RECONSTRUCTION FOR DISENTANGLEMENT, CONTRAST FOR INVARIANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Disentangled and invariant representations are two critical goals of representation learning and many approaches have been proposed to achieve one of them. However, those two goals are actually complementary to each other and we propose a framework to accomplish both of them simultaneously. We introduce a weakly supervised signal to learn disentangled representation while using contrastive method to enforce representation invariance. Experimental evaluation shows that the proposed method outperforms state-of-the-art methods on a number of standard benchmarks.

1 INTRODUCTION

Deep neural networks (DNN) has achieved astonishing success in many computer vision tasks, such as image classification (Deng et al., 2009), image generation (Goodfellow et al., 2014) and face recognition (Schroff et al., 2015). In both prediction and generation tasks, learning to encode input data x into a lower dimensional representation z is the critical first step that facilities down-steam tasks (He et al., 2015; Kingma & Welling, 2014; van Steenkiste et al., 2019). For robust representation learning, *increasing generality* and *preventing overfitting* are two fundamental challenges. Typically, a DNN learns to encode representation which contains all factors of variation of data, such as pose, expression, illumination, and angle for face recognition, as well as other nuisance factors which may not have semantic meanings. Disentangled representation learning and invariant representation learning are often used to address these challenges.

For disentangled representation learning, Bengio et al. (2014) define a disentangled representation z, where a change in a given dimension z_i corresponds to a change in one and only one underlying factor of variation of the data. Although many unsupervised learning methods have been proposed (Higgins et al., 2017; Burgess et al., 2018; Kim & Mnih, 2018), Locatello et al. (2019) have shown both theoretically and empirically that the factor variants disentanglement is impossible without supervision or inductive bias. To this end, recent works adopted the concept of semi-supervised learning (Locatello et al., 2020b) and weakly supervised learning (Chen & Batmanghelich, 2020; Locatello et al., 2020a). On the other hand, Jaiswal et al. (2018) take an invariant representation learning perspective in which they split representation z into two parts $z = [z_p, z_n]$, where z_p only contains predictive related information, and z_n merely contains nuisance factors.

Invariant representation learning aims to learn to encode predictive latent factors which is invariant to nuisance factors in inputs (Xie et al., 2017; Jaiswal et al., 2018; Ganin et al., 2016; Louizos et al., 2016; Moyer et al., 2018; Sanchez et al., 2020b). By removing information of nuisance factors, invariant representation learning achieves good performance when facing challenges like adversarial attack (Chen et al., 2020) and out-of-distribution generalization (Arjovsky et al., 2020). Further, invariant representation learning has also been studied in the reinforcement learning settings (Hafner et al., 2019; Castro, 2020; Zhang et al., 2021).

Despite the success of both disentangled and invariant representation learning methods, the relation between the two has not been thoroughly investigated. As shown in Figure 1.a, invariant representation learning methods learn representations that maximize prediction accuracy, while leaving the representations of both known and unknown confounding factors entangled. Meanwhile, as illustrated in Figure 1.b, supervised disentangled representation methods cannot handle unknown nuisance factors, which may hurt downstream prediction tasks. Based on this observation, we seek

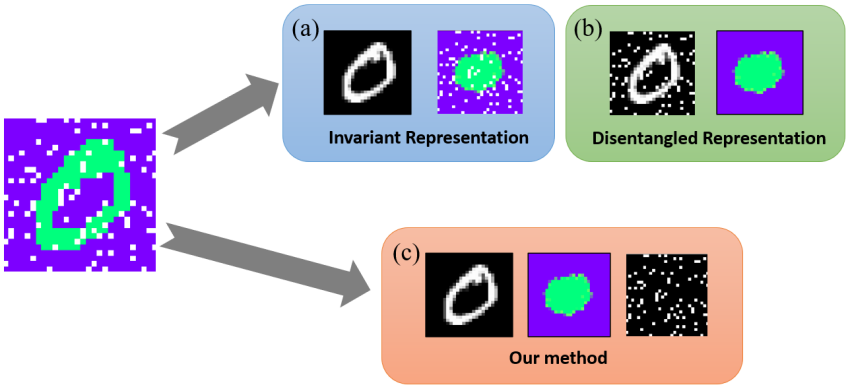


Figure 1: Given an image with nuisance factors, (a): invariant representation learning splits predictive factors from all nuisance factors; (b): disentangled representation learning splits the known nuisance factors but mixing predictive and unknown nuisance factors; (c): Our method splits all predictive, known nuisance and unknown nuisance factors simultaneously.

to achieve disentanglement and invariance of representation simultaneously and propose a new training framework. To split the known nuisance factors z_{nk} from predictive z_p and unknown nuisance factors z_{nu}, we introduce weak supervision signals to achieve disentangled representation learning. To make predictive factors z_p independent of all nuisance factors z_n, we introduce a new invariant regularizer via reconstruction. The predictive factors from the same class are further aligned through contrastive loss to enforce invariance. In summary our main contributions are:

- We extend and combine both disentangled and invariant representation learning and propose a novel approach to robust representation learning.
- We propose a novel approach that split the predictive, known nuisance factors and unknown nuisance factors. Mutual independence of those factors is achieved by the reconstruction step we use during training.
- Our new model outperforms state-of-the-art models on both disentangling and invariance tasks on standard benchmarks.

2 Related Work

Disentangled representation learning: Early works on disentangled representation learning aim at learning disentangled latent factors z by implementing an autoencoder framework (Higgins et al. (2017); Kim & Mnih (2018); Burgess et al. (2018)). Variational autoencoder (VAE) (Kingma & Welling, 2014) is the basic framework used in most state-of-the-art disentanglement learning methods. VAE uses DNN to map the high dimension input x to low dimension representation z. The latent representation z is then mapped to high dimension reconstruction $\hat{x}$. As shown in Equation (1), the overall objective function to train VAE is the evidence lower bounds (ELBO) of likelihood $\log p_\theta(x_1, x_2, ...x_n)$, which contains two parts: quality of reconstruction and Kullback-Leibler divergence (D_{KL}) between distribution $q_\phi(z|x)$ and the assumed prior $p(z)$. Then, VAE uses the negative of ELBO, $L_{VAE} = -ELBO$, as loss function to update the parameters in the model.

$$ELBO = \sum_{i=1}^{N} \left[\mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z)) \right] \leq \log p_\theta(x_1, x_2, ...x_n) \quad (1)$$

Advanced methods based on VAE improve the disentanglement performance by implementing new disentanglement regularization. **β-VAE** (Higgins et al., 2017) modified the original VAE by adding a hyper-parameter β to balance the weights of reconstruction loss and D_{KL}. When $\beta > 1$, the model gains stronger disentanglement regularization. **AnnealedVAE** (Burgess et al., 2018) further studied the effects of different value of β on reconstruction quality and disentangled representations. Based on its finding, **AnnealedVAE** implemented a dynamic algorithm to change the β from large to small value during training. **FactorVAE** (Kim & Mnih, 2018) proposes to use a discriminator in order to

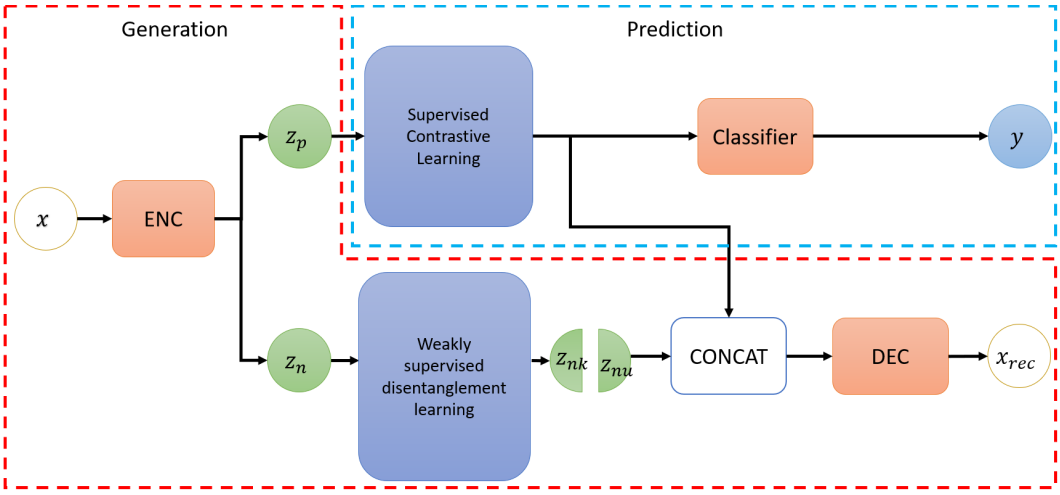


Figure 2: Architecture of the model. Red box is the generation part and the blue box is the prediction part

distinguish between the joint distribution of latent factors $q(z)$ and multiplication of marginal distribution of every latent factor $\prod q(z_i)$. By using the discriminator, **FactorVAE** finds a better trade-off between reconstruction quality and disentangled representation. Comparing to β**-VAE**, **DIP-VAE** (Kumar et al., 2018) adds another regularization $D(q_\phi(z)||p(z))$ between the marginal distribution of latent factors $q_\phi(z) = \int q_\phi(z|x)p(x)dx$ and the prior $p(z)$ to further encourage disentangled representation learning. D here can be any proper distance function. Chen et al. (2019) proposed β**-TCVAE** which decomposed the D_{KL} used in β**-VAE** into: total correlation, index-coded mutual information and dimension-wise KL divergence. During training, the quality of reconstruction and disentanglement are controlled by three different hyper-parameters applied on those three regularization. To overcome the challenge proposed by Locatello et al. (2019), **AdaVAE** (Locatello et al., 2020a) purposely choose pairs of inputs as supervision signal to learn representation disentanglement. Besides, Locatello et al. (2020a) proved theoretically that under some assumptions, the ideal representation disentanglement can be achieved without compromise.

Invariant Representation Learning: The methods that aim at learning invariant representation can be classified into two groups: those methods that require annotations of nuisance factors (Li et al., 2014; Louizos et al., 2016) and those that do not. A considerable number of approaches using nuisance factors annotations have been recently proposed. By implementing a regularizer which minimizes the Maximum Mean Discrepancy (MMD) (Gretton et al., 2007) on neural network (NN), The **NN+MMD** approach (Li et al., 2014) removes affects of nuisance from predictive factors. The Variational Fair Autoencoder (**VFAE**) (Louizos et al., 2016) uses special priors which encourage independence between nuisance factors and ideal invariant factors. Besides, **VFAE** also incorporates MMD as the regularizer to further remove any dependencies. The Controllable Adversarial Invariance (**CAI**) (Xie et al., 2017) approach applies the gradient reversal trick (Ganin et al., 2016) which penalizes the model if latent representation has information of nuisance factors. **CVIB** (Moyer et al., 2018) proposes a conditional form of Information Bottleneck (IB) and encourages the invariant representation learning by optimizing its variational bounds. Approaches that need annotations are suitable for removing specific affects of nuisance factors, such as race or gender bias in face recognition. However, due to the constrains of demanding annotations, those methods take more effort to pre-process the data and encounter challenges when the annotations are inaccurate or insufficient. Comparing to annotation-needed approaches, annotation-free methods are easier to be implemented in practice. The Unsupervised Adversarial Invariance (**UAI**) (Jaiswal et al., 2018) splits the latent factors into factors useful for prediction and nuisance factors. **UAI** encourages the independence of those two latent factors by incorporating competition between the prediction and the reconstruction objectives. Sanchez et al. (2020a) achieve invariant representation by using pairs of inputs and applying a neural network based mutual information estimator to minimize the mutual information between two shared representations. Furthermore, Sanchez et al. (2020a) employ a discriminator to distinguish the difference between shared representation and nuisance representation.

3 LEARNING DISENTANGLED AND INVARIANT REPRESENTATION

3.1 OVERVIEW OF MODEL ARCHITECTURE

As illustrated in Figure 2, the architecture of the proposed model contains two components: a generation module and a prediction module. Similar to VAE, the generation module performs an encoding-decoding task. However, it encodes the input x into latent factors z, where $z = [z_p, z_n]$, where z_p is the latent predictive factors that contains useful information for the prediction task, whereas z_n is the latent nuisance factors that can be further divided into known latent factors z_{nk} and unknown nuisance factors z_{nu}. z_{nk} are discovered and separated from z_n via weakly supervised disentangled representation learning, where the joint distribution $p(z_{nk}) = \prod_i p(z_{nk_i})$. Since z_n is the split containing nuisance factor, after z_{nk} is separated, the remaining factors of z_n naturally result in unknown nuisance factors z_{nu}. Then, z_p and z_n are concatenated for generating a reconstruction x_{rec} which are used to measure the quality of reconstruction and disentanglement degree. To enforce the independence between z_p and z_n, we add a regularizer using another reconstruction task, where the average mean and variance of predictive factors z_p are used to form new latent factors $\bar{z}_p$ as it will be discussed in Section 3.3. In the prediction module, we incorporate contrastive loss to cluster the predictive latent factors from the same class.

3.2 LEARNING INDEPENDENT KNOWN NUISANCE FACTORS z_{nk}

The known nuisance factors z_{nk} are discovered and separated from z_n, where $p(z_{nk}) = \prod_i p(z_{nk_i})$, because nuisance information is expected to be present only in z_n. As illustrated in Figure 1, since the goal of such procedure is learning disentangled representation, we follow the evaluation protocol in previous disentangled representation works for estimating the performance (Higgins et al., 2017; Kim & Mnih, 2018; Kumar et al., 2018; Shu et al., 2020).

To fulfill the requirement of including supervision signal for disentangled representation learning as proven in (Locatello et al., 2019), we use selected pairs of inputs $x^{(l)}$ and $x^{(m)}$ as supervision signals, where only a few common generative factors are shared. As illustrated in Figure 3, during training, the network encodes a pair of inputs $x^{(l)}$ and $x^{(m)}$ into two latent factors $z^{(l)} = [z_p^{(l)}, z_n^{(l)}]$ and $z^{(m)} = [z_p^{(m)}, z_n^{(m)}]$ respectively, which are then decoded to reconstruct $x_{rec}^{(l)}$ and $x_{rec}^{(m)}$. To encourage representation disentanglement, certain elements of $z_n^{(l)}$ and $z_n^{(m)}$ are *detected and swapped* to generate two new corresponding latent factors $\hat{z}^{(l)}$ and $\hat{z}^{(m)}$. The two new latent factors are then decoded to new reconstructions $\hat{x}_{rec}^{(l)}$ and $\hat{x}_{rec}^{(m)}$. By comparing $\hat{x}_{rec}$ with x_{rec}, the known nuisance factors z_{nk} are discovered and the elements of z_{nk} are enforced to be disentangled with each other.

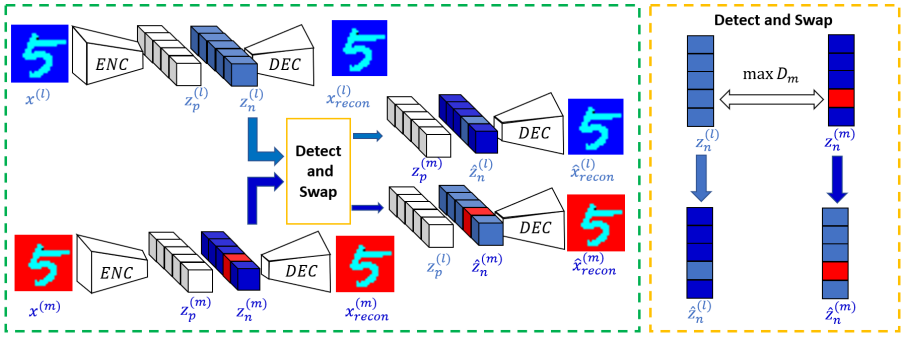
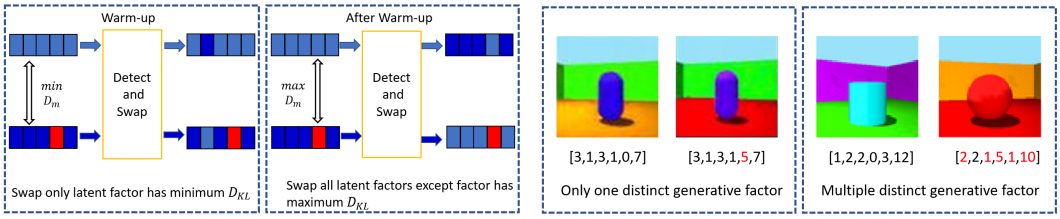


Figure 3: Disentanglement representation learning for known nuisance factors z_{nk}

Selecting image pairs for training and latent factors assumptions: As mentioned by Higgins et al. (2017), the true world simulator using generative factors to generate x can be modeled as: $p(x|v, w) = Sim(v, w)$, where v is the generative factors and w is other confounding factors. Inspired by this, we choose pairs of image by randomly selecting some generative factors to be the same and keeping the value of other generative factors to be random. As the image pair example in Figure 4b indicates, each image x has corresponding generative factors v, and the training



(a) In early training stages, small number of latent factors are swapped. the number of latent factors to be swapped increases gradually.

(b) In early training stages, input pair with small number of generative factors are chosen. The number of generative factors increases gradually.

Figure 4: Two training strategies for learning known nuisance factors z_{nk}

pair is generated as follows: we first randomly select a sample $x^{(l)}$ whose generative factors are $v^{(l)} = [v_1, v_2, ...v_n]$. We then randomly change the value of k elements in $v^{(l)}$ to form a new generative factors $v^{(m)}$ and choose another sample $x^{(m)}$ according to $v^{(m)}$. During training, indices of different generative factors between $v^{(l)}$ and $v^{(m)}$, and the groundtruth value of all generative factors are not available to the model. The model is weakly supervised since it is trained with only the knowledge of the number of factors k that have changed. Ideally, if the model can learn a disentangled representation, the model will encode the image pair $x^{(l)}$ and $x^{(m)}$ to the corresponding representations $z^{(l)}$ and $z^{(m)}$ which have the characteristic shown in Equation (2). We annotate the set of all different elements between $z^{(l)}$ and $z^{(m)}$ to be df_z and set of all latent factors to be d_z such that $df_z \subseteq d_z$.

$$\begin{aligned} p(z_{n_j}^{(l)}|\mathbf{x^{(l)}}) &= p(z_{n_j}^{(m)}|\mathbf{x^{(m)}});\ j \notin df_z \\ p(z_{n_i}^{(l)}|\mathbf{x^{(l)}}) &\neq p(z_{n_i}^{(m)}|\mathbf{x^{(m)}});\ i \in df_z \end{aligned} \tag{2}$$

Detecting and Swapping the distinct latent factors In VAE, it is commonly assumed that the posterior distribution of latent factors is a factorized multivariate Gaussian, $p(z|x) = q_\theta(z|x)$ (Kingma & Welling, 2014) and the reparameterization trick is used to make the posterior differentiable. By this assumption, we can directly measure the mutual information between the corresponding dimensions of the two latent representations $z^{(l)}$ and $z^{(m)}$ by measuring the divergence (D_m), which can be either KL divergence (D_{KL}) or Jensen-Shannon divergence (JSD). We show the process of detecting distinct latent factors in Equations (3) and (4), where a larger value of D_{KL} or JSD implies higher difference between the two corresponding latent factor distributions.

$$D_{KL}(q_\phi(z_{n_i}^{(l)}|x^{(l)})||q_\phi(z_{n_i}^{(m)}|x^{(m)})) = \frac{(\sigma_{n_i}^{(l)})^2 + (\mu_{n_i}^{(l)} - \mu_{n_i}^{(m)})^2}{2(\sigma_{n_i}^{(m)})^2} + \log(\frac{\sigma_{n_i}^{(m)}}{\sigma_{n_i}^{(l)}}) - \frac{1}{2} \tag{3}$$

$$\begin{aligned} JSD(q_\phi(z_{n_i}^{(l)}|x^{(l)})||q_\phi(z_{n_i}^{(m)}|x^{(m)})) &= \frac{1}{2}\big(D_{KL}(q_\phi(z_{n_i}^{(l)}|x^{(l)})||M) + D_{KL}(q_\phi(z_{n_i}^{(m)}|x^{(m)})||M)\big) \\ where \quad M &= \frac{1}{2}(q_\phi(z_{n_i}^{(l)}|x^{(l)}) + q_\phi(z_{n_i}^{(m)}|x^{(m)})) \end{aligned} \tag{4}$$

Since the model only has the knowledge of the number of different generative factors k, we swap all corresponding dimension elements of $z_n^{(l)}$ and $z_n^{(m)}$ except the top k highest D_m value elements. We incorporate this swapping step to create two new latent representations $\hat{z}_n^{(l)}$ and $\hat{z}_n^{(m)}$ shown in Equation (5).

$$\begin{aligned} \hat{z}_{n_i}^{(l)} &= z_{n_i}^{(m)}; \hat{z}_{n_i}^{(m)} = z_{n_i}^{(l)};\ i \notin df_z \\ \hat{z}_{n_j}^{(l)} &= z_{n_j}^{(l)}; \hat{z}_{n_j}^{(m)} = z_{n_j}^{(m)};\ j \in df_z \end{aligned} \tag{5}$$

Disentangled representation loss function: After we obtain $\hat{z}_n^{(l)}$ and $\hat{z}_n^{(m)}$, they are concatenated with $z_p^{(m)}$ and $z_p^{(l)}$, respectively, to generate two new latent representations $\hat{z}^{(l)} = [z_p^{(m)}, \hat{z}_n^{(l)}]$ and $\hat{z}^{(m)} = [z_p^{(l)}, \hat{z}_n^{(m)}]$. $\hat{z}^{(l)}$ and $\hat{z}^{(m)}$ are decoded into new reconstructions $\hat{x}^{(l)}$ and $\hat{x}^{(m)}$. Reminding

that there are only k different generative factors between pair of images, ideally, if the model perfectly detects the top k most different latent factors, by swapping other latent factors except them, the new representations $\hat{z}^{(l)}$ and $\hat{z}^{(m)}$ are same with the original representations $z^{(l)}$ and $z^{(m)}$. Accordingly, the new reconstructions $\hat{x}^{(l)}_{rec}$ and $\hat{x}^{(m)}_{rec}$ should be identical to the original reconstruction $x^{(l)}_{rec}$ and $x^{(m)}_{rec}$. Therefore, we design the disentangled representation loss in Equation (6), where D can be any suitable distance function *e.g*, mean square error (MSE) or binary cross-entropy (BCE).

$$L = L_{VAE}(x^{(l)}_{rec}, z^{(l)}) + L_{VAE}(x^{(m)}_{rec}, z^{(m)}) + D(\hat{x}^{(l)}_{rec}, x^{(l)}_{rec}) + D(\hat{x}^{(m)}_{rec}, x^{(m)}_{rec}) \tag{6}$$

Training Strategies for disentangled representation learning: To further improve the performance of disentangled representation learning, we design two strategies: *warmup by amount* and *warmup by difficulty*. Remembering that in swapping step, the model needs to swap $|d_z| - k$ elements of latent representations. Thus, in early stages of training, exchanging too many latent factors will easily lead to mistakes. Therefore, in the first strategy, we gradually increase the number of latent factors being swapped from 1 to $|d_z| - k$ during training. Further, to smoothly increase the training difficulty, we set the number of different generative factors to be 1 in the beginning and increase the number of different generative factors as training progresses. Those two training strategies highly improve the performance of the disentanglement learning and they are illustrated in Figure 4.

3.3 Learning invariant predictive factors z_p

After we obtain the disentangled representation z_{nk}, the predictive factors z_p may still be entangled with z_{nu}. Therefore, we need to add other constraints to achieve fully invariant representation of z_p.

Making z_p independent of z_n: As proved by Locatello et al. (2019), supervision signals need to be introduced for disentangled representation. Similarly, the independence of z_p and z_n also needs the help from a supervision signals as we discuss in Appendix A.1. Luckily, for supervised training, a batch of samples naturally contains supervision signal. Similar to Equation (2), the distribution of the representations z_p should be same for same class and can be expressed as Equation (7) where $C(x^{(l)})$ means the class of sample $x^{(l)}$.

$$\begin{aligned} p(z_p^{(l)}|\mathbf{x^{(l)}}) = p(z_p^{(m)}|\mathbf{x^{(m)}});\ C(x^{(l)}) = C(x^{(m)}) \\ p(z_p^{(l)}|\mathbf{x^{(l)}}) \neq p(z_p^{(m)}|\mathbf{x^{(m)}});\ C(x^{(l)}) \neq C(x^{(m)}) \end{aligned} \tag{7}$$

Similar to the method we use for disentangled representation learning, we generate new latent representation $\bar{z}_p$ and its corresponding reconstruction $\bar{x}_{rec-p}$. Then, we enforce the disentanglement between z_p and z_n by comparing the new reconstruction $\bar{x}_{rec-p}$ and x_{rec}. In contrast to the swapping method mentioned in Section 3.2, since the samples batch used for training usually contains more than two (2) samples from the same class, the swapping method is hard to be implemented in this situation. Therefore, we generate the new latent representations $\bar{z}_p$ by calculating the average mean $\bar{\mu}_p$ and average variance $\bar{V}_p$ of the latent representations from the same class as shown in Equation (8).

$$\begin{aligned} \bar{z_p} = \mathcal{N}(\bar{\mu_p}, \bar{V_p});\ \bar{x}_{rec-p} = Decoder([\bar{z}_p, z_n]) \\ \bar{\mu_p} = \frac{1}{|C|}\sum \mu_p^{(i)}\ ;\ \bar{V_p} = \frac{1}{|C|}\sum V_p^i;\ where\ \forall i \in C \end{aligned} \tag{8}$$

We then generate the new reconstruction $\bar{x}_{rec-p}$ using the same decoder as in other reconstruction tasks and enforce the disentanglement of z_p and z_n by calculating the $D(x_{rec}, \bar{x}_{rec_p})$ and update the parameters of the model according to its gradient.

Constrastive feature alignment: To achieve invariant representation, we need to make sure the latent representation that is useful for prediction can also be clustered according to their corresponding classes. Even though the often used cross-entropy (CE) loss can accomplish similar goals, the direct goal of CE loss is to achieve logit-level alignment and change the representations distribution according to the logits, which does not guarantee the uniform distribution of features. Alternatively, we incorporate contrastive methods to assure that representation/feature alignment can be accomplished effectively (Wang & Liu, 2021).

Similar to Khosla et al. (2020), we use supervised contrastive loss to achieve feature alignment and cluster the representations z_p according to their classes as shown in Equation (9) where C is the set that contains samples from the same class and $y_p = y_i$.

$$\mathcal{L}_{sup} = \sum_{i \in I} \frac{-1}{|C|} \sum_{p \in C} \log \frac{\exp\left(z_i \cdot z_p / \tau\right)}{\sum\limits_{a \in A(i)} \exp\left(z_i \cdot z_a / \tau\right)} \tag{9}$$

The final loss function used to train the model after adding the standard cross-entropy(CE) loss to train the classifier, is given by Equation (10).

$$\begin{aligned} &L = L_{CE}(x, y) + L_{VAE} + \alpha L_{disentangle} + \beta L_{Sup} + \gamma L_{Z_p} \\ &L_{disentangle} = D(\hat{x}_{rec}^{(l)}, x_{rec}^{(l)}) + D(\hat{x}_{rec}^{(m)}, x_{rec}^{(m)});\ L_{Z_p} = D(\bar{x}_{rec-p}, x_{rec}) \end{aligned} \tag{10}$$

4 Experiments Evaluation

4.1 Benchmark Datasets, Baseline Models and Metrics

Since our model learns disentangled and invariant representation simultaneously, our method are evaluated using multiple metrics and datasets. We evaluate the performance of our model on following four (4) datasets with different underlying factor of variations.

- **Colored-MNIST** Colored-MNIST dataset is augmented version of MNIST (LeCun & Cortes, 2010) with two known nuisance factors: digit color and background color. During training, the background color is chosen from 3 colors and digit color is chosen from other 6 colors. In test, we set the background color into 3 new colors which is different from training set.
- **Rotation-Colored-MNIST** This dataset is an further augmented version of Colored-MNIST. The background color and digit color setting is the same with the Colored-MNIST and this dataset further contains digits rotated to angles $\theta \in \Theta_{train} = \{0, \pm 22.5, \pm 45\}$. For test data, the rotation angles for digit is set to $\theta \in \Theta_{test} = \{0, \pm 65, \pm 75\}$. The rotation angles are set to be unknown nuisance factors.
- **3dShapes** (Burgess & Kim, 2018) 3dShapes contains 480,000 RGB $64 \times 64 \times 3$ images and the whole dataset has 6 different generative factors: object shape, object scale, object color, wall color, floor color and scene orientation. We choose object shape (4 classes) as the prediction task and only half of object colors during training and the remaining half of object color is used to evaluate performance of invariant representation.
- **MPI3D** (Gondal et al., 2019) MPI3D is a real-world dataset contains 1,036,800 RGB images and the whole dataset has 7 generative factors: object color, object shape, object size, camera height, background color, horizontal degree of robotic arm, vertical degree of robotic arm. Like 3dShapes, we choose object shape (6 classes) as the prediction target and half of object colors for training.

Table 1: Test average and worst accuracy results on Colored-MNIST, 3dShapes and MPI3D. **Bold, Black**: best result

Models	Colored-MNIST		3dShapes		MPI3D	
	Avg Acc	Worst Acc	Avg Acc	Worst Acc	Avg Acc	Worst Acc
Baseline	0.9512	0.6617	0.9887	0.9689	0.9012	0.8789
β-VAE	0.9265	0.5879	0.9866	0.9577	0.8698	0.8489
VFAE	0.9312	0.6554	0.9772	0.9334	0.8669	0.8243
CAI	0.9356	0.6317	0.9762	0.9432	0.8663	0.8216
CVIB	0.9331	0.7012	0.9711	0.9446	0.8704	0.8561
UAI	0.9474	0.7425	0.9713	0.9521	0.8789	0.8301
Our model	**0.9796**	**0.9043**	**0.9852**	**0.9763**	**0.9132**	**0.8917**

Since our method aims at achieving both disentanglement and invariance at the same time, we categorize the baseline state-of-the-art models into two groups according their goals.

Table 2: Test average accuracy and worst accuracy results on Rotation-Colored-MNIST with different rotation angles. **Bold, Black**: best result

Models	Rotation-Colored-MNIST							
	Avg Acc	Worst Acc	Avg Acc	Worst Acc	Avg Acc	Worst Acc	Avg Acc	Worst Acc
	-75		-65		+65		+75	
Baseline	0.770	0.623	0.897	0.775	0.858	0.658	0.683	0.499
β-VAE	0.771	0.612	0.852	0.750	0.834	0.630	0.687	0.472
VFAE	0.722	0.589	0.858	0.744	0.841	0.646	0.717	0.48
CAI	0.749	0.593	0.865	0.773	0.842	0.678	0.647	0.429
CVIB	0.761	0.592	0.886	0.791	0.856	0.688	0.722	0.534
UAI	0.780	0.611	0.888	0.800	0.854	0.682	0.702	0.511
Our model	**0.810**	**0.753**	**0.908**	**0.857**	**0.873**	**0.823**	**0.732**	**0.633**

- **Representation Disentanglement:** The following state-of-the-art model are used to compare performance of disentangled representation — (1) β-VAE (Higgins et al., 2017), (2) AnnealedVAE (Burgess et al., 2018), (3) FactorVAE (Kim & Mnih, 2018), (4) DIP-VAE-I (Kumar et al., 2018), (5) DIP-VAE-II (Kumar et al., 2018), (6) β-TCVAE (Chen et al., 2019) and (7) Ada-VAE (Locatello et al., 2020a).
- **Representation Invariance:** The following state-of-the-art model are used for invariant representation comparisons: (1) VFAE (Louizos et al., 2016), (2) CAI (Xie et al., 2017), (3) CVIB (Moyer et al., 2018), and (4) UAI (Jaiswal et al., 2018).

We adopt the following metrics to evaluate the performance of disentangled representation. All metrics range from 0 to 1, where 1 indicates that the latent factors are fully disentangled — (1) **Mutual Information Gap (MIG)** (Chen et al., 2019) which evaluates the gap of top two highest mutual information between a latent factors and generative factors. (2) **Separated Attribute Predictability (SAP)** (Kumar et al., 2018) which measures the mean of the difference of perdition error between the top two most predictive latent factors. (3) **Interventional Robustness Score (IRS)** (Suter et al., 2019) which evaluates reliance of a latent factor solely on generative factor regardless of other generative factors. (4) **FactorVAE (FVAE) score** (Kim & Mnih, 2018) which implements a majority vote classifier to predict the index of a fixed generative factor and take the prediction accuracy as the final score value. (5) **DCI-Disentanglement (DCI)** (Eastwood & Williams, 2018) which calculates the entropy of the distribution obtained by normalizing among each dimension of the learned representation for predicting the value of a generative factor.

To evaluate the performance of invariant representation learning, the test accuracy is the metric to evaluate the performance of models. Furthermore, we record both average test accuracy and worst-case test accuracy which was suggested by Sagawa* et al. (2020). During the experiment, we found out that using Equation (10) directly does not guarantee good performance. This may be caused by inconsistent behavior of CE loss and supervised contrastive loss. Thus, we separately train the classifier using CE loss and use remaining part of total loss to train the rest of the model.

4.2 Comparison with Previous Work

We show invariance learning results which are the test average accuracy and worst accuracy in Tables 1 and 2. For Color-Rotation-MNIST dataset, since we rotate the test samples with $\theta \in \Theta_{test} = \{\pm 65, \pm 75\}$ and those angles are different with training rotation angles $\theta \in \Theta_{train} = \{0, \pm -22.5, \pm 45\}$, we record each test average accuracy and worst accuracy under each rotation angles. The baseline model is the regular CNN model with no extra components for representation invariance, and β-VAE in Tables 1 and 2 is the regular CNN model with β-VAE structure to do reconstruction at the same time. Our proposed model significantly outperforms prior work.

To compare the performance of disentanglement, we show the results of disentanglement metrics on different datasets in Table 3. For divergence measurement D_m for latent factor, we test both D_{KL} and JSD, and choose the best results. As shown, our approach surpass both unsupervised and weakly-supervised method in almost all metrics.

Table 3: Disentanglement metrics on 3dShapes and MPI3D. **Bold, Black**: best result

Models	3dShapes					MPI3D				
	MIG	**SAP**	**IRS**	**FVAE**	**DCI**	**MIG**	**SAP**	**IRS**	**FVAE**	**DCI**
Unsupervised Disentanglement Leanring										
β-VAE	0.194	0.063	0.473	0.847	0.246	0.135	0.071	0.579	0.369	0.317
AnnealedVAE	0.233	0.087	0.545	0.864	0.341	0.098	0.038	0.490	0.397	0.228
FactorVAE	0.224	0.0440	0.630	0.792	0.304	0.092	0.031	0.529	0.379	0.164
DIP-VAE-I	0.143	0.026	0.491	0.761	0.137	0.104	0.073	0.476	0.491	0.223
DIP-VAE-II	0.137	0.020	0.424	0.742	0.083	0.131	0.075	0.509	0.544	0.244
β-TCVAE	0.364	0.096	0.594	0.970	0.601	0.189	0.146	**0.636**	0.430	0.322
β-FactorTCVAE	0.071	0.021	0.496	0.612	0.131	0.066	0.034	0.493	0.464	0.217
Weakly-Supervised Disentanglement Learning										
Ada-ML-VAE	0.509	0.127	0.620	0.996	0.940	0.240	0.074	0.576	0.476	0.285
Ada-GVAE	0.569	0.150	0.708	0.996	**0.946**	0.269	0.215	0.604	0.589	0.401
Our model	**0.716**	**0.156**	**0.784**	0.996	0.919	**0.486**	**0.225**	0.615	**0.565**	**0.560**

4.3 Ablation Study

Effectiveness of training strategies in disentanglement learning: To prove the effectiveness of the training strategies illustrated in Figure 4, we compare results of three situations: (1) none of those strategies is used, (2) only *warmup by amount* strategy is used, and (3) both strategies are used. As shown in Table 4, using both training strategies clearly outperforms the others.

Separately training the classifier and the rest of the model: To prove the importance of the two-step training as mentioned in Section 4.1, we compare the results of training the entire model together versus separately training classifier and other parts. Further, we also record the results of our model which does not use contrastive loss for feature-level alignment. As shown in Table 5, either only using CE loss (L_{CE}) or training the whole together ($L_{CE} + L_{contrastive}$) will harm the performance of the model. By separately training the classifier and other parts ($L_{CE} \leftrightarrow L_{contrastive}$), the framework has the best results for both average and worst accuracy.

Table 4: Disentanglement metrics of with different training strategies applied to 3dShapes

warmup by amount	warmup by difficulty	MIG	SAP	IRS	FVAE	DCI
		0.492	0.096	0.661	0.902	0.697
✓		0.512	0.126	0.674	0.944	0.781
✓	✓	**0.716**	**0.156**	**0.784**	**0.996**	**0.919**

Table 5: Performance of different scheme on Colored-MNIST and Rotation-Colored-MNIST

Training Scheme	Colored-MNIST		Rotation-Colored-MNIST (65)	
	avg acc	**worst acc**	**avg acc**	**worst acc**
L_{CE}	0.932	0.680	0.821	0.653
$L_{CE} + L_{contrastive}$	0.935	0.732	0.842	0.678
$L_{CE} \leftrightarrow L_{contrastive}$	**0.980**	**0.904**	**0.873**	**0.823**

5 Conclusion

In this work, we extend the ideas of representation disentanglement and representation invariance by combining them to achieve both goals at the same time. Further, we propose a new framework for weakly supervised disentanglement representation learning and achieve better performance than state-of-the-art disentangled learning methods. By introducing contrastive loss and new invariant regularization loss, we make predictive factor z_p to be more invariant to nuisance and increase both average and worst accuracy on invariant learning tasks.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives, 2014.

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

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 10069–10076. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6564.

Jiawei Chen, Janusz Konrad, and Prakash Ishwar. A cyclically-trained adversarial network for invariant representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pp. 3393–3402. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPRW50498.2020.00399. URL https://openaccess.thecvf.com/content_CVPRW_2020/html/w47/Chen_A_Cyclically-Trained_Adversarial_Network_for_Invariant_Representation_Learning_CVPRW_2020_paper.html.

Junxiang Chen and Kayhan Batmanghelich. Weakly supervised disentanglement by pairwise similarities. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 3495–3502. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/5754.

Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Muhammad Waleed Gondal, Manuel Wüthrich, Đorđe Miladinović, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Arthur Gretton, Karsten M. Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J. Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. C. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems 19*, pp. 513–520. MIT Press, 2007.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Ayush Jaiswal, Rex Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Unsupervised adversarial invariance. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 5097–5107. Curran Associates, Inc., 2018.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *CoRR*, abs/2004.11362, 2020. URL https://arxiv.org/abs/2004.11362.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2649–2658, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/kim18b.html.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Abhishek Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *ICLR*, abs/1711.00848, 2018.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Yujia Li, Kevin Swersky, and Richard Zemel. Learning unbiased features. *arXiv preprint arXiv:1412.5244*, 2014.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations, 2019.

Francesco Locatello, Ben Poole, Gunnar Raetsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6348–6359. PMLR, 13–18 Jul 2020a. URL http://proceedings.mlr.press/v119/locatello20a.html.

Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. Disentangling factors of variations using few labels. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=SygagpEKwB.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zeme. The variational fair autoencoder. In *Proceedings of International Conference on Learning Representations*, 2016.

Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. Invariant representations without adversarial training. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 9102–9111. Curran Associates, Inc., 2018.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

E. Sanchez, M. Serrurier, and M. Ortner. Learning disentangled representations via mutual information estimation. In *ECCV*, 2020a.

Eduardo Hugo Sanchez, Mathieu Serrurier, and Mathias Ortner. Learning disentangled representations via mutual information estimation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 205–221, Cham, 2020b. Springer International Publishing. ISBN 978-3-030-58542-6.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015. doi: 10.1109/CVPR.2015.7298682.

Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgSwyBKvr.

Raphael Suter, Đorđe Miladinović, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness, 2019.

Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? *CoRR*, abs/1905.12506, 2019. URL http://arxiv.org/abs/1905.12506.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2495–2504, June 2021.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 585–596. Curran Associates, Inc., 2017.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

A APPENDIX

A.1 WHY RECONSTRUCTION JOB IS NECESSARY FOR INDEPENDENT PREDICTIVE FACTORS

This proof is largely dependent on proof used in Locatello et al. (2019) and the proof can be shown below:

By the assumption, we hope the latent factors can be separated into two part z_p and z_n. z_p and z_n are expected to be independent to each other. We have :

$$p(z) = p(z_p) \cdot p(z_n)$$

Thus, if we choose any latent factor z_i from z_p, it should be independent to any other latent factor z_j chosen from z_n. $z_i \perp\!\!\!\perp z_j$.

It can be claimed that there exists an infinite family of bijective functions $f : supp(z) \to supp(z)$ such that $\frac{\partial f_i(u)}{\partial u_j} \neq 0$ almost everywhere for all i in z_p and j from z_n (i.e., z and $f(z)$ are completely entangled) and $P(z \leq u) = P(f(z) \leq u)$ for all $u \in supp(z)$ (i.e., they have the same marginal distribution). Since the unsupervised method only has access to observations x and y, it hence cannot distinguish between the two equivalent generative models and thus has to be entangled to at least one of them.

We first choose two bijective functions $g_i(v_i) = P(z_i \leq v_i)$ and $g_j(v_j) = P(z_j \leq v_j)$. By construction $g(z) = g(z_i) \cdot g(z_j)$ is a 2-dimensional uniform distribution. Similarly, consider function $h_i(v_i) = \psi^{-1}(v_i)$ and $h_j(v_j) = \psi^{-1}(v_j)$. Where,

$\psi(\cdot)$ is the cumulative density function of standard normal distribution. By this further construction, the random variable $h(g(z))$ is a 2-dimensional standard normal distribution.

Let $A \in R^{2\times 2}$ be an arbitrary orthogonal matrix with $A_{km} \neq 0$ for all $k = 1, 2$ and $m = 1, 2$. An infinite family of such matrices can be constructed using a Householder transformation: Choose an arbitrary $\alpha \in (0, 0.5)$and consider the vector v with $v_1 = \sqrt{\alpha}$ and $v_2 = \sqrt{1-\alpha}$. By construction, we have $\mathbf{v}^T\mathbf{v} = 1$. Define the matrix $A = \mathbf{I}_2 - 2\mathbf{v}\mathbf{v}^T$. Furthermore, A is orthogonal since

$$A^T A = (\mathbf{I}_2 - 2\mathbf{v}\mathbf{v}^T)^T(\mathbf{I}_2 - 2\mathbf{v}\mathbf{v}^T) = \mathbf{I}_2 - 4\mathbf{v}\mathbf{v}^T + 4\mathbf{v}(\mathbf{v}^T\mathbf{v})\mathbf{v}^T = \mathbf{I}_2$$

.

Since A is orthogonal, it is invertible and thus defines a bijective linear operator. The random variable $Ah(g(z)) \in R^2$ is hence an independent, multivariate standard normal distribution since the covariance matrix $A^T A$ is equal to I_2.

Since h is bijective, it follows that $h^{-1}(Ah(g(z)))$is an independent 2-dimensional uniform distribution. Define the function $f : supp(z) \to supp(z)$

$$f(u) = g^{-1}(h^{-1}(Ah(g(u))))$$

and note that by definition $f(z)$ has the same marginal distribution as z under P,i.e.,$P(z \leq u) = P(f(z) \leq u)$for all u .Finally, for almost every $u \in supp(z)$, it holds that

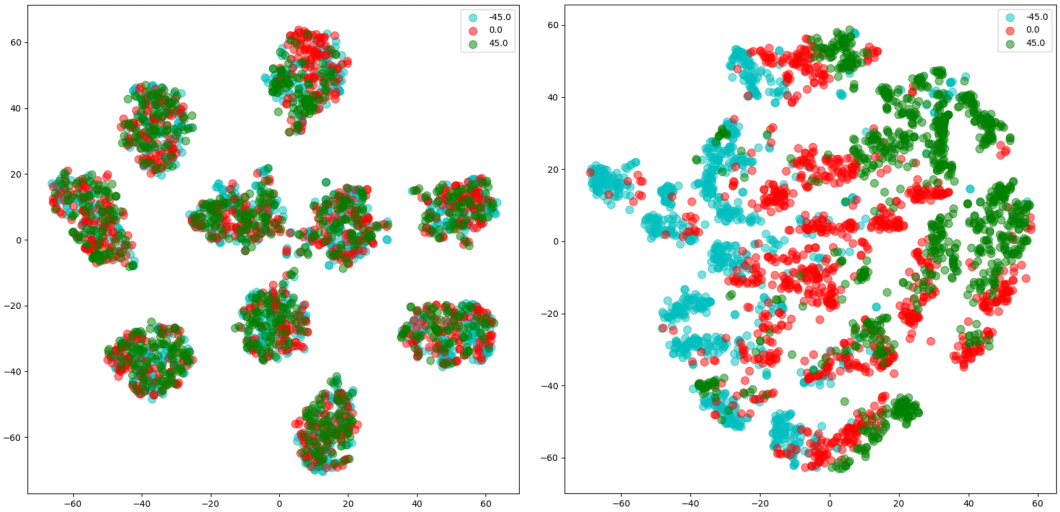
$$\frac{\partial f_i(u)}{\partial u_j} \neq 0,$$

Since A was chosen arbitrarily, there exists an infinite family of such function f.

To overcomes this problem, we need to do similar reconstruction job like we do for disentangled representation learning, where introducing supervision signal to enforce independence between z_p and z_n is necessary. However, previous works (Jaiswal et al., 2018; Xie et al., 2017; Louizos et al., 2016; Moyer et al., 2018) fail to do that. The detail of the process is described in Section 3.3.

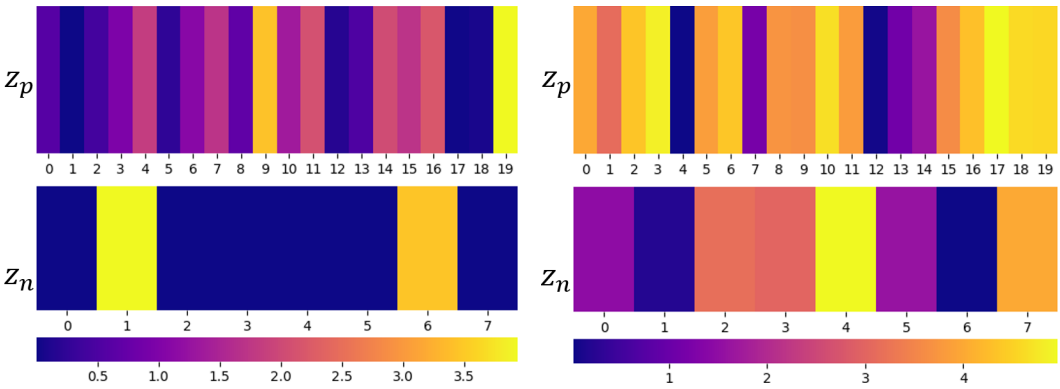
A.2 VISUALIZATION OF LATENT SPACE

To illustrate the performance of the results of the model, we visualize the latent representation by different methods. We first visualize the t-SNE results of z_p and z_{nu} which we tested on Color-Rotation-MNIST in Figure 5.



(a) z_p embedding (b) z_{nu} embedding

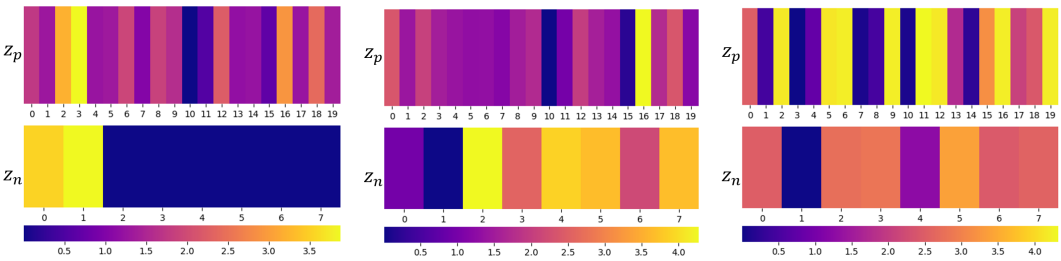
Figure 5: t-SNE visualization of z_p and z_{nu} embeddings of Color-Rotation-MNIST images colored by rotation angle. As desired, the z_p embedding does not encode rotation information, which migrates to z_{nu}.



(a) latent factors change by changing the background color and digit color

(b) latent factors change by changing the digit number and make the color unchanged

Figure 6: Heat map visualization for Colored-MNIST dataset



(a) latent factors change by changing the color

(b) latent factors change by changing the angle

(c) latent factors change by changing the angle

Figure 7: Heat map visualization for Rotation-Colored-MNIST dataset

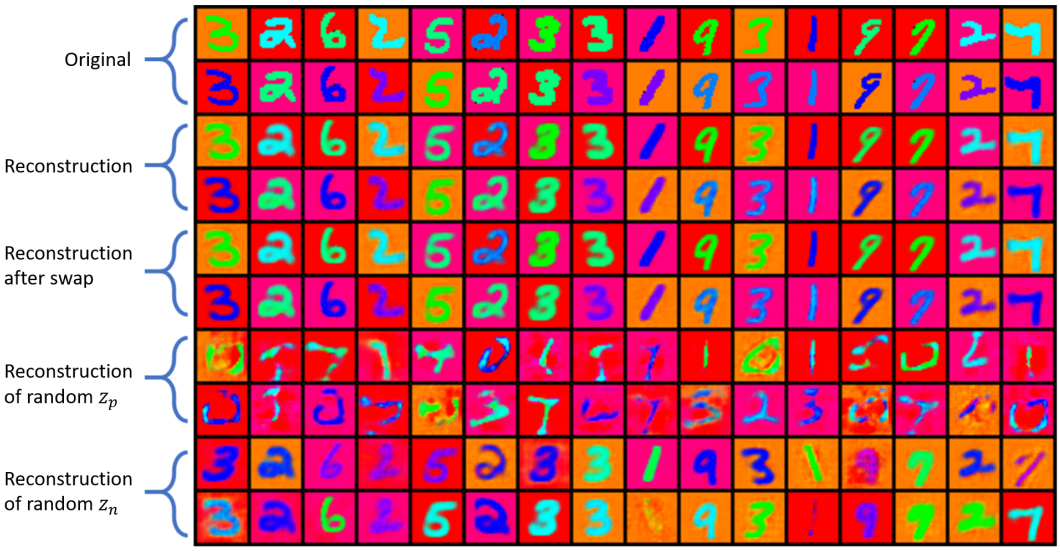


Figure 8: Reconstruction results of Colored-MNIST data

Further, we visualize the heatmap of latent space change by giving the model with different inputs. In Figure 6, we visualize the latent factors change of model tested on Colored-MNIST and visualize the latent factors change of model tested on Rotation-Colored-MNIST in Figure 7.

We finally visualize the results of reconstruction of model we tested on Colored-MNIST in Figure 8. Images in line 1-2 are original images used for training. Images in line 3-4 are reconstructions which are expected to be same with original inputs. Images in line 5-6 are reconstructions after swapping operation which are also expected to be same with original inputs and reconstruction in line 2-3. Images in line 7-8 are decoded from $[rand(z_p), z_n]$, where $rand(z_p)$ is normal random noise. Since we random sample z_p, the outputs of decoder should only contains the color information and unrecognized digits. In the contrary, images in line 9-10, we randomly sampled z_n and the latent factors for decoding is $[z_p, rand(z_n)]$. Therefore, images in line 9-10 should have same digit pattern with original inputs but have random digit color and background color.