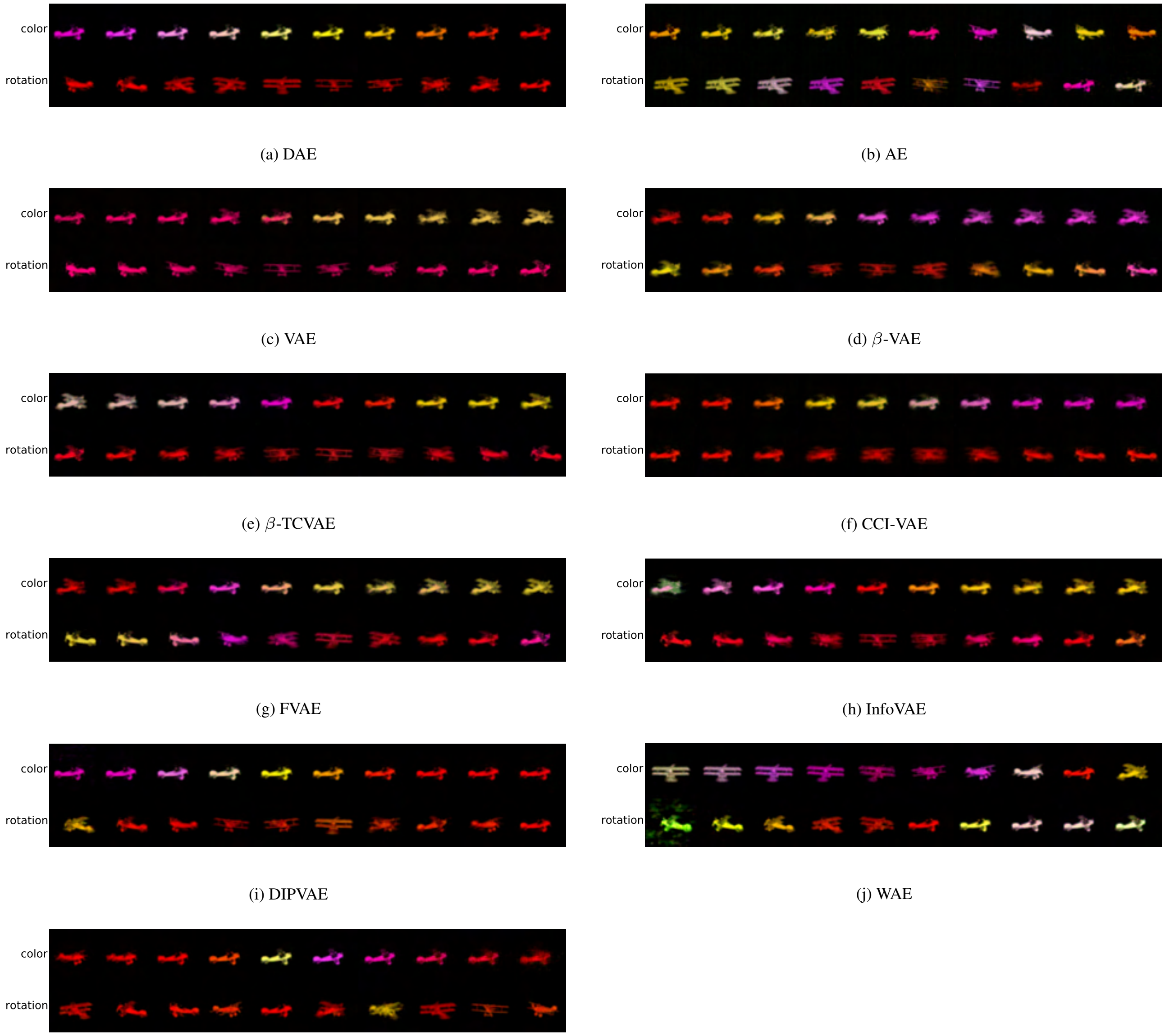
(h) InfoVAE

(i) DIPVAE

(j) WAE

(k) LSBDVAE

Figure 5. Reconstructions of latent traversals across each latent dimension in the 2D Arrow dataset.



(a) DAE

(b) AE

(c) VAE

(d) β-VAE

(e) β-TCVAE

(f) CCI-VAE

(g) FVAE

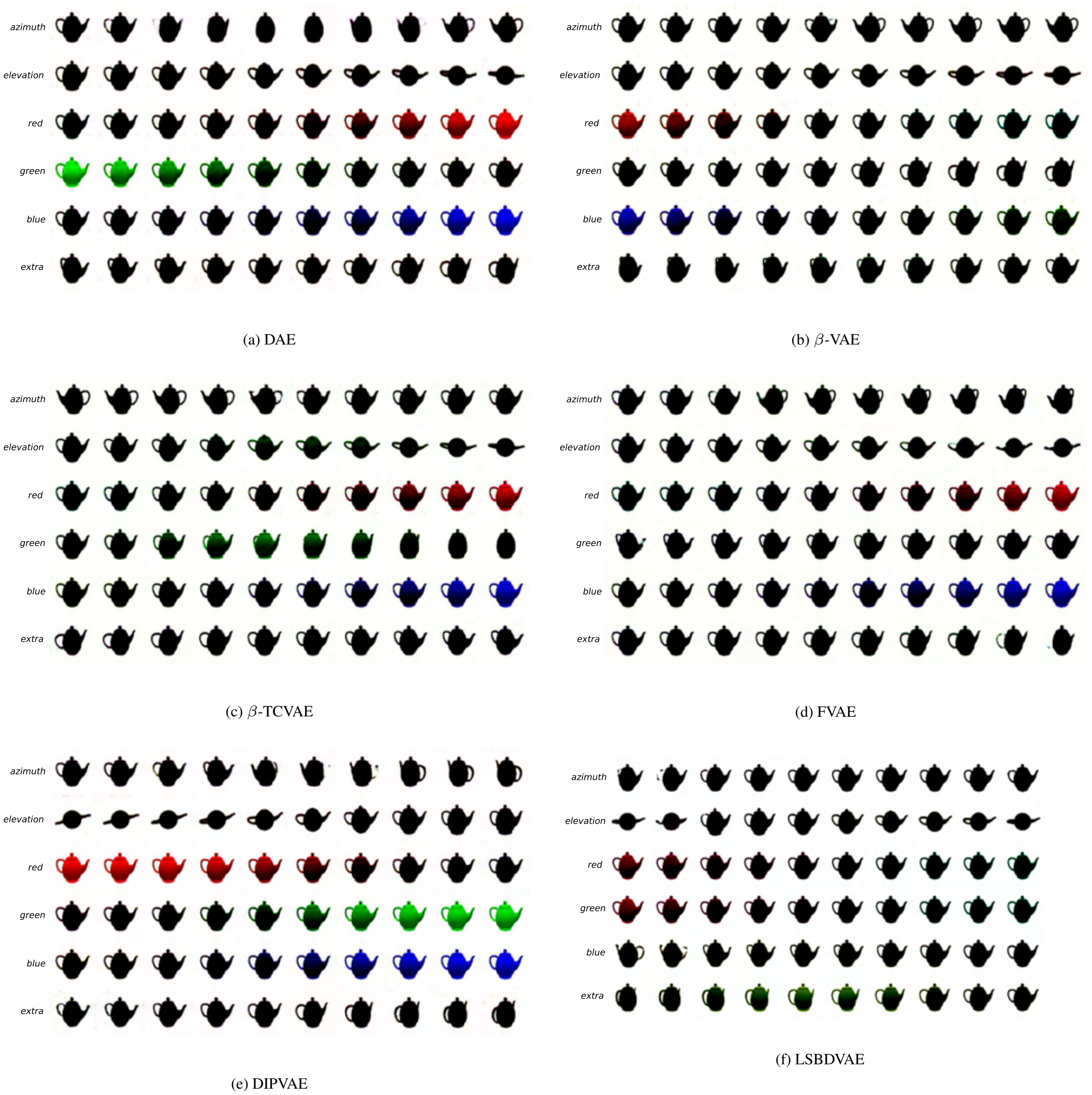
(h) InfoVAE

(i) DIPVAE

(j) WAE

(k) LSBDVAE

Figure 6. Reconstructions of latent traversals across each latent dimension in the 3D Airplane dataset.



(a) DAE

(b) β-VAE

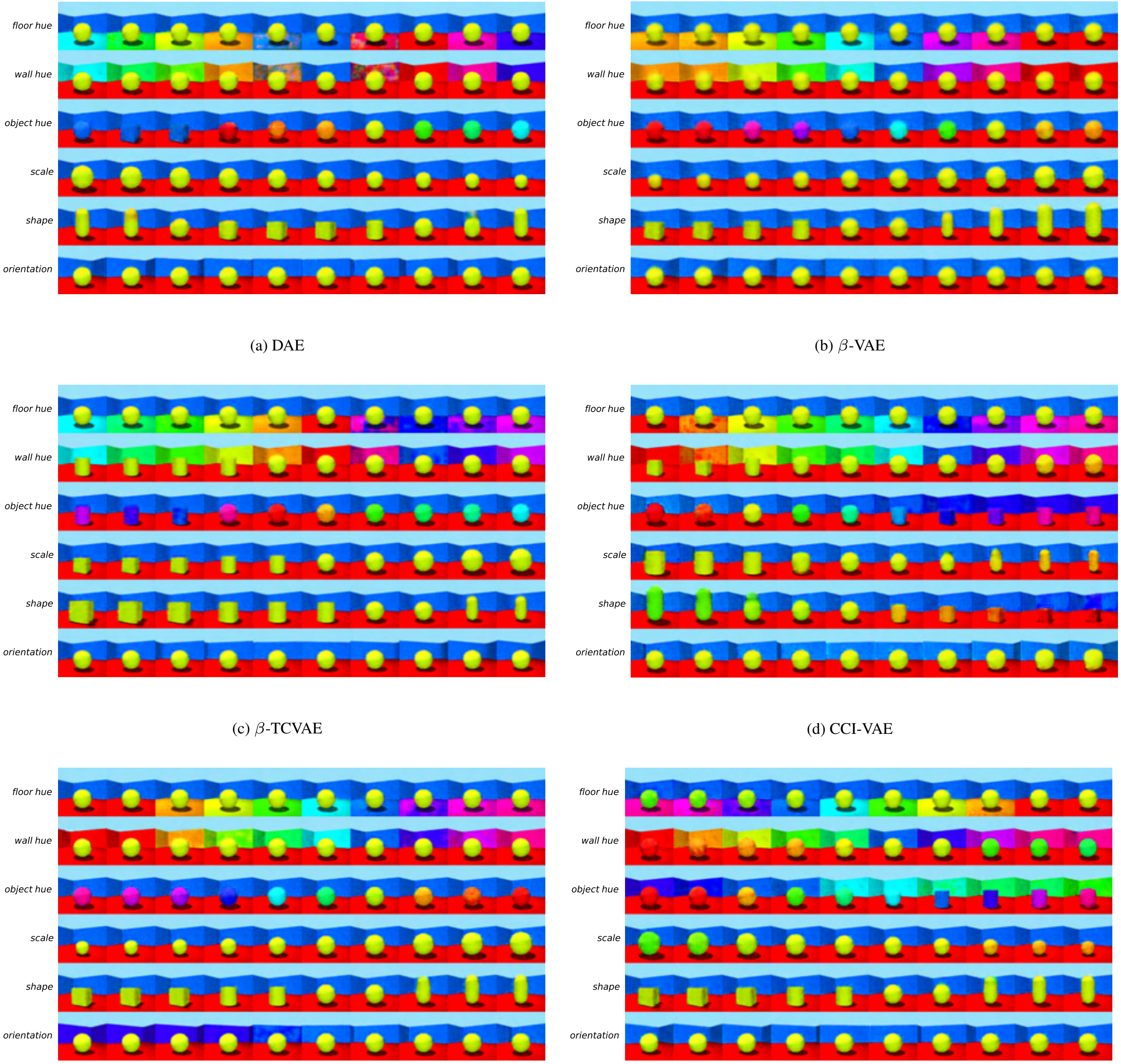
(c) β-TCVAE

(d) FVAE

(e) DIPVAE

(f) LSBDVAE

Figure 7. Reconstructions of latent traversals across each latent dimension in the 3D Teapots dataset. Due to the space limit, we omit the result from AE, VAE, CCI-VAE, InfoVAE and WAE, which have low scores in 3D Teapots dataset.



(a) DAE

(b) β-VAE

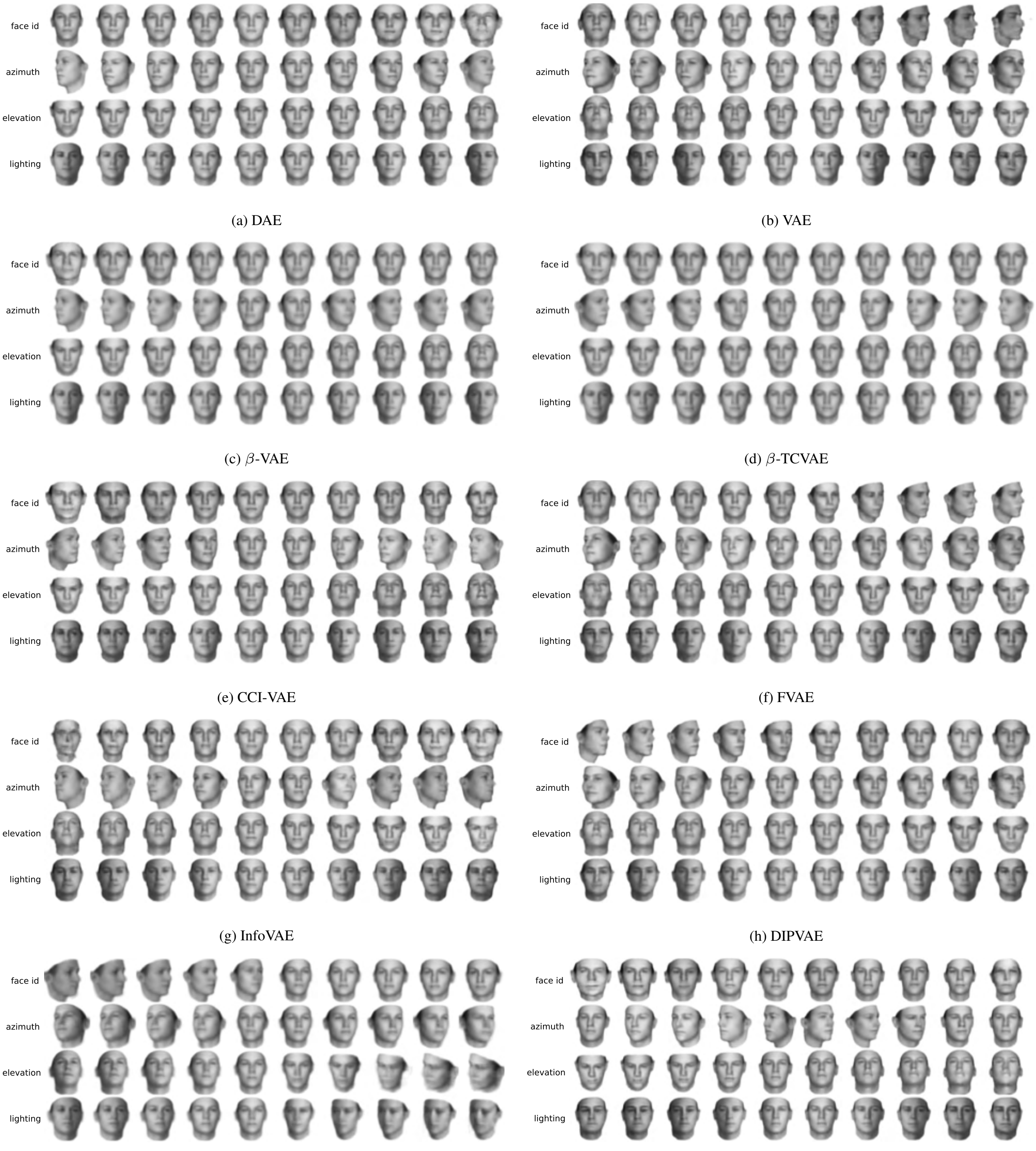
(c) β-TCVAE

(d) CCI-VAE

(e) FVAE

(f) DIPVAE

Figure 8. Reconstructions of latent traversals across each latent dimension in the 3D Shape dataset. Due to the space limit, we omit the result from AE, VAE, InfoVAE, LSBDVAE and WAE, which have low scores in 3D Shape dataset.



(a) DAE

(b) VAE

(c) β-VAE

(d) β-TCVAE

(e) CCI-VAE

(f) FVAE

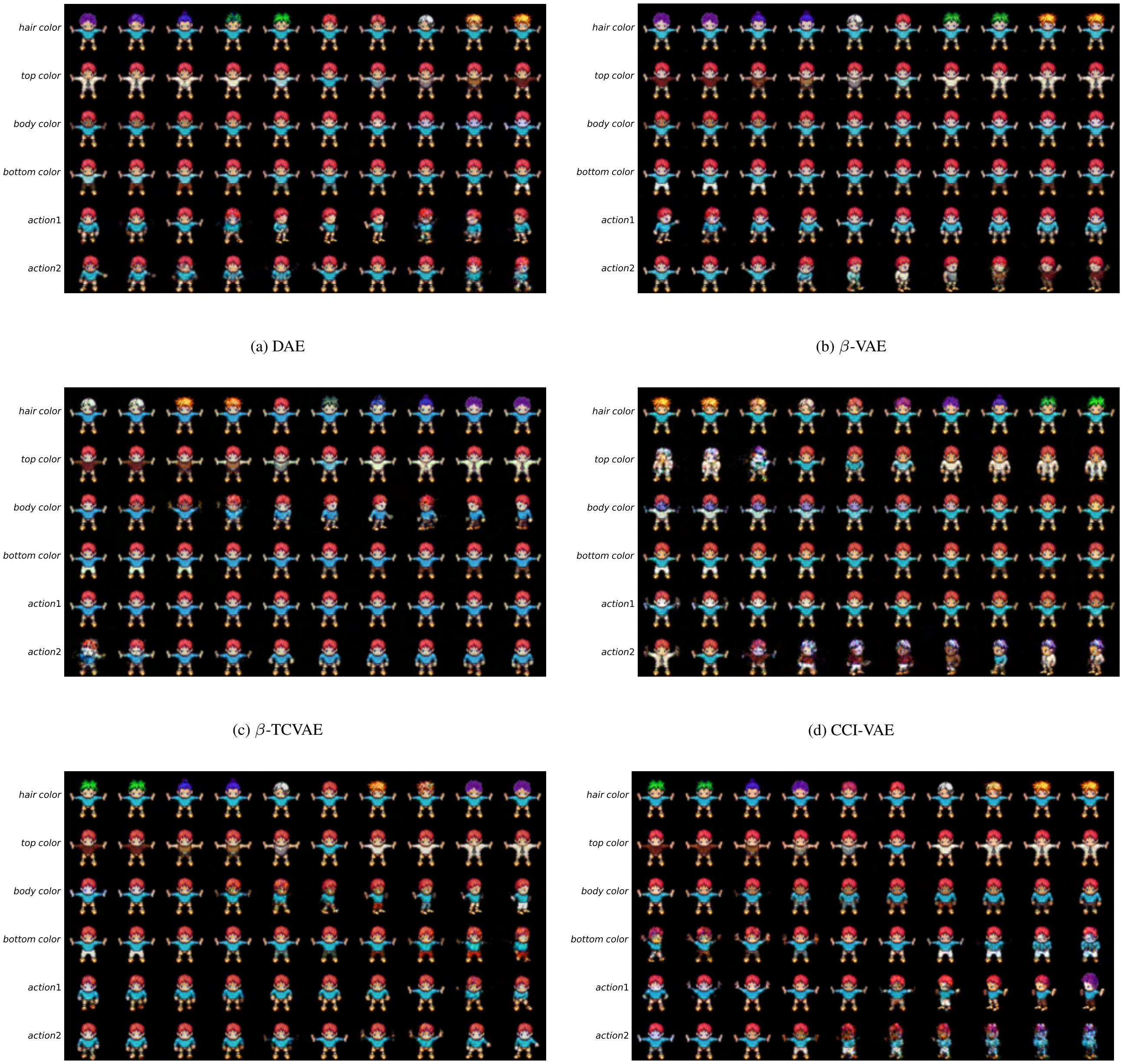
(g) InfoVAE

(h) DIPVAE

(i) WAE

(j) LSBDVAE

Figure 9. Reconstructions of latent traversals across each latent dimension in the 3D Face Model dataset. Due to the space limit, we omit the result from AE, which have low scores in 3D Shape dataset.



(a) DAE

(b) β-VAE

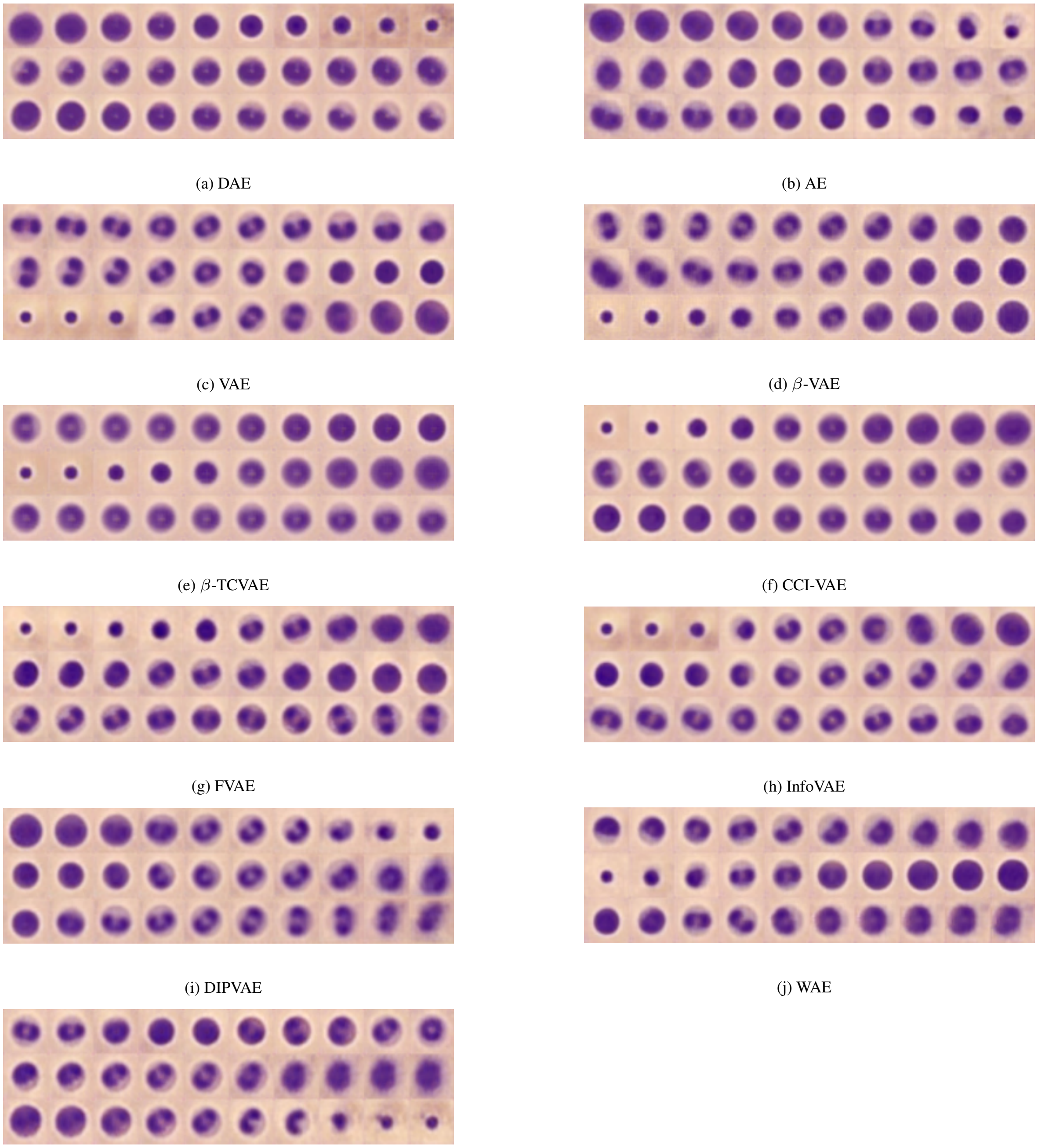
(c) β-TCVAE

(d) CCI-VAE

(e) FVAE

(f) DIPVAE

Figure 10. Reconstructions of latent traversals across each latent dimension in the Sprites dataset. Due to the space limit, we omit the result from AE, VAE, InfoVAE, LSBDVAE and WAE. The ground truth factors are the color of hair, tops, body, bottom and actions.



(a) DAE

(b) AE

(c) VAE

(d) β-VAE

(e) β-TCVAE

(f) CCI-VAE

(g) FVAE

(h) InfoVAE

(i) DIPVAE

(j) WAE

(k) LSBDVAE

Figure 11. Reconstructions of latent traversals across each latent dimension in the Blood Cell dataset. Some of the models independently learn size and shape of blood cells. For example, for DAE, the first dimension in the latent space corresponds to the size of cells and the second and the third dimensions correspond to changes in shape in different directions.

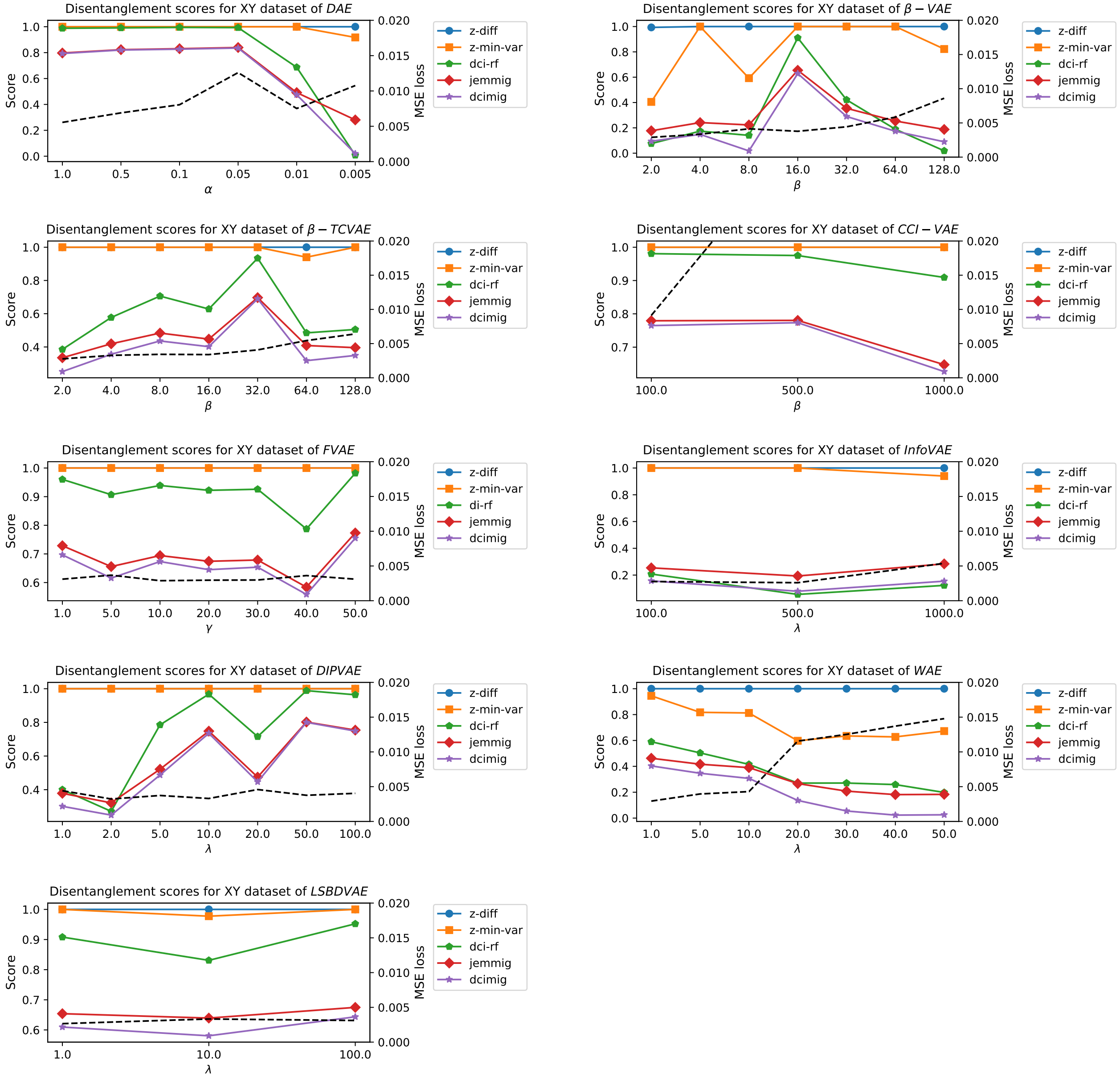
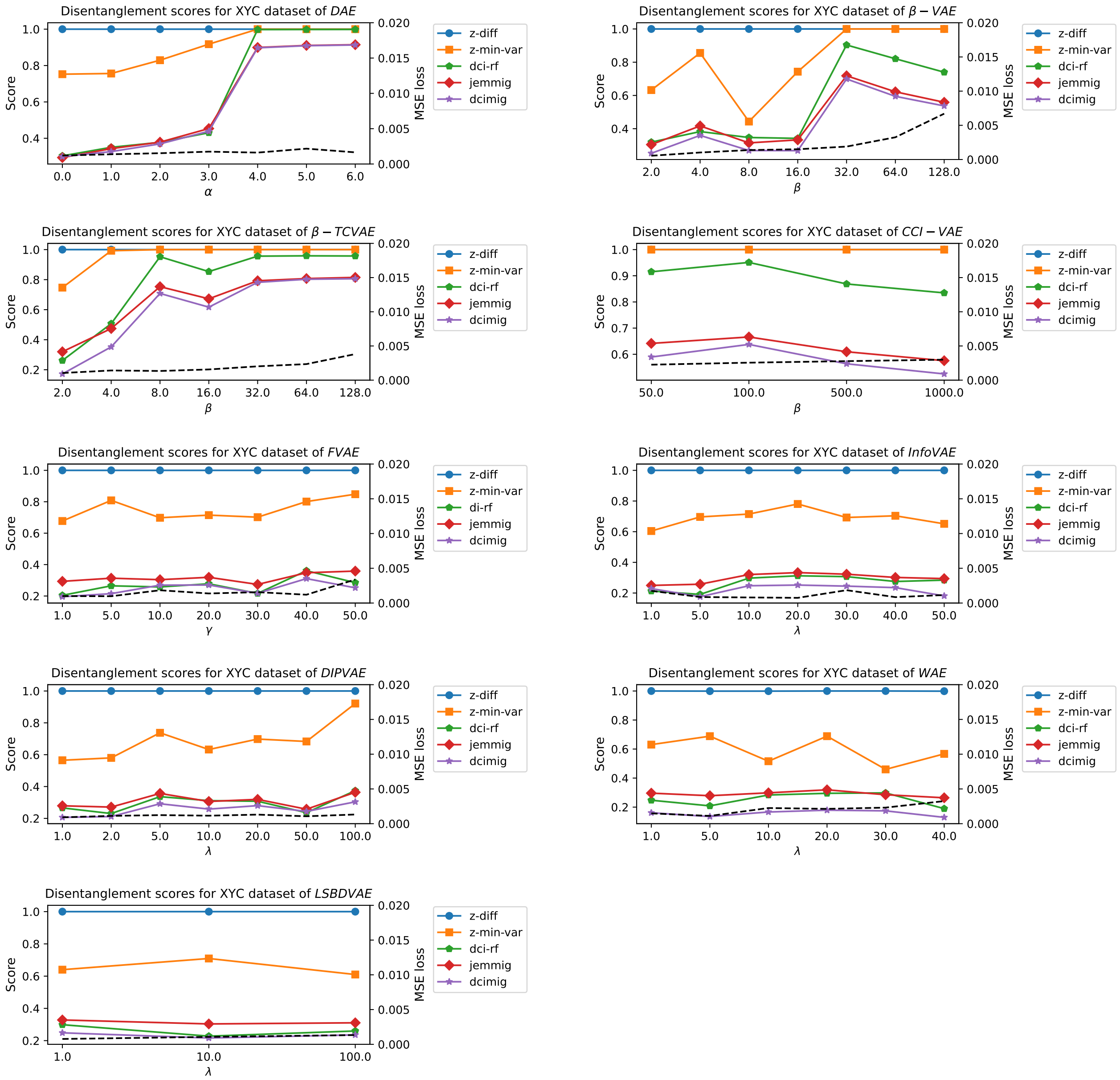


Figure 12. Disentanglement scores with the XY dataset with respect to hyperparameters. For DAE, w_1 and w_2 obtained by Algorithm 1 are the same in this dataset. Hence, the desirable value for both w_1 and w_2 is 1. The result shows that when α, lower than 0.05, is assigned to the second dimension, the disentanglement scores also become lower.



Disentanglement scores for XYC dataset of *DAE*

Disentanglement scores for XYC dataset of *β − VAE*

Disentanglement scores for XYC dataset of *β − TCVAE*

Disentanglement scores for XYC dataset of *CCI − VAE*

Disentanglement scores for XYC dataset of *FVAE*

Disentanglement scores for XYC dataset of *InfoVAE*

Disentanglement scores for XYC dataset of *DIPVAE*

Disentanglement scores for XYC dataset of *WAE*

Disentanglement scores for XYC dataset of *LSBDVAE*

Figure 13. Disentanglement scores with the XYC dataset with respect to hyperparameters.

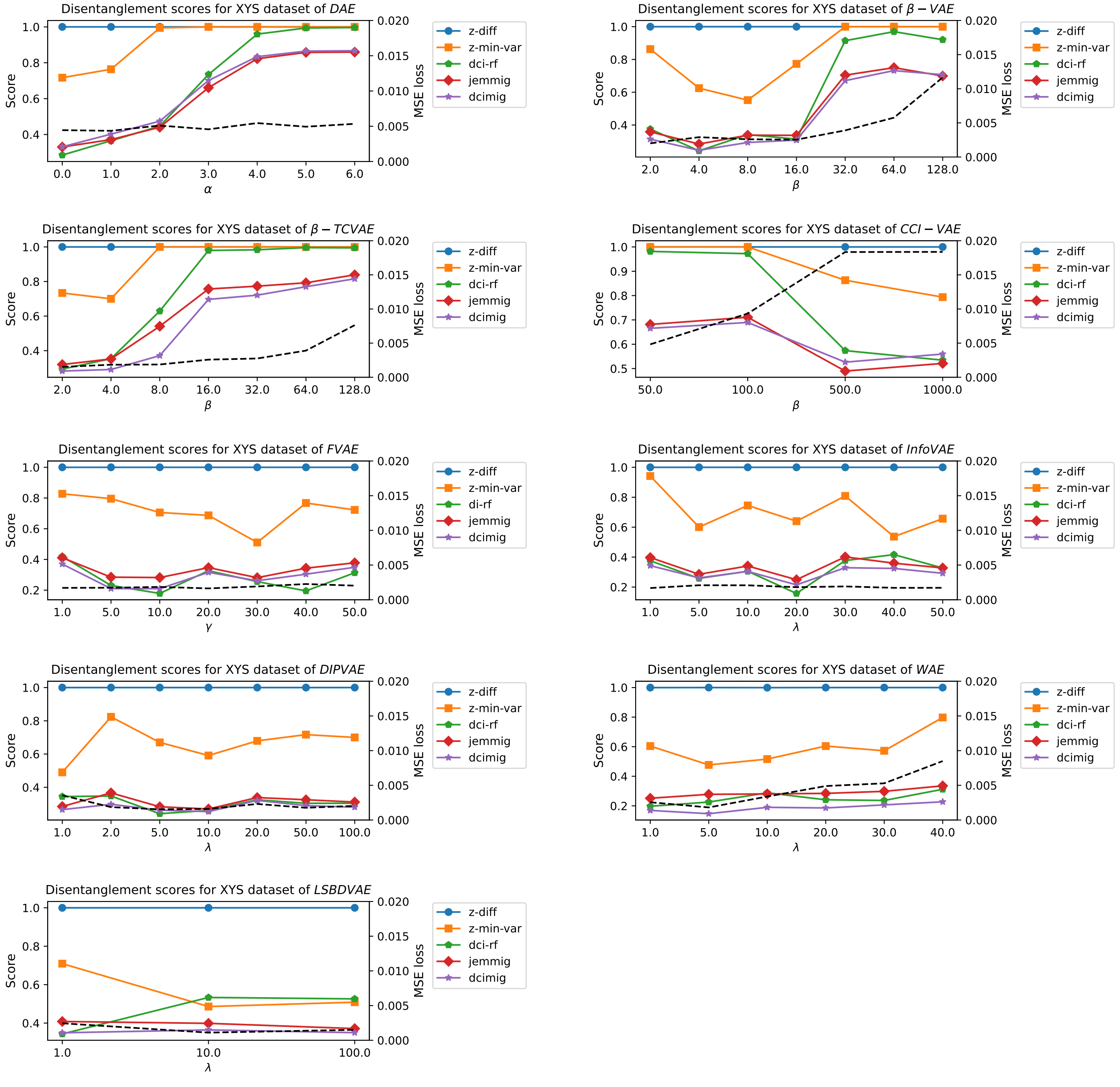


Figure 14. Disentanglement scores with the XYS dataset with respect to hyperparameters.

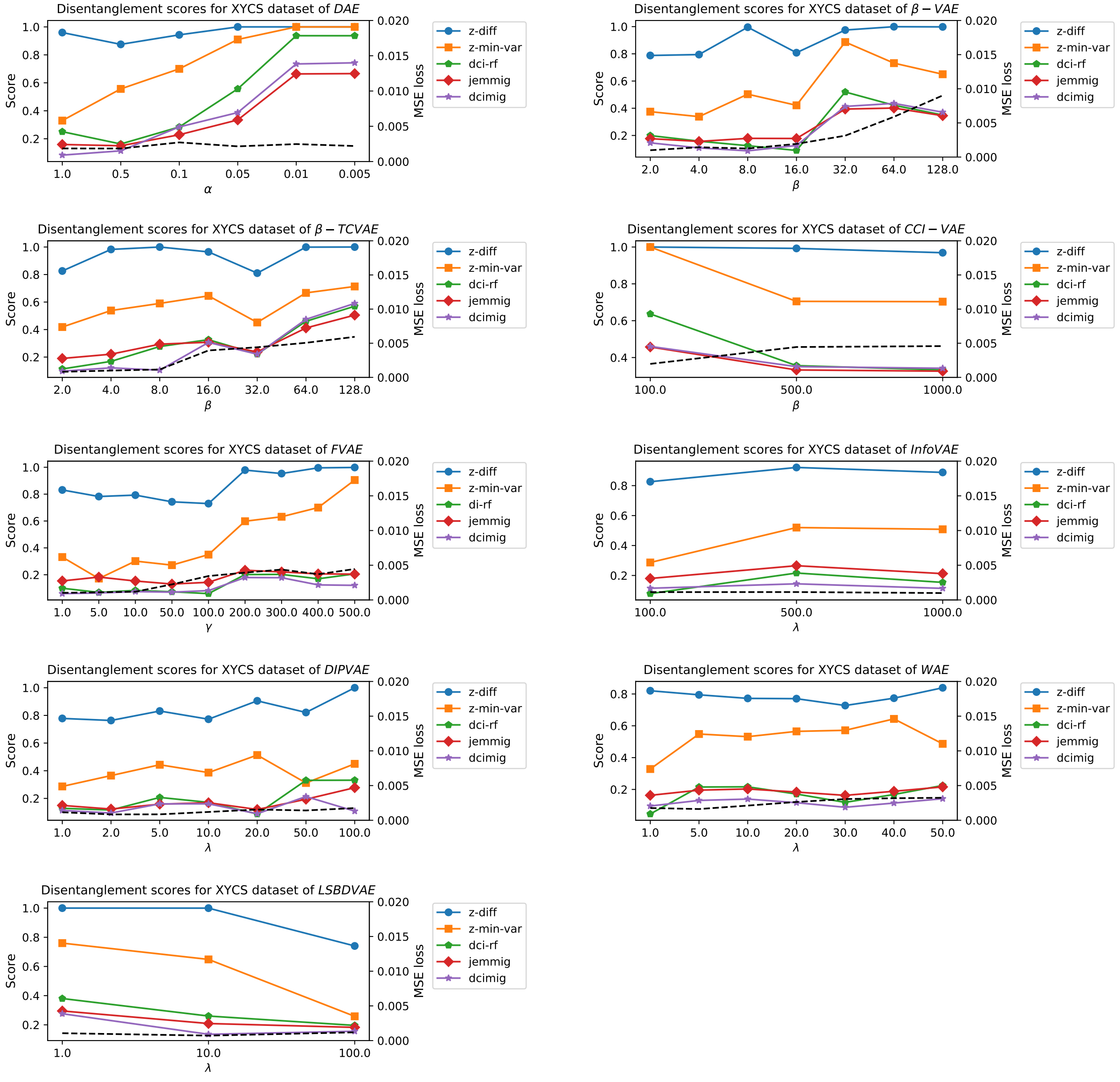


Figure 15. Disentanglement scores with the $XYCS$ dataset with respect to hyperparameters.