

# Weakly Supervised Disentanglement by Pairwise Similarities

**Junxiang Chen, Kayhan Batmanghelich**

Department of Biomedical Informatics  
University of Pittsburgh, Pittsburgh, PA 15232, US  
{juc91,kayhan}@pitt.edu

## Abstract

Recently, researches related to unsupervised disentanglement learning with deep generative models have gained substantial popularity. However, without introducing supervision, there is no guarantee that the factors of interest can be successfully recovered (Locatello et al. 2018). Motivated by a real-world problem, we propose a setting where the user introduces weak supervision by providing similarities between instances based on a factor to be disentangled. The similarity is provided as either a binary (yes/no) or a real-valued label describing whether a pair of instances are similar or not. We propose a new method for weakly supervised disentanglement of latent variables within the framework of Variational Autoencoder. Experimental results demonstrate that utilizing weak supervision improves the performance of the disentanglement method substantially.

## Introduction

Disentanglement learning is a task of finding latent representations that separate the explanatory factors of variations in the data (Bengio, Courville, and Vincent 2013). In recent years, several methods (Higgins et al. 2017; Kim and Mnih 2018; Chen et al. 2018; Lopez et al. 2018) have been proposed to solve disentanglement learning under the Variational Autoencoder (VAE) framework. However, most of these existing methods are unsupervised. In this paper, we focus on improving the disentangling performance by utilizing weak supervisions in terms of pairwise similarities.

Locatello et al. (2018) showed that unsupervised disentanglement learning is fundamentally impossible if no inductive biases on models and datasets are provided. Existing unsupervised methods control the implicit inductive biases by choosing the hyperparameters. However, the factor of interest is not guaranteed to be successfully recovered by only tuning the hyper-parameters. Providing strong supervisions with discrete or real-valued labels have been previously suggested (Narayanaswamy et al. 2017;

Kulkarni et al. 2015). However, such supervision can be expensive to acquire.

Our method is motivated by a real-world problem. In this problem, we want to understand how the Computer Tomography (CT) images are related to the severity of Chronic Obstructive Pulmonary Disease (COPD), which is a devastating disease related to cigarettes smoking. Since COPD manifests itself as airflow limitation, its severity can be measured via spirometry (meaning the measuring of breath). However, the disease severity is usually measured by combining two (Vestbo et al. 2013) or three (Quanjer et al. 2012) spirometric measures. It is not obvious how we can represent disease severity with one real value. Therefore, we represent disease severity using real-valued pairwise similarities between subjects, which are computed based on spirometric measures. The available CT images and the pairwise similarities motivate us to develop a disentanglement method that utilizes pairwise similarities when analyzing images.

In this paper, we assume that we are provided a measure of similarity between instances based on a specific factor of interest, in addition to the observations. The pairwise similarity can be binary (yes/no) or real-valued and may only be provided for a few pairs of instances. The goal is to learn disentangled representations such that a subset of the latent variables explain the factor of interest, but do not convey information about other factors of variations. We propose to achieve this goal by constructing a VAE model that generates both the samples and the pairwise similarities based on latent representations. We achieve disentanglement by letting the pairwise similarities depend on a subset of the latent variables but independent of the other latent variables, and penalizing the information capacity of the dependent latent variables. Our empirical evaluations on several benchmark datasets and the COPD dataset show that providing pairwise similarities improves the performance of the disentanglement method substantially.

**Contributions** We make the following contributions in this paper: (1) We design a latent variable model that enables a user to provide similarities between instances in the desired latent space. (2) The similarity can be a binary or real-valued value provided for all or a subset of the pairs of instances. We formulate the model with a VAE framework and propose

an efficient algorithm to train the model. (3) We conduct extensive experiments on benchmark datasets and a real-world dataset. Experimental results demonstrate that introducing weak supervision improves the disentanglement performance in different tasks.

## Background

**$\beta$ -VAE** The  $\beta$ -VAE (Higgins et al. 2017) is base for many disentanglement methods. It introduces the inductive bias by increasing the weight of the KL divergence term in the evidence lower bound (ELBO) objective function, defined as

$$\mathcal{L}_{\beta\text{-VAE}} = \max_{\theta, \phi} \mathbb{E}_{\mathbf{x}_n \sim p_{\text{data}}} [\mathbb{E}_{q_{\phi}(\mathbf{z}_n | \mathbf{x}_n)} [\log p_{\theta}(\mathbf{x}_n | \mathbf{z}_n)] - \beta \mathcal{D}_{KL}(q_{\phi}(\mathbf{z}_n | \mathbf{x}_n) || p(\mathbf{z}))], \quad (1)$$

where  $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$  and  $\mathbf{Z} = \{\mathbf{z}_n\}_{n=1}^N$  denote the observed samples and the corresponding latent variables respectively, and  $N$  is the number of samples. We use  $p_{\theta}(\mathbf{x}_n | \mathbf{z}_n)$  and  $q_{\phi}(\mathbf{z}_n | \mathbf{x}_n)$  to represent the decoder and encoder networks that are parameterized by  $\theta$  and  $\phi$ , respectively. We let  $\mathcal{D}_{KL}(\cdot || \cdot)$  denote the KL divergence and  $p(\mathbf{z})$  denote prior distribution for  $\mathbf{z}$ . In this paper, we let  $p(\mathbf{z})$  be an isotropic unit Gaussian distribution. In the equation,  $\beta \geq 1$  is a hyperparameter that controls the weight for the KL divergence term.

## Method

We assume that we have access to the noisy observations of the similarities for pairs of instances. We use  $\mathbf{Y} = \{y_{ij}\}_{(i,j) \in \mathcal{J}}$  to represent the set of observed similarities, where  $\mathcal{J} \subseteq \{(i, j) | i, j \in \{1, \dots, N\}\}$ . Note that not all pairwise similarity labels are necessarily observed. We allow  $y_{ij}$  to be either binary ( $y_{ij} \in \{0, 1\}$ ) or real-valued between 0 and 1, where a larger value of  $y_{ij}$  indicates a stronger similarity between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ .

In the following sections, we first explain the general framework of our model. We then discuss how the pairwise similarities can be incorporated into the model. Finally, we introduce a regularization term that encourages disentanglement.

### The General Framework

We assume that both  $\mathbf{X}$  and  $\mathbf{Y}$  are noisy observations; hence, we use a probabilistic approach to model uncertainty. We adopt the VAE framework (Kingma and Welling 2013) such that  $\mathbf{x}_n$  is reconstructed based on the latent variables  $\mathbf{z}_n$ . We assume that the latent variable  $\mathbf{z}$  is divided into two subspaces, i.e.,  $\mathbf{z} = [\mathbf{z}^{(u)}, \mathbf{z}^{(v)}]$ , where  $\mathbf{z}^{(u)}$  (with  $d^{(u)}$  dimensions) accounts for the latent variables *relevant* to the factors of interest, while  $\mathbf{z}^{(v)}$  (with  $d^{(v)}$  dimensions) accounts for the rest of information. Since  $y_{ij}$  represents pairwise similarity based on the factors of interest, it is only dependent on the coordinates of the latent variables of  $\mathbf{x}_i$  and  $\mathbf{x}_j$  in the  $\mathbf{z}^{(u)}$  subspace; i.e.,  $p(y_{ij} | \mathbf{z}_i, \mathbf{z}_j) = p(y_{ij} | \mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)})$ . Therefore, the joint distribution of the observed instances and similar-

ties has the following factorization,

$$p_{\theta}(\mathbf{X}, \mathbf{Y} | \mathbf{Z}) = \prod_{n=1}^N p_{\theta}(\mathbf{x}_n | \mathbf{z}_n) \prod_{(i,j) \in \mathcal{J}} p(y_{ij} | \mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}). \quad (2)$$

This model can be represented using a graphical model as shown in Figure 1. In this equation,  $p_{\theta}(\mathbf{x}_n | \mathbf{z}_n)$  represents the reconstruction model of the VAE framework. We explain  $p(y_{ij} | \mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)})$  in the next section.

### Modeling Pairwise Similarity

We view  $y_{ij}$  as the noisy observation of the similarity between  $i$ 'th and  $j$ 'th instances, which can be either a binary or a real-value measurement. We use the following function to model conditional of  $y_{ij}$  for both cases,

$$p(y_{ij} | \mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}) = \frac{1}{\mathcal{C}} \left( g(\mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}) \right)^{y_{ij}} \left( 1 - g(\mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}) \right)^{1-y_{ij}}, \quad (3)$$

where  $\mathcal{C}$  is the normalization constant and  $g(\cdot, \cdot)$  is a function encoding the strength of the similarity given the relevant latent variables  $\mathbf{z}_i^{(u)}$  and  $\mathbf{z}_j^{(u)}$ . In Equation (3), when  $y_{ij}$  is a binary variable,  $g$  can be viewed as probability that a user labels  $y_{ij}$  as 1. Hence, we choose  $g$  to return a value between 0 and 1 and  $\mathcal{C} = 1$ . When  $y_{ij}$  is real-valued between 0 and 1, Equation (3) enables us to compute the normalization constant in a closed form:

$$\begin{aligned} \mathcal{C} &= \int_0^1 \left( g(\mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}) \right)^{y_{ij}} \left( 1 - g(\mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}) \right)^{1-y_{ij}} dy_{ij} \\ &= \frac{2g(\mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}) - 1}{\log(g(\mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)})) - \log(1 - g(\mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}))}. \end{aligned} \quad (4)$$

We adopt the following form for  $g$ :

$$g(\mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}) = \sigma \left( \eta_1 \left( \eta_2 - \|\mathbf{z}_i^{(u)} - \mathbf{z}_j^{(u)}\|_2^2 \right) \right), \quad (5)$$

where  $\eta_1$  and  $\eta_2$  are positive real hyperparameters controlling the ‘‘steepness’’ and ‘‘threshold’’ of the similarity, respectively; and  $\sigma(\cdot)$  is the sigmoid function. When  $\eta_1 \rightarrow \infty$ ,  $g(\mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)})$  can be regarded as a hard thresholding function indicating whether or not  $\|\mathbf{z}_i^{(u)} - \mathbf{z}_j^{(u)}\|_2^2$  is smaller than  $\eta_2$ . We replace the hard thresholding function with a sigmoid function  $\sigma(\cdot)$  to make sure this function differentiable. The Figure 3 shows that when  $\|\mathbf{z}_i^{(u)} - \mathbf{z}_j^{(u)}\|_2^2$  is small, it is more likely to have a large  $y_{ij}$  and vice versa.

### Disentanglement via Regularization

Our goal of disentanglement is to encode all information about the factor of interest into  $\mathbf{z}^{(u)}$  and to prevent it from containing irrelevant information. The general idea is to limit the capacity of  $\mathbf{z}^{(u)}$ ; hence, its capacity can be used only for the relevant factors. Similar to the  $\beta$ -VAE, we use a regularized ELBO that increases the weight of the KL divergence between the approximate posterior (i.e.,  $q_{\phi}(\mathbf{z}_n | \mathbf{x}_n)$ ) and the prior (i.e.,  $p(\mathbf{z}^{(u)})$ ), but we do not impose extra regular-

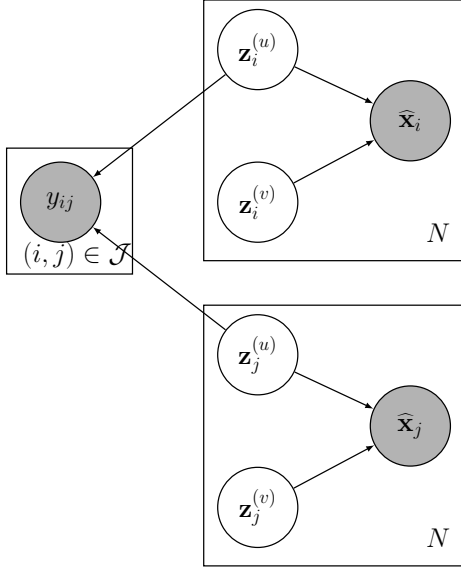


Figure 1: The decoder model  $p(\mathbf{X}, \mathbf{Y}|\mathbf{Z})$ .

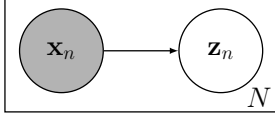


Figure 2: The encoder model  $q(\mathbf{Z}|\mathbf{X})$ .

ization for  $\mathbf{z}^{(v)}$ . The regularization term is defined as

$$\begin{aligned} \mathcal{R} = & -\mathbb{E}_{\mathbf{x}_n \sim p_{\text{data}}} \left[ \beta \mathcal{D}_{KL} \left( q_{\phi}(\mathbf{z}_n^{(u)} | \mathbf{x}_n) || p(\mathbf{z}^{(u)}) \right) \right] \\ & -\mathbb{E}_{\mathbf{x}_n \sim p_{\text{data}}} \left[ \mathcal{D}_{KL} \left( q_{\phi}(\mathbf{z}_n^{(v)} | \mathbf{x}_n) || p(\mathbf{z}^{(v)}) \right) \right], \end{aligned} \quad (6)$$

where  $\beta \geq 1$  is a real-valued hyperparameter that controls the weight of KL divergence.

## Overall Model

The overall objective can be written as follows,

$$\begin{aligned} \mathcal{L} = & \max_{\theta, \phi} \mathbb{E}_{\mathbf{x}_n \sim p_{\text{data}}} \left[ \mathbb{E}_{q_{\phi}(\mathbf{z}_n | \mathbf{x}_n)} [\log p(\mathbf{x}_n | \mathbf{z}_n)] \right] \\ & + \mathbb{E}_{(i,j) \in \mathcal{J}} \left[ \mathbb{E}_{q_{\phi}(\mathbf{z}_i, \mathbf{z}_j | \mathbf{x}_i, \mathbf{x}_j)} \left[ \log p(y_{ij} | \mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)}) \right] \right] + \mathcal{R}, \end{aligned} \quad (7)$$

where  $p(y_{ij} | \mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)})$  is defined in Equation (3) and  $\mathcal{R}$  is defined in Equation (6). We use the encoder  $q(\cdot)$ , to disentangle the factors at test time. Since we only have access to the weak labels  $\mathbf{Y}$  at training time, the encoder can only take  $\mathbf{x}_n$  as an argument. We use stochastic gradient descent (SGD) to optimize for  $\theta$  and  $\phi$ .

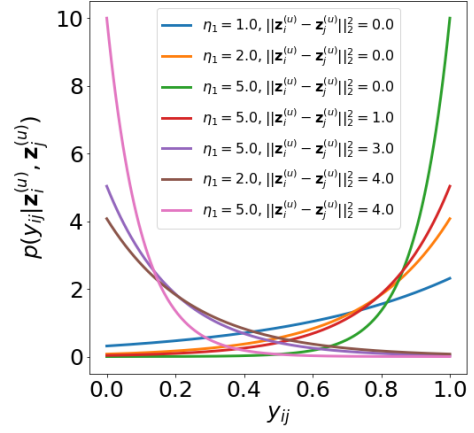


Figure 3: Plot for  $p(y_{ij} | \mathbf{z}_i^{(u)}, \mathbf{z}_j^{(u)})$  for real-valued  $y_{ij}$ . We fix the thresholding hyperparameter  $\eta_2 = 2$ . When  $\|\mathbf{z}_i^{(u)} - \mathbf{z}_j^{(u)}\|_2^2$  is small, it is more likely to have a large  $y_{ij}$  and vice versa. The hyperparameter  $\eta_1$  controls the “steepness” of the distribution.

## Related Work

There have been several unsupervised methods for learning disentangled representations with VAE, including  $\beta$ -VAE (Higgins et al. 2017), factor VAE (Kim and Mnih 2018),  $\beta$ -TCVAE (Chen et al. 2018) and HCV (Lopez et al. 2018). These methods achieve disentanglement by encouraging latent variables to be independent with each other. With these methods, the users can impact the disentanglement results only by tuning the hyperparameter. However, without explicit supervision, it is difficult to control the correspondence between a learned latent variable and a semantic meaning, and it is not guaranteed that the factor of interest can always be successfully disentangled (Locatello et al. 2018). In contrast, our proposed method utilizes the pairwise similarities as explicit supervision, which encourages the model to disentangle the factor of interest.

There have been attempts to improve disentanglement performance by introducing supervision. Narayanaswamy et al. (2017) and Kulkarni et al. (2015) propose semi-supervised VAE methods that learn disentangled representation, by making use of partially observed class labels or real-value targets. Bouchacourt, Tomioka, and Nowozin (2018) introduces supervision via grouping the samples. Our proposed method utilizes pairwise similarities.

Gaussian Process Prior VAE (GPPVAE) (Casale et al. 2018) assigns a Gaussian process prior to the latent variables. It makes use of the pairwise similarities between instances, by modeling the covariances between instances with a kernel function. GPPVAE does not focus on learning disentangled representation. Besides, GPPVAE requires the covariance matrix to be positive semi-definite, and the complete covariance matrix is observed without any missing values. In practice, a user might fail to provide labels satisfying these requirements. Our proposed method allows unobserved similarities and does not require the similarity matrix to be positive semi-

definite.

Dual Swap Disentangling (DSD) (Feng et al. 2018) and Generative Multi-view Model (Chen, Denoyer, and Artières 2018) are VAE and GAN models that make use of binary similarity labels, respectively. They both assume that the latent variables  $\mathbf{z}$  can be separated into subspaces  $\mathbf{z}^{(u)}$  and  $\mathbf{z}^{(v)}$ , which is similar to our proposed model. However, both methods assume that similar instances share similar  $\mathbf{z}^{(u)}$ , but do not force dissimilar instances to be encoded differently in  $\mathbf{z}^{(u)}$ . As shown in our experiments, DSD is likely to converge into a trivial solution that all instances share similar  $\mathbf{z}^{(u)}$ , despite the similarity labels. In contrast, our proposed model is able to make use of both binary and real-valued similarities and it avoids this trivial solution by utilizing both similarity and dissimilarity labels.

## Experiments

In this section, we evaluate our method quantitatively and qualitatively. We perform experiments for both binary and real-value similarity values. Our method is compared against a few competing methods qualitatively in terms of recovering semantic factors for rotating object or identifying the labels on benchmark datasets, where we evaluate our approach quantitatively on the recovery of the ground-truth factors. Then we apply our method to analyze the real-world COPD dataset. Finally, we study the robustness of our method for the choice of hyperparameters, the proportion of the observed pairwise similarity and the noisiness of the observed similarities.

In the following, we first introduce datasets used for our experiments, followed by the discussion of the various quantitative metrics used in this paper. We then report the results of the experiments.

### Datasets and Competitive Methods

We evaluate our methods on five datasets: MNIST (LeCun and Cortes 2010), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), Yale Faces (Georgiades, Belhumeur, and Kriegman 2001), 3D chairs (Aubry et al. 2014) and 3D cars (Krause et al. 2013). The details of these datasets are summarized in Table 1. For each dataset, we generate a subset of pairwise similarities based on one ground-truth factor of variations, as shown in the table. Unless specified otherwise, we let the number of observed pairwise labels be 0.01% of the number of all possible pairs. For the MNIST and fashion-MNIST datasets, we define  $y_{ij} = \mathbb{1}(t_i = t_j)$  where  $t_i$  and  $t_j$  are the ground-truth labels for the sample  $i$  and  $j$ , and  $\mathbb{1}$  is the indicator function. For Yale faces, 3D chairs and 3D cars, we use the Gaussian RBF kernel to define the similarities, i.e.,  $y_{ij} = \exp(-\delta(t_i, t_j)^2 / \sigma^2)$ . Since the ground-truth factors in all three datasets involve azimuth angles, we use  $\delta$  to denote the difference between two azimuth angles, e.g.,  $\delta(350^\circ, 20^\circ) = 30^\circ$ .

In addition to regular VAE (Kingma and Welling 2013), we compare our proposed method with three disentanglement approaches based on VAE, including  $\beta$ -VAE (Higgins et al. 2017), factor VAE (Kim and Mnih 2018),  $\beta$ -TCVAE (Chen et al. 2018). As a supervised disentanglement method, we compare our approach with Dual Swap Disentangling

(DSD) (Feng et al. 2018). The DSD is designed to analyze binary similarities and cannot be applied to real-valued similarities. To make all methods comparable, we use the same encoder and decoder architectures for all the methods, which include four convolutional layers and one fully connected layer. To select the hyperparameters for our method, we use 5-fold cross validation on the training data. Since most of the competing methods are unsupervised, we choose the hyperparameters for them that achieves the best performance on the held-out data, which is advantageous for the competing methods resulting in an over-estimation of their performances. We define the metrics for the performance in the following section.

### Quantitative Comparison

In this section, we perform two quantitative experiments. One is computing the Mutual Information Gap (MIG), which is a popular metric for evaluation of the disentanglement method, and the second experiment is a prediction task.

**Mutual Information Gap (MIG)** We evaluate the disentanglement performance by computing the Mutual Information Gap (MIG) as introduced in (Chen et al. 2018). Let  $t$  represent the ground-truth factor and  $\mathcal{I}(\cdot, \cdot)$  represent the mutual information between two random variables (with 1 or more dimensions). In our model, since we assume  $\mathbf{z}^{(u)}$  is relevant to  $t$ , we expect  $\mathcal{I}(\mathbf{z}^{(u)}; t)$  to be large; while  $\mathcal{I}(\mathbf{z}_d^{(v)}; t)$  to be small for each dimension  $d \in \{1 \dots d^{(v)}\}$ . Therefore, we can measure the disentanglement by computing the mutual information gap, defined as

$$\frac{1}{\mathcal{H}(t)} \left( \mathcal{I}(\mathbf{z}^{(u)}; t) - \max_{d \in \{1 \dots d^{(v)}\}} \mathcal{I}(\mathbf{z}_d^{(v)}; t) \right), \quad (8)$$

where  $\mathcal{H}(\cdot)$  represents the entropy of a random variable. The values of  $\mathcal{I}(\cdot, \cdot)$  and  $\mathcal{H}(\cdot)$  can be empirically estimated as explained in (Chen et al. 2018). For each dataset, the dimensionality of  $\mathbf{z}^{(u)}$ , denoted by  $d^{(u)}$ , is shown in the Table 2. Our method directly produces the  $\mathbf{z}^{(u)}$  and  $\mathbf{z}^{(v)}$  terms that can be plugged into Equation (8). Since the competing methods are unsupervised, the choice of the indices for  $\mathbf{z}^{(u)}$  and  $\mathbf{z}^{(v)}$  is not clear. For those methods, we first rank all latent variables based on the mutual information with respect to the ground-truth. Then, we pick the top  $d^{(u)}$  random variables to form  $\mathbf{z}^{(u)}$  and the remaining latent variables are assigned to  $\mathbf{z}^{(v)}$ . The MIG values are estimated on the held-out data.

The values in Table 2 report the MIG for various methods. Our proposed method achieves substantially higher MIG values than other approaches. It outperforms the second-best methods by more than 40% in all five datasets. The results illustrate the importance of introducing supervision in disentanglement tasks. Although DSD is a supervised method that is formulated to incorporate binary pairwise similarities, it fails to disentangle the ground-truth factor. We speculate that the failure is due to convergence to a trivial solution, as mentioned in the Related Work Section.

**Prediction Task** We use  $\mathbf{z}^{(u)}$  as an input to a regression or classification method to predict the ground truth. We use the 5 Nearest Neighbour (5-NN) method for both classification and regression. Table 3 reports the outcome for different datasets,

Table 1: The Dataset

Name	Training instances	Held-out instances	Image size	The ground-truth factor
MNIST	60,000	10,000	$28 \times 28 \times 1$	discrete labels
Fashion-MNIST	60,000	10,000	$28 \times 28 \times 1$	discrete labels
Yale Faces	1,903	513	$64 \times 64 \times 1$	azimuth lighting
3D chairs	69,131	17,237	$64 \times 64 \times 3$	azimuth rotations
3D cars	14,016	3,552	$64 \times 64 \times 3$	azimuth rotations

measured by Cohen’s kappa ( $\kappa$ ) and  $R^2$  with respect to the ground-truth. We measure Cohen’s kappa rather than classification accuracy because it corrects for the possibility of the agreement occurring by chance. For both measurements, a higher value indicates a better performance. We observe that our proposed method outperforms the competing methods in all tasks. This implies that instances with similar ground-truth factors are located near each other in the latent space  $\mathbf{z}^{(u)}$ .

### Qualitative Comparison

In this subsection, we illustrate the disentanglement performance of the proposed method via qualitative comparison. We use the results on the MNIST and 3D-chairs datasets as examples (for more experimental results, see the supplementary materials<sup>1</sup>).

**MNIST** Figure 4(a) demonstrates  $\mathbf{z}^{(u)}$  of the held-out instances from the MNIST dataset. Different colors represent different class labels. Figure 4(b) shows a similar concept for the competing method that achieves the highest MIG value in Table 2. We observe that the proposed model is able to learn  $\mathbf{z}^{(u)}$  such that it explains the ground-truth factor (*i.e.*, the digit class). All ten classes are well separated in the latent space with distinct centers, and instances from the same class are located close to each other. As shown in Figure 4(b), the factor-VAE is also able to learn a disentangled representation. However, regions of the instances of digit 4 and 9 are overlapping in the latent space.

To illustrate the performance of the generative model, we plot some images generated by the proposed and the competing method in Figure 6(a) and 6(b), respectively. We first randomly sample an image from the held-out data and encode it into  $\mathbf{z} = [\mathbf{z}^{(u)}, \mathbf{z}^{(v)}]$ . Then, we keep  $\mathbf{z}^{(v)}$  constant and manipulate  $\mathbf{z}^{(u)}$ . Using the new code, we generate new images that are displayed at their corresponding locations. In Figure 6(a), we find that the writing styles of ten digits are similar. This implies that  $\mathbf{z}^{(u)}$  only contains the information about the ground-truth factor and not the other factors of variation. In contrast, we observe changes in writing styles in Figure 6(b). The figure shows that the reconstructed digits have different thicknesses, angles, widths.

**3D-chairs** We repeat the same plotting process for the 3D-chairs dataset. The results are shown in Figures 5 and 7. Since the ground truth (*i.e.*, azimuth) is a cyclic value, the ideal shape of the latent variable should look like a ring, which is approximately captured by our method in Figure 5(a). For

some images, it is more challenging to determine which direction the chair faces (some chairs are almost centrosymmetric). These images are encoded into the regions close to the origin. Without proper supervision,  $\beta$ -VAE is not able to fully recover the complex underlying structure of the ground-truth factor, as shown in Figure 5(b).

We manipulate  $\mathbf{z}^{(u)}$  and generate the images in Figure 7. As shown in 7(a), we observe the images of chairs facing various directions, located at the ring displayed in Figure 5(a). In Figure 7(b), we observe that  $\beta$ -VAE can reconstruct the chair images facing left and right, but other reconstructed images are blurry.

### COPD dataset

A real-world application of the proposed model is to analyze the COPD dataset. The purpose of this application is to identify factors in the Computer Tomography (CT) images of the chest that are related to disease severity. We applied our method on a large-scale dataset (over 9K patients), where all patients have CT images as well as spirometric measurements. We use the spirometric measures to construct pairwise real-value similarities using Radial Basis Function.

In the COPD dataset, a ground-truth measure for disease severity is not available. Therefore, we use  $\mathbf{z}^{(u)} \in \mathbb{R}$  learned by our model to predict several clinical measurements of disease severity from different aspects, via a 5-nearest neighbor regression and classification. The clinical measurements include (1) FEV<sub>1pp</sub> measuring how quickly one can exhale, (2) Emphesyma% measuring the percentage of destructed lung tissue, (3) GasTrap% indicating amount of gas trapped in lung, and (4) GOLD score which is a six-categorical value indicating the severity of airflow limitation. In Table 4, we report  $R^2$  for the first three measurements and Cohen’s kappa coefficient ( $\kappa$ ) for the last measurement. The results suggest that our method is better than the unsupervised approach in disentangling the disease factor, as it outperforms them in predicting various measures of disease severity.

### Choice of Hyperparameters

To illustrate how the hyperparameter  $\beta$  affects the performance of our proposed method, we first plot generated images with an improperly chosen  $\beta$  in Figure 8. In this figure, we find all ten digits. However, unlike the results shown in Figure 6(a), the writing styles (thicknesses, angles, widths, sizes, *etc.*) of the generated digits change significantly. This implies a failure of disentanglement, because  $\mathbf{z}^{(u)}$  explains some factors of variations other than the one of interest (*i.e.*, digit class).

<sup>1</sup> Supplementary materials are available at <https://arxiv.org/abs/1906.01044>

Table 2: MIG metrics on the held-out data

Dataset	$d^{(u)}$	Proposed	VAE	$\beta$ -VAE	Factor-VAE	TCVAE	DSD
MNIST	2	<b>0.68</b>	0.01	0.03	0.33	0.04	0.01
Fashion-MNIST	2	<b>0.52</b>	0.11	0.28	0.36	0.19	0.01
Yale Faces	1	<b>0.42</b>	0.02	0.07	0.06	0.29	N/A
3D chairs	2	<b>0.37</b>	0.02	0.15	0.11	0.08	N/A
3D cars	2	<b>0.41</b>	0.02	0.22	0.15	0.16	N/A

DSD is designed for analyzing binary similarities, and cannot analyze real-valued similarities.

Table 3: Prediction Performance

	Dataset	Proposed	VAE	$\beta$ -VAE	Factor-VAE	TCVAE	DSD
$\kappa$	MNIST	<b>.969</b>	.494	.326	.704	.260	.030
	Fashion-MNIST	<b>.857</b>	.389	.460	.613	.482	.003
$R^2$	Yale Faces	<b>.968</b>	.397	.760	.699	.692	N/A
	3D chairs	<b>.912</b>	.155	.357	.224	.196	N/A
	3D cars	<b>.584</b>	.391	.418	.177	.110	N/A

DSD is designed for analyzing binary similarities, and cannot analyze real-valued similarities.

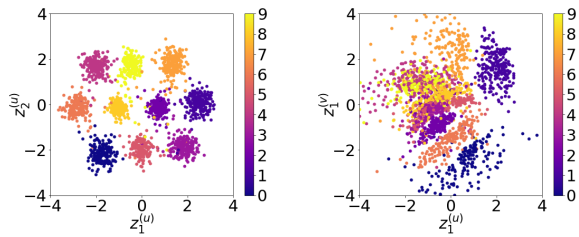
(a) Proposed ( $\beta = 10$ )(b) Factor-VAE ( $\gamma = 3$ )

Figure 4: Held-out instances in MNIST dataset.

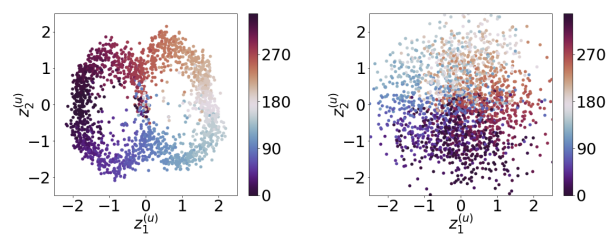
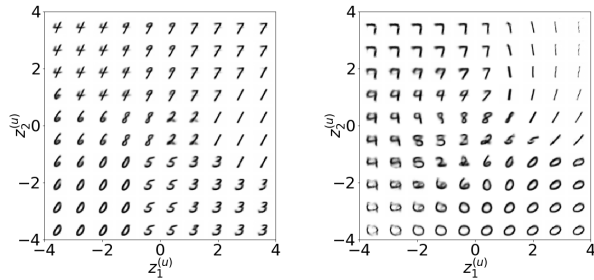
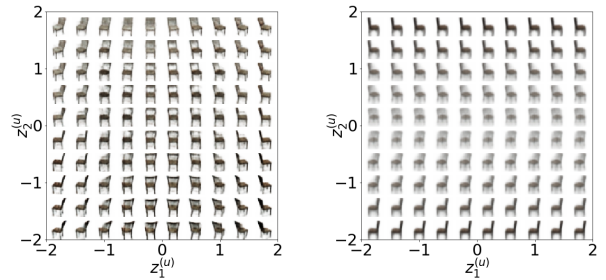
(a) Proposed ( $\beta = 3$ )(b)  $\beta$ -VAE ( $\beta = 15$ )

Figure 5: Held-out instances in 3D-chairs dataset.

(a) Proposed ( $\beta = 10$ )(b) Factor-VAE ( $\gamma = 3$ )Figure 6: Manipulating  $\mathbf{z}^{(u)}$  on MNIST dataset.(a) Proposed ( $\beta = 3$ )(b)  $\beta$ -VAE ( $\beta = 15$ )Figure 7: Manipulating  $\mathbf{z}^{(u)}$  on 3D-chairs dataset.

To find a proper  $\beta$  for each dataset, we vary  $\beta$  and conduct 5-fold cross validation with the training instances. We plot the mean log-likelihood ( $\log p_{\theta}(\mathbf{X}, \mathbf{Y}|\mathbf{Z})$ ) of five validations sets in Figure 9. We observe that a maximum log likelihood is achieved with choices of  $\beta$  between 2 to 10, but the optimal  $\beta$  differs across datasets. We choose  $\beta$  that maximizes the log-likelihood for each dataset.

We illustrate how the hyperparameters  $\eta_1$  and  $\eta_2$  affect the

disentanglement performance in Figure 10. In Figure 10(a), we fix  $\eta_2 = 2$  and vary  $\eta_1$ ; while in Figure 10(b), we fix  $\eta_1 = 1e3$  and vary  $\eta_2$ . Because the log-likelihood is a function of  $\eta_1$  and  $\eta_2$ , we report the MIG metrics for the held-out data, instead. We observe that when  $\eta_1 \geq 1e3$  and  $\eta_2 \geq 1$ , these hyperparameters have limited effects on the MIG metrics. In all other experiments, we choose  $\eta_1 = 1e3$  and  $\eta_2 = 2$ .

Table 4: Prediction Performance in the COPD dataset

		Proposed	VAE	$\beta$ -VAE	Factor-VAE	TCVAE
$R^2$	FEV <sub>1</sub> pp	<b>.431</b>	.002	.013	.010	.040
	Emphesyma%	<b>.441</b>	.027	.252	.191	.081
	GasTrap%	<b>.522</b>	.104	.279	.067	.110
$\kappa$	GOLD	<b>.236</b>	.023	.089	.061	.088

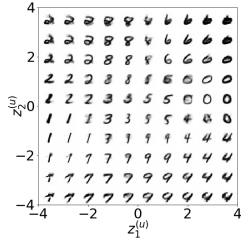


Figure 8: Generative results of the proposed result ( $\beta = 1$ ) on MNIST. With improperly chosen  $\beta$ ,  $\mathbf{z}^{(u)}$  might explain other factors of variations (thicknesses, angles, etc.) other than the digit class.

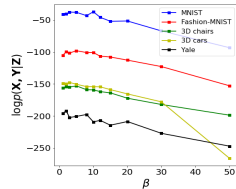


Figure 9: The mean value for the log-likelihood with different  $\beta$  in 5-fold cross validation. For each dataset, we choose  $\beta$  that maximizes the log-likelihood.

## Number of Pairwise Labels

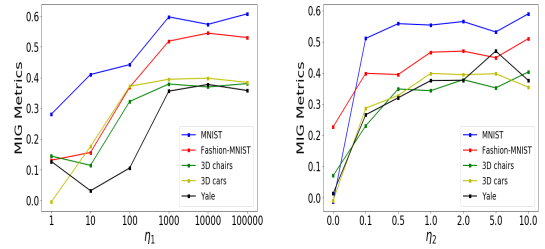
We investigate how the number of pairwise labels affects the performance of our proposed model. In Figure 11, we plot the MIG metrics for the held-out data versus the proportion of observed pairwise labels in training. We observe that in general, with more pairwise labels provided, the disentanglement performance improves. However, as the proportion approaches  $1e-4$ , the rate of improvement tapers. In all other experiments, we fix the proportion to be  $1e-4$ .

## Noisy Similarity Labels

In all previous experiments, we do not introduce noise to the pairwise similarity labels. In this section, we introduce noise controlled by the noise level  $\gamma$ . For binary labels, we flip the labels with probability  $\gamma$ . For real-valued similarities, we let  $\gamma$  be the variance of the Gaussian noise, *i.e.*, we add Gaussian noise  $\epsilon \sim \mathcal{N}(0, \gamma)$  and clip the results. We observe Figure 12 that the performance of our proposed method deteriorates as the noise level increases. Our proposed method is sensitive to noisy labels. By comparing the results to values in Table 2, we conclude that when  $\gamma \leq 0.1$ , our proposed method gives better or comparable MIG metrics than the competing methods.

## Conclusion

In this paper, we investigate the disentanglement learning problem, assuming a user introduces weak supervision by providing similarities between instances based on a factor to be disentangled. The similarity is provided as either a discrete (yes/no) or real-valued label between 0 and 1, where



(a) We fix  $\eta_2 = 2$  and vary  $\eta_1$ . (b) We fix  $\eta_1 = 1e3$  and vary  $\eta_2$ .

Figure 10: The Plot of MIG metrics versus  $\eta_1$  and  $\eta_2$ . When  $\eta_1 \geq 1e3$  and  $\eta_2 \geq 1.$ , these hyperparameters have limited effect on the MIG metrics for all datasets. In other experiments, we fix  $\eta_1 = 1e3$  and  $\eta_2 = 2$ .

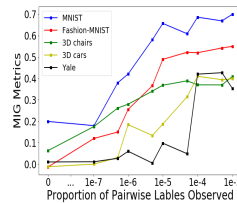


Figure 11: The plot of MIG metrics versus the proportion of pairwise labels observed. In general, with more pairwise labels observed, the disentanglement performance improves. In the experiments, we choose the proportion to be  $1e-4$  (0.01%).

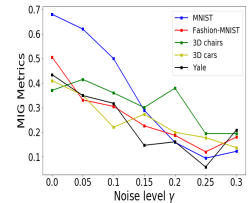


Figure 12: The plot of MIG metrics versus noise level  $\gamma$ . The performance of our proposed method deteriorates as the noise level increases. Our proposed method is sensitive to noisy labels.

a larger value indicates a stronger similarity. We propose a new formulation for weakly supervised disentanglement of latent variables within the Variational Auto-Encoder (VAE) framework. Experimental results on both benchmark and real-world datasets demonstrate that utilizing weak supervision improves the performance of VAE in disentanglement learning tasks.

## Acknowledgments

This work was partially supported by NIH Award Number 1R01HL141813-01, NSF 1839332 Tripod+X, and SAP SE. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. We were also grateful for the computational resources provided by Pittsburgh SuperComputing grant number TG-ASC170024.



## References

- [Aubry et al. 2014] Aubry, M.; Maturana, D.; Efros, A. A.; Russell, B. C.; and Sivic, J. 2014. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3762–3769.
- [Bengio, Courville, and Vincent 2013] Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35(8):1798–1828.
- [Bouchacourt, Tomioka, and Nowozin 2018] Bouchacourt, D.; Tomioka, R.; and Nowozin, S. 2018. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [Casale et al. 2018] Casale, F. P.; Dalca, A.; Saglietti, L.; Listgarten, J.; and Fusi, N. 2018. Gaussian process prior variational autoencoders. In *Advances in Neural Information Processing Systems*, 10369–10380.
- [Chen et al. 2018] Chen, T. Q.; Li, X.; Grosse, R. B.; and Duvenaud, D. K. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2610–2620.
- [Chen, Denoyer, and Artières 2018] Chen, M.; Denoyer, L.; and Artières, T. 2018. Multi-view data generation without view supervision. In *International Conference on Learning Representations*.
- [Feng et al. 2018] Feng, Z.; Wang, X.; Ke, C.; Zeng, A.-X.; Tao, D.; and Song, M. 2018. Dual swap disentangling. In *Advances in Neural Information Processing Systems*, 5894–5904.
- [Georghiades, Belhumeur, and Kriegman 2001] Georghiades, A. S.; Belhumeur, P. N.; and Kriegman, D. J. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (6):643–660.
- [Higgins et al. 2017] Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3.
- [Kim and Mnih 2018] Kim, H., and Mnih, A. 2018. Disentangling by factorising. In *International Conference on Machine Learning*, 2654–2663.
- [Kingma and Welling 2013] Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- [Krause et al. 2013] Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 554–561.
- [Kulkarni et al. 2015] Kulkarni, T. D.; Whitney, W. F.; Kohli, P.; and Tenenbaum, J. 2015. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, 2539–2547.
- [LeCun and Cortes 2010] LeCun, Y., and Cortes, C. 2010. MNIST handwritten digit database.
- [Locatello et al. 2018] Locatello, F.; Bauer, S.; Lucic, M.; Gelly, S.; Schölkopf, B.; and Bachem, O. 2018. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*.
- [Lopez et al. 2018] Lopez, R.; Regier, J.; Jordan, M. I.; and Yosef, N. 2018. Information constraints on auto-encoding variational bayes. In *Advances in Neural Information Processing Systems*, 6114–6125.
- [Narayanaswamy et al. 2017] Narayanaswamy, S.; Paige, T. B.; Van de Meent, J.-W.; Desmaison, A.; Goodman, N.; Kohli, P.; Wood, F.; and Torr, P. 2017. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, 5925–5935.
- [Quanjer et al. 2012] Quanjer, P. H.; Stanojevic, S.; Cole, T. J.; Baur, X.; Hall, G. L.; Culver, B. H.; Enright, P. L.; Hankinson, J. L.; Ip, M. S.; Zheng, J.; et al. 2012. Multi-ethnic reference values for spirometry for the 3–95-yr age range: the global lung function 2012 equations.
- [Vestbo et al. 2013] Vestbo, J.; Hurd, S. S.; Agustí, A. G.; Jones, P. W.; Vogelmeier, C.; Anzueto, A.; Barnes, P. J.; Fabbri, L. M.; Martinez, F. J.; Nishimura, M.; et al. 2013. Global strategy for the diagnosis, management, and prevention of chronic obstructive pulmonary disease: Gold executive summary. *American journal of respiratory and critical care medicine* 187(4):347–365.
- [Xiao, Rasul, and Vollgraf 2017] Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.



# Supplementary Material

## 1 The Plots in the Latent Space

**Fashion-MNIST dataset and 3D cars** We include the plots  $\mathbf{z}^{(u)}$  for the held-out instances and the generative instances by manipulating  $\mathbf{z}^{(u)}$ , for fashion-MNIST and 3D cars-datasets. We observe that although it is noisy, most of the instances are well separation according to the ground-truth factors. This implies that  $\mathbf{z}^{(u)}$  can explain the ground-truth factors.

It is not visually clear whether  $\mathbf{z}^{(u)}$  explains other factors of variations else than the ground-truth factors, as shown in Figure 1. In Figure 2b, we observe that the shape and color of the cars do not change in the rotations. In this example,  $\mathbf{z}^{(u)}$  does not seem to explain other factors other than the ground-truth factor.

**Yale Faces dataset** Since we choose  $d^{(u)} = 1$  for the Yale dataset, in Figure 4c, we plot  $\mathbf{z}^{(u)}$  against one dimension in  $\mathbf{z}^{(v)}$ . we observe that most instances with different ground-truth values can be separated according to  $\mathbf{z}^{(u)}$ . We manipulate  $\mathbf{z}^{(u)}$  and one dimension  $\mathbf{z}^{(v)}$  and plot the generated images in Figure 4e. We observe that the identify of the face changes when we change  $\mathbf{z}^{(v)}$ . However, when we change  $\mathbf{z}^{(u)}$ , the azimuth lighting changes, but the identity does not change. This suggests good disentangling results for this dataset.

## 2 Reconstructed and Generated Images

In Figure 4, we plot some reconstructed and generated images. The first row of each of the subplots are the original images in the held-out data. The second row shows the reconstructed image. On the right-hand side of these subplots, we show the reconstructed images by manipulating the latent variables  $\mathbf{z}^{(u)}$  but keep  $\mathbf{z}^{(v)}$  constants. We observe that our proposed model does not always reconstruct the images perfectly, probably because the models have never observed the held-out instances. However, for the generated images on the right-hand side, we observe the factor of interest changes, but other factors of variations do not change as much. In particular, the factor of interest is similar for those images with the same  $\mathbf{z}^{(u)}$  values. This implies excellent disentangling performance.

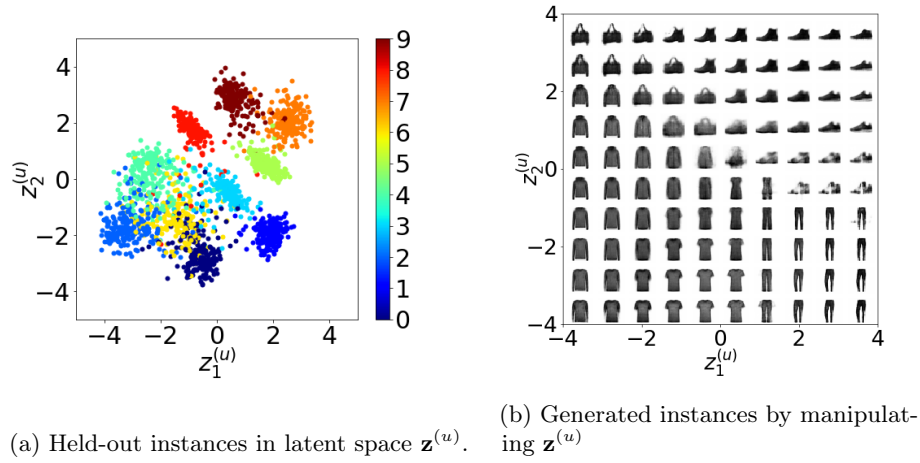


Figure 1: Fashion-MNIST dataset

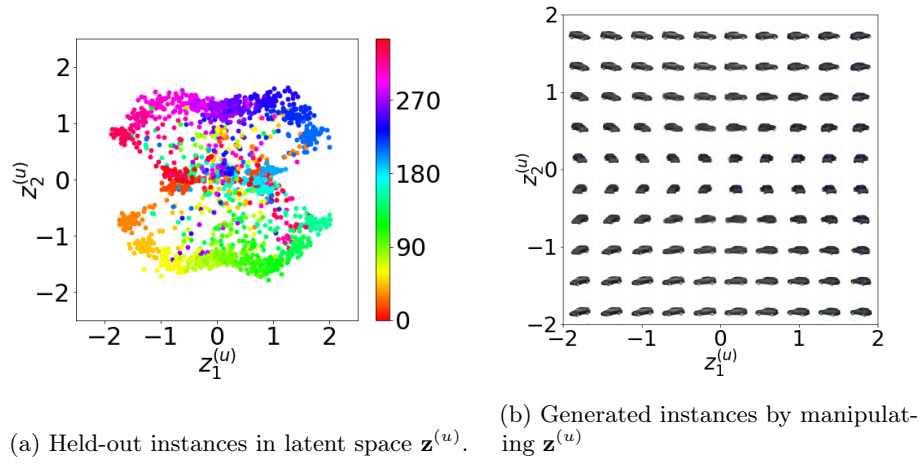


Figure 2: 3D cars dataset

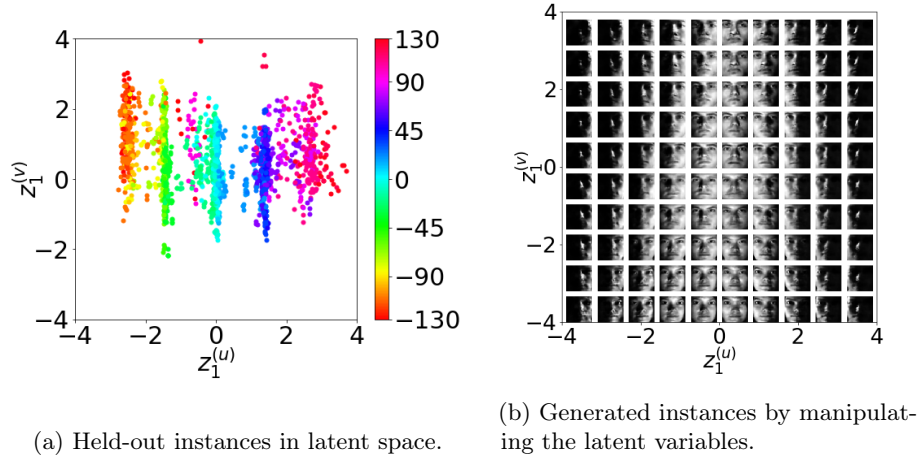
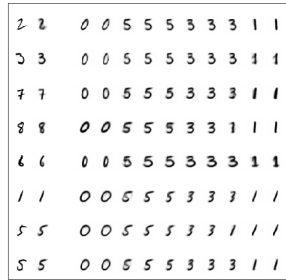
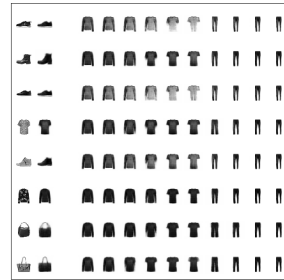


Figure 3: Yale Faces dataset



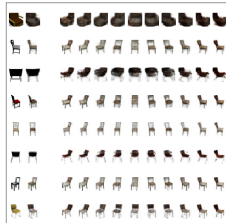
(a) MNIST dataset



(b) Fashion-MNIST dataset



(c) Yale faces Dataset



(d) 3D chairs dataset



(e) 3D cars dataset

Figure 4: Reconstructed and Generated Images