Unsupervised Disentanglement Learning by intervention

Anonymous authors
Paper under double-blind review

Abstract

Recently there has been an increased interest in unsupervised learning of disentangled representations on the data generated from variation factors. Existing works rely on the assumption that the generative factors are independent despite this assumption is often violated in real-world scenarios. In this paper, we focus on the unsupervised learning of disentanglement in a general setting which the generative factors may be correlated. We propose an intervention-based framework to tackle this problem. In particular, first we apply a random intervention operation on a selected feature of the learnt image representation; then we propose a novel metric to measure the disentanglement by a downstream image translation task and prove it is consistent with existing ground-truth-required metrics experimentally; finally we design an end-to-end model to learn the disentangled representations with the self-supervision information from the downstream translation task. We evaluate our method on benchmark datasets quantitatively and give qualitative comparisons on a real-world dataset. Experiments show that our algorithm outperforms baselines on benchmark datasets when faced with correlated data and can disentangle semantic factors compared to baselines on real-world dataset.

1 Introduction

Learning disentangled representation from the data generated from variation factors gives interpretable insight on real-world applications such as face recognition, self-driving and explainable healthcare. The notion of disentangled representations was theoretically proposed in (Bengio et al. (2013)). One conceptually agreed definition is that the disentangled representation comprises a number of latent factors, with each factor controlling an interpretable aspect of the generated data (Bengio et al. (2013)). For example, in flower images, disentangled latent factors might control variations in color, shape, and background. Disentangled representations promise several advantages: better generalization ability (Higgins et al. (2017)), increased interpretability (Adel et al. (2018)), and faster learning on downstream tasks such as reasoning (van Steenkiste et al. (2019)).

Despite the recent growth of the field, most of the works of unsupervised disentanglement learning rely on the assumption that the generative factors are independent. One line of these works includes variants of Variational AutoEncoder(VAE)(Kingma & Welling (2013)). They directly minimize the total correlation of the features (Higgins et al. (2016); Chen et al. (2018); Kim & Mnih (2018); Kumar et al. (2018)). Another kind of disentanglement learning models, GAN-based models (Chen et al. (2016); Jeon et al. (2018); Lin et al. (2020)) are also restricted by independence assumption since they randomly sample the latent representation in the data generation process. However, independence assumption is often violated in real-world scenarios. For the images used for object classification, factors such as texture and color are confounded by the species of objects (e.g. stripes and black/white are correlated since they co-occur in the images of zebras). There are also recent works which use self-supervised learning to boost disentanglement (Zhu et al. (2020); Lin et al. (2020)). They propose to achieve an additional self-supervision information by self-supervised learning assuming independence assumption is satisfied.

The disentanglement performance of algorithms proposed under independence assumption may reduce when generative factors are correlated as in the work Träuble et al. (2020) shows. In this paper, we consider the unsupervised learning of disentanglement in a general setting which the generative factors of the data may be correlated. We development an intervention-based framework to address this problem. In particular, we define a random intervention operation which assigns a sampled value to one selected feature of the learnt image representation. Random intervention operation allows us to obtain an adjusted image representation which satisfies the selected feature is independent of the rest fixed features. To measure and improve disentanglement, we propose a novel metric by an elaborated downstream image translation task. A well disentangled representation may result in an relatively easy translation task and this translation task provides self-supervision information for our model. We prove the effectiveness of our novel metric experimentally and it correlates well with existing ground-truth-required metrics.

Our main contributions can be summarized as follows:
1) We address the unsupervised learning of disentanglement under a general setting that the independence assumption may be violated;
2) We propose an end-to-end framework to tackle the disentanglement, meanwhile we propose a novel metric and prove it is consistent with existed ground-truth-required metrics experimentally;
3) We evaluate our framework on benchmark datasets under independent/correlated factors assumption and compare the quality of disentanglement with baselines. The results show that our model outperforms baselines given correlated data. For the experiments on real-world dataset without ground truth factors, our model extracts semantic factors compared to baselines.

2 Related Work

2.1 Variational Autoencoders

There are plenty of works of learning disentangled features based on the VAE framework Kingma & Welling (2013). Most of them modify VAE structure to obtain disentangled features under independence assumption. Higgins *et al* proposed to reweight the KL-term in the learning objective of VAEs with a hyper-parameter $\beta > 1$ to encourage the model to learn independent features (β-VAE) (Higgins et al. (2016)). Kim *et al.* used a discriminator on the representation space and minimized total correlation (TC) among features by adversarial training (Kim & Mnih (2018)). Chen *et al.* extend β-VAE and estimated and optimized the TC term by employing a mini-batch weighted sampling. Other works improved the performance by making a specific design for discrete factors (Dupont (2018); Jeong & Song (2019)) and they also assumed independence in the latent space. The above works are built on the independence assumption and quantitatively evaluated on datasets generated from independent factors assumption despite the assumption may not be practical for real-world data. Recently Träuble *et al.* systematically induced correlations and found that the classic VAE-based models fall short of capturing ground-truth disentangled factors.

2.2 Generative Adversarial Networks

Models based on Generative Adversarial Networks(GAN) (Goodfellow et al. (2014)) have also been proposed to learn disentangled representations. Chen *et al.* first proposed InfoGAN, which learns disentangled representations by maximizing the mutual information between a subset of latent variables and the generated samples (Chen et al. (2016)). Lin *et al.* enhanced InfoGAN with a contrastive regularization motivated by the self-supervised learning (Lin et al. (2020)). There are also some other works based on GAN (Jeon et al. (2018),Liu et al. (2020)). Liu *et al.* add an orthogonal regularization to encourage independent representations. Hu *et al.* propose a GAN-based learning framework based on mixing operation(Hu et al. (2018)).

2.3 Self-Supervised Learning

Recent some works have been proposed to boost disentanglement learning by self-supervision learning (Lin et al. (2020); Zhu et al. (2020)). Zhu *et al.* defined variation predictability as how easy the following prediction tasks can be solved: given image pairs generated by latent codes varying in a single dimension, we predict which dimension is different. Zhu *et al.* show empirically that variation predictability and other disentanglement metrics are correlated. Lin *et al.* proposed a similar task of predicting the dimension index of the representations computed from a given image pair that only one dimension is the same in the representations. They try to add their proposed objectives to a basic disentangled learning model. A relative complex training strategy is needed (Lin et al. (2020))

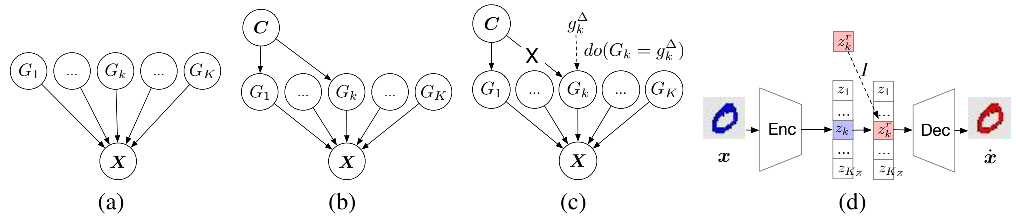

Figure 1: (a) The graphical model of data generation process under independence assumption; (b) The graphical model of the case when $\boldsymbol{C}$ is a confounder of the generative factors and (a) can be seen as a special case that $\boldsymbol{C} = \varnothing$; (c) The intervention operation $do(G_k = g_k^{\Delta})$ which cut offs all edges pointing to G_k and set its value to g_k^{Δ};(d) The intervention operation on representation $\boldsymbol{Z}$.

or the performance improvement may not be significant (Zhu et al. (2020)) or. Similarly, existing works based on self-supervised learning also rely on independence assumption.

3 Method

We use the notations following the work of Suter et al. (2019) to illustrate the problem of disentangling these factors in Fig.(1). We denote the generative factors of high dimensional observations $\boldsymbol{X}$ as $\boldsymbol{G}$. $\boldsymbol{G}$ includes K generative factors $\boldsymbol{G} = [G_1, G_2, ..., G_K]$ where each generative factor can have single or multi dimensions. We consider the multi-dimension case because some factors such as shape cannot be represented by one dimension. Then $\boldsymbol{X}$ is generated from $\boldsymbol{G}$. We denote $\boldsymbol{G} \backslash G_k$ as $\boldsymbol{G}_{\backslash k}$. The generative factors could be independent (Fig.(1(a))) or confounded by $\boldsymbol{C}$ (Fig.(1(b))). The latter situation will be more common in real-world. When $\boldsymbol{G}$ can be color, shape, texture and background of an object, they are confounded by the species $\boldsymbol{C}$. In fact, the independence situation is a special case when $\boldsymbol{C} = \varnothing$.

Our target is to learn a feature representation $\boldsymbol{Z} = [Z_1, Z_2, ..., Z_{K_Z}]$ matched with $\boldsymbol{G}$. We learn an encoder Enc to extract features $\boldsymbol{Z}$. We would hope for the ideal result that $K_Z = K$ and find one-to-one correspondence of Z_k and G_k. However, $\boldsymbol{G}$ is unobserved during training in unsupervised setting. So we will not assume to access the true value of K and set K_Z to be greater than K in real problem. Based on different assumptions on $\boldsymbol{G}$, corresponding restrictions will be applied on $\boldsymbol{Z}$. For example, the common independence assumption that $p(\boldsymbol{g}) = \prod_{k=1}^{K} p(g_k)$ requests $\boldsymbol{Z}$ also to be independent. Keep in mind that we use uppercase to represent variables (e.g. $\boldsymbol{G}$) and use lower case to represent the specific values corresponding to the variables (e.g. $\boldsymbol{g}$).

We use do-calculus to represent the intervention operation as the Fig.(1(c)) shows. $do(G_k = g_k^{\Delta})$ means that we assign a value g_k^{Δ} to variable G_k rather than passively observe $G_k = g_k^{\Delta}$. From the Proposition 1 of (Suter et al. (2019)), if one factor G_k is intervened, other factors will remain unchanged: $p(g_j | do(G_k = g_k^{\Delta})) = p(g_j), \forall g_k^{\Delta}, j$. As we cut off the relation between the possible $\boldsymbol{C}$ and the factor G_k by intervention as in Fig.(1(c)) and G_k is also separate from G_j. We extend G_j to $\boldsymbol{G}_{\backslash k} : p(\boldsymbol{g}_{\backslash k} | do(G_k = g_k^{\Delta})) = p(\boldsymbol{g}_{\backslash k})$.

In this paper we define our intervention process as follows: 1) Randomly sample an image $\boldsymbol{x}$ and select a factor G_k from its corresponding generative factors $\boldsymbol{g}$; 2) Randomly Sample a value g_k^{Δ} from the distribution of G_k; 3) Intervene on G_k (assign g_k^{Δ} to G_k) and generate a new image $\boldsymbol{x}(do(G_k = g_k^{\Delta}))$. We denote the distribution of $\boldsymbol{G}$ after such intervention operation as $\dot{p}_{\boldsymbol{G}}$ and

$$\dot{p}(\boldsymbol{g}) = \frac{1}{K} \sum_{k=1}^{K} p(\boldsymbol{g}_{\backslash k}) p(g_k). \tag{1}$$

Similarly, we denote the image distribution after intervention as $\dot{p}_{\boldsymbol{X}}$. Since intervening on G_k will not change other factors, we have the proposition as follows:

Proposition 1. *Suppose there is image $\boldsymbol{x}$ and the value on the factor G_k is g_k, we intervene G_k by $do(G_k = g_k^{\Delta})$ and get $\boldsymbol{x}(do(G_k = g_k^{\Delta}))$. If we further intervene $\boldsymbol{x}(do(G_k = g_k^{\Delta}))$ by $do(G_k = g_k)$, then we can get the original image $\boldsymbol{x}$: $\left[\boldsymbol{x}(do(G_k = g_k^{\Delta}))\right] (do(G_k = g_k)) = \boldsymbol{x}$.*

The proposition above encourages us to design a disentangling objective in a self-supervised manner. However it is still inefficient to obtain the disentangled representations as we may learn a trivial optimal solution. To overcome trivial optimization, we give two additional propositions considering a general factors setting (e.g. correlated data).

Assumption 1. *$\exists k \in \{1, ..., K\}$, $\forall \boldsymbol{g}$, $p(g_k|\boldsymbol{g}_{\setminus k})$ and $p(\boldsymbol{g}_{\setminus k}|g_k)$ are finite, and if $\boldsymbol{g}$ satisfies $p(\boldsymbol{g}_{\setminus k}) > 0$ and $p(g_k) > 0$, we get that $\frac{p(\boldsymbol{g})}{p(\boldsymbol{g}_{\setminus k})p(g_k)}$ is bounded by a finite value C.*

From the assumption 1 above, we assume that the distribution of g can't shrink to several finite cases, otherwise $p(g)$ may approaches to infinity when $p(\boldsymbol{g}_{\setminus k}) > 0$ and $p(g_k) > 0$.

Consider a special case that G_1,G_2,...,G_K are independent, $p_{\boldsymbol{X}}$ is exactly $\dot{p}_{\boldsymbol{X}}$ after intervention from Eq.(1). However, $\dot{p}_{\boldsymbol{X}}$ may be different from $p_{\boldsymbol{X}}$ when the independence assumption is violated. In this case, we propose to match the original image distribution $p_{\boldsymbol{X}}$ with a reweighted distribution of $\dot{p}_{\boldsymbol{X}}$.

Proposition 2. *Under the assumption 1, there exists a weight function W on $\boldsymbol{X}$: $w = W(x)$, s.t. $w \circ \dot{p}_{\boldsymbol{X}}(x) = p_{\boldsymbol{X}}(x)$.*

Definition 1 (Image Translation Task)**.** *Suppose the input, output and model of image translation task are U, V and M. For a learnt representation z, given a translation index k, we first sample two different values z_k^a and z_k^b from the distribution of Z_k and assign them to the feature z_k respectively, then we get a pair of images as in the left part of Fig.(2(c)) (u, v) in which $u = \dot{\boldsymbol{x}}(Z_k = z_k^a), v = \dot{\boldsymbol{x}}(Z_k = z_k^b)$. The task aims at predicting v given u as in the right part of Fig.(2(c)) by minimizing:*

$$L_{trans}(\theta_{\mathrm{M}}) = E\ crossentropy(v, M(u)). \tag{2}$$

If we obtain well disentangled representation $\boldsymbol{Z}$, we will achieve a relatively better translation performance for the task defined in definition 1 as the input images and output images in the dataset we obtained differ only in one specific feature (e.g. the pair of image (u, v) in Fig (2(c)), they are the same except for the color.) However, if the learnt representation $\boldsymbol{Z}$ is still entangled, the images in the pair (u, v) defined in the translation task will differ in more than one factors and there exists complex relationship between the features, and thus it may result in a relatively worse translation performance. Based on the analysis above, we give our proposition as follows:

Proposition 3. *The performance of the downstream translation task defined in definition 1 reflects the disentanglement quality of the learnt representation $\boldsymbol{Z}$.*

From Proposition 3, we propose a novel metric which measures the quality of the disentangled representation $\boldsymbol{Z}$ by a downstream image translation task. Compared to the metric proposed in the work of (Zhu et al. (2020)) which predicts the varied feature index given an image pair, our measurement is less likely to over fit and correlates well with existed ground-truth-required disentanglement metrics.

4 Model Structure and Optimization

We propose an autoencoder-based structure in Fig.(2). The dimension of each feature Z_k in the learnt representation Z is D_Z. We conduct the intervention operation on feature space $\boldsymbol{Z}$: 1). Sample two images $\boldsymbol{x}$, $\boldsymbol{x}^r$ and a feature index $k \in \{1, ..., K_Z\}$ independently, then we extract the representation $\boldsymbol{z} = \mathrm{Enc}(\boldsymbol{x})$ and $\boldsymbol{z}^r = \mathrm{Enc}(\boldsymbol{x}^r)$; 2). Intervention 1 ($I_1$ in Fig.(2(a))): change the k_{th} feature of $\boldsymbol{z}$ to z_k^r then get the reconstruction result as $\dot{\boldsymbol{x}}(Z_k = z_k^r)$. Our objectives together with our model structure are described as follows:

1. Reconstruction losses: the original image $\boldsymbol{x}$ is reconstructed from the representation $\boldsymbol{z}$: $\hat{\boldsymbol{x}} = \mathrm{Dec}(\boldsymbol{z})$ and the reconstruction loss function is $\mathcal{L}_{rec}(\theta_{\mathrm{Enc}}, \theta_{\mathrm{Dec}}) = E_x\ crossentropy(\boldsymbol{x}, \hat{\boldsymbol{x}})$ in which $crossentropy$ means crossentropy between the two images as shown in the upper part in Fig.(2(a)); according to Propositon 1, if we further extract the representation of $\dot{\boldsymbol{x}}(Z_k = z_k^r)$ and change Z_k back to z_k as shown I_2 in Fig.(2(a)), then after reconstruct the image as $\ddot{\boldsymbol{x}}(\dot{Z}_k = z_k)$, we hope $\ddot{\boldsymbol{x}}(\dot{Z}_k = z_k)$ is the same as $\boldsymbol{x}$ and the second reconstruction loss function $L_{in}(\theta_{\mathrm{Enc}}, \theta_{\mathrm{Dec}}) = E_{\boldsymbol{x},\boldsymbol{x}^r,k}\ crossentropy(\boldsymbol{x}, \ddot{\boldsymbol{x}})$.

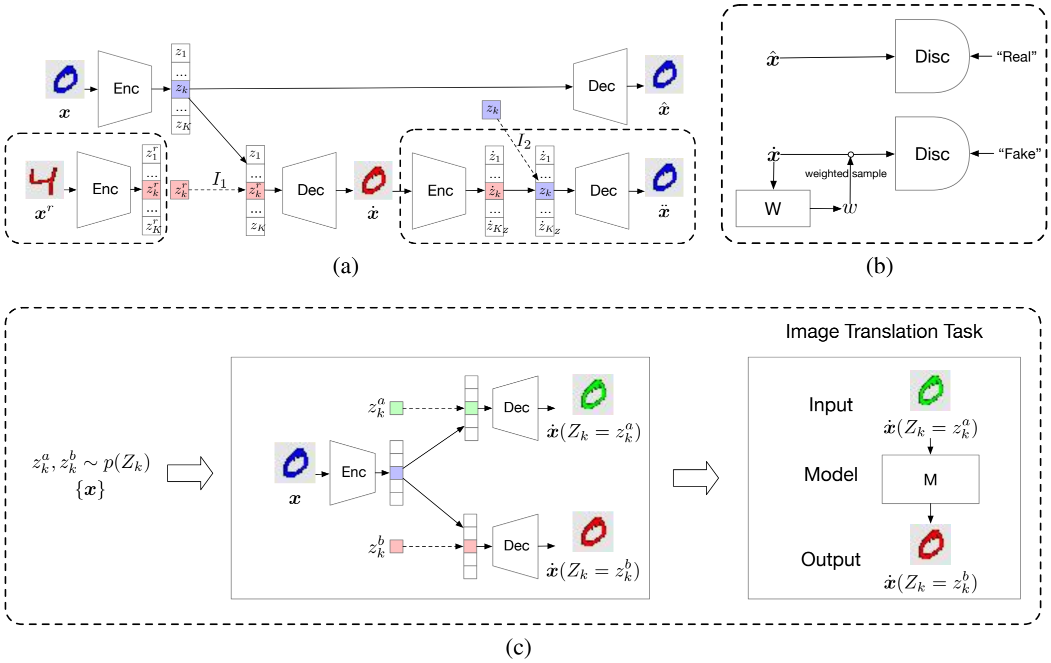

Figure 2: Overall structure of our model. (a) We do two interventions on the original image and make the reconstruct image meet Proposition 1. (b) We use a network W to calculate the weights of images from $\dot{p}_X$ and make its weighted distribution match p_X. (c) We construct image translations by intervention.

2. Adversarial loss: according to the Proposition 2, we can match the orignial image distribution p_X by learning a weight function W, thus we propose an adversarial loss $\min\limits_{\theta_{\text{Wt}},\theta_{\text{Enc}},\theta_{\text{Dec}}} \max\limits_{\theta_{\text{Disc}}} L_{adv}$ to optimize the module W in Fig.(2(b)) and L_{adv} is calculate as:

$$L_{adv}(\theta_{\text{Wt}}, \theta_{\text{Enc}}, \theta_{\text{Dec}}, \theta_{\text{Disc}}) = E_{\boldsymbol{x},\boldsymbol{x}^r,k} \ \log \text{Disc}(\hat{x}) + \frac{\text{W}(\dot{x})}{N} \log(1 - \text{Disc}(\dot{x}))$$

where N is for normalization and calculated as $E_{\boldsymbol{x},\boldsymbol{x}^r,k} \ \text{W}(\dot{x})$, we use the reconstructed image $\hat{\boldsymbol{x}}$ as the real image, Disc means the module discriminator.

3 Disentanglement loss measured by image translation: inspired by the work in (Zhu et al. (2020)) which argues that a prediction task could be used to measure the quality of unsupervised learning disentanglement. Instead of predicting the changed index of the representation which is vulnerable to hyperparameters and easy to over fit, we propose an image translation task to measure the quality of the disentanglement. Specifically, our novel metric which measures the difficulty of the translation task shown detailed in the Algorithm 1. As in Fig.(2(c)), we sample two features z_k^a and z_k^b to generate the dataset for input and output for our image translation task. The difficulty of the image translation task reflects the quality of our disentanglement, in which L_{trans} is defined in Eq.(1) and $L_{diff}(\theta_{\text{Enc}}, \theta_{\text{Dec}}) = E_{k,z_k^a,z_k^b,\mathcal{D},\tilde{\mathcal{D}}} \tilde{L}_{trans}$.

Finally, we train our model with the weighted sum of the components described above:

$$\min_{\theta_{\text{Wt}},\theta_{\text{Enc}},\theta_{\text{Dec}}} \max_{\theta_{\text{Disc}}} L_{rec} + \lambda_1 L_{in} + \lambda_2 L_{adv} + \lambda_3 L_{diff} \tag{3}$$

```
Algorithm 1: Calculation of Image Translation Loss
Input: $\theta_{\text{Enc}}$,$\theta_{\text{Dec}}$, $k$, two sampled feature $z_k^a$ and $z_k^b$ of from the representation distribution $p_Z$,
       two sampled mini-batches of originial images $\mathcal{D}$ and $\tilde{\mathcal{D}}$ for training and testing, learning
       step $s$ and learning rate $\alpha$
Generate training data $\{u_i, v_i\}$ with $k$, $z_k^a$, $z_k^b$ and images $\mathcal{D}$
Generate testing data $\{\tilde{u}_i, \tilde{v}_i\}$ with $k$, $z_k^a$, $z_k^b$ and images $\tilde{\mathcal{D}}$
for $i \leq s$ do
    Calculate $\nabla L_{trans}(\theta_M)$ on $\{u_i, v_i\}$
    Update $\theta_M$ with gradient descent: $\theta_M = \theta_M - \alpha \nabla L_{trans}(\theta_M)$
    $i = i + 1$
end
return $\tilde{L}_{trans}$ calculated with $\{\tilde{u}_i, \tilde{v}_i\}$ and model parameter $\theta_M$
```

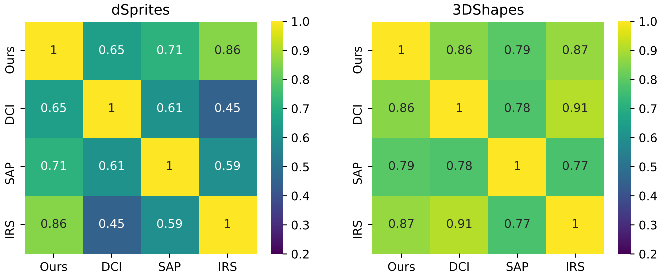

Figure 3: Spearman rank correlation of existed different disentanglement metrics and ours on two datasets. The first row/column is our self-supervised image-translation metric, which is highly correlated with all other disentanglement scores (row/column 2-4).

5 Experiments

5.1 Dataset and Evaluation

We run experiments on benchmark datasets with pre-defined generative factors 1) $\boldsymbol{dSprites}$: images of 2D shapes generated from five ground truth independent latent factors: shape (heart, oval and square)), x-position information (32 values), y-position information (32 values), scale (6 values) and rotation (40 values); 2) $\boldsymbol{3DShapes}$: RGB images of 3D shapes with ground truth factors: shape (4 values), scale(8 values), orientation(15 values), floor color(10 values), wall color(10 values) and object color(10 values); 3) $\boldsymbol{CMNIST}$: we generate a colored MNIST dataset with two generative factors: content and color; 4) $\boldsymbol{OxfordFlowers102}$: RGB images of flowers, a real-world dataset without ground-truth generative factors.

We evaluate our algorithm and baselines quantitatively on the first three datasets under two settings: independent/correlated generative factors. First, we hold out a part of the data as test set. Then we construct a training set with correlated factors by resampling on the rest of the data. For each dataset, the correlated factors are: x-position and scale(dSprites), floor color and wall color(3DShapes), color and content(CMNIST). Similarly, we build a training set with independent factors with the same size of the correlated training set. As the images in OxfordFlowers102 have no ground-truth generative factors, we therefore evaluate this dataset by querying: first, we select a query image. For each Z_k, we find out the nearest neighbours of the query images on space Z_k.

We choose the following baselines: FactorVAE(Kim & Mnih (2018)), β-TCVAE(Chen et al. (2018)) and InfoGAN-CR(Lin et al. (2020)). We evaluate unsupervised learning for disentanglement using the popular metrics: DCI(Eastwood & Williams (2018)), SAP(Kumar et al. (2018)) and IRS(Suter et al. (2019)). For CMNIST dataset, since it is hard to apply most of the baselines and metrics to because of multi-dimension factors. We only compare our algorithm with FactorVAE using the metric SAP. The details on evaluation metrics can also be found in the Appendix.

Table 1: Comparisons of the popular disentanglement metrics on the dSprites dataset

	Independent Factors			**Correlated Factors**		
Method	DCI	SAP	IRS	DCI	SAP	IRS
FactorVAE($\gamma = 10$)	0.73	0.55	0.62	0.60	0.46	0.54
FactorVAE($\gamma = 40$)	0.72	0.56	**0.63**	0.63	0.47	0.57
β-TCVAE	0.62	0.54	0.55	0.55	0.46	0.46
InfoGAN-CR	0.73	**0.58**	0.50	0.68	**0.51**	0.45
Ours	**0.76**	0.55	0.62	**0.74**	**0.51**	**0.59**

Table 2: Comparisons of the popular disentanglement metrics on the 3DShapes dataset

	Independent Factors			**Correlated Factors**		
Method	DCI	SAP	IRS	DCI	SAP	IRS
FactorVAE($\gamma = 10$)	0.75	0.43	0.65	0.60	0.33	0.57
FactorVAE($\gamma = 40$)	0.73	0.53	0.67	0.49	0.26	0.42
β-TCVAE	**0.76**	0.46	**0.70**	**0.68**	0.23	0.64
InfoGAN-CR	0.53	0.40	0.57	0.36	0.25	0.44
Ours	0.63	0.45	0.62	0.67	**0.47**	**0.66**

5.2 Implementation

We choose the network structures of baselines according to their papers. We implement our model with the similar structures of shared components (Encoder, Decoder) with FactorVAE. In dSprites and 3DShapes dataset, we follow the common setting of $K_Z = 10, D_Z = 1$. In CMNIST, we set $K_Z = 2, D_Z = 16$ and in OxfordFlowers102, we set $K_Z = 3, D_Z = 16$. Since some baselines are designed for one dimension of representation to a factor and difficult to apply when $D_Z > 1$. We only compare our algorithm with FactorVAE on CMNIST and OxfordFlowers102 datasets. For each model, we repeat 10 times and report the average performance. More details about implementation are shown in the Appendix.

Table 3: Comparisons of the popular disentanglement metrics on the CMNIST dataset

	Independent Factors	**Correlated Factors**
Method	SAP	SAP
FactorVAE($\gamma = 10$)	0.83 ± 0.02	0.41 ± 0.02
FactorVAE($\gamma = 40$)	0.72 ± 0.08	0.47 ± 0.11
Ours	0.74 ± 0.03	$\mathbf{0.70 \pm 0.02}$

5.3 Image Translation Task Difficulty as Disentanglement Metric

In this section, we evaluate the effectiveness of our proposed metric and prove that our novel metric correlates well with existed ground-truth-required metrics. For fair comparison, we run FactorVAE with different hyper-parameters, random seeds and factor relationships(independent/correlated) to cover a large range of model performance. Then we calculate various disentanglement metrics with ours of each model on the held-out test set. Finally, we compute the spearman rank correlation of these metrics on dSprites and 3DShapes datasets. Note that the two models are not trained without overlap objectives when using metrics. From the results in Fig. (3), our proposed self-supervised-based metrics correlates well with other ground-truth-required metrics, thus it can be used to select models, choose hyper-parameters, and act as a part of the model training objective.

5.4 Evaluation on disentanglement

The results on dSprites, Shapes3D and CMNIST are reported in Table 1, 2 and 3. Under independence assumption, our model achieves comparable performance in some cases. For the data generated by independent generative factors, factors can be disentangled by directly minimizing total correlation as existing algorithms do.

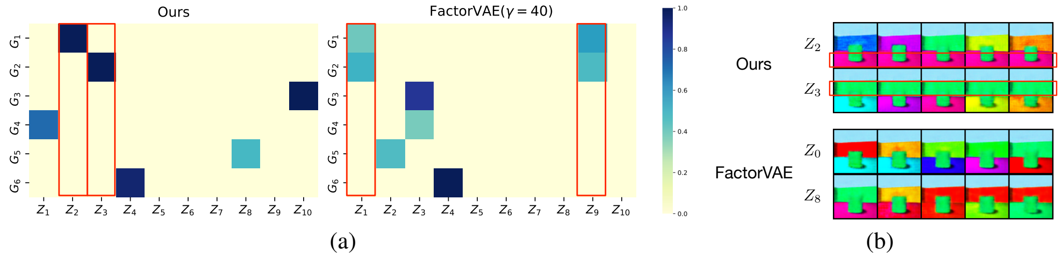

Figure 4: Qualitative comparison on 3DShapes under correlated factors. The results are from our model and FactorVAE($\gamma = 40$) respectively. (a) The importance of individual feature predicting the value of a given generative factor. (b)The result of latent traversal, we fix other features, change one feature from its 10%-quantile to 90%-quantile.)

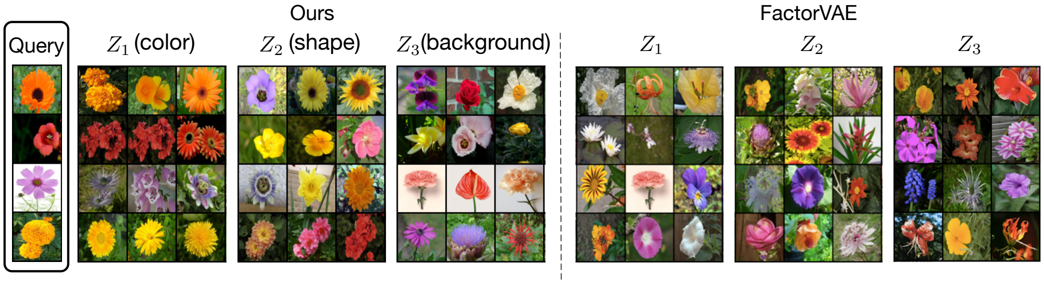

Figure 5: Qualitative comparison on OxfordFlowers102. The left column are query images. The rest are the nearest 3 neighbors of query images in feature space obtained by Ours and FactorVAE.

However, if the factors are correlated, our model proposed under this general setting outperforms baselines in almost all cases and has stable performances under independent/correlated settings.

We conduct a qualitative comparison between our algorithm and baselines on 3DShapes dataset. We define the correlated factors floor color and wall color as (G_1, G_2) and choose the models with best DCI obtained by our algorithm and FactorVAE ($\gamma = 40$). For the feature importance matrices shown in Fig. (4(a)), our model obtains well-disentangled features (highlight in red) while FactorVAE obtains entangled features of floor color and wall color. This is also supported by latent traversal visualization shown in Fig (4(b)) (highlight in red). In the first two rows in Fig (4(b)), the other factor remains fixed while changing the value of floor color feature or wall color feature in our learnt disentangled representation. In the last two rows shown the result of FactorVAE, if we just change one value of the learnt feature, the other factor can't remain fixed.

5.5 Qualitative evaluation on Real-world dataset OxfordFlowers102

As shown in Fig. (5), the query images are in the left column, then we present the 3-nearest neighbors of each query image measured by the distance on Z_k. Our model can learn semantic factors of flower color(Z_1), flower shape(Z_2) and background color(Z_3) while there is no semantic factors obtained by FactorVAE model in the right part of Fig. (5).

6 Conclusion

In this paper, we study the problem of unsupervised disentanglement learning under the situation where independence assumption may be violated. We propose to design an image translation task to measure the disentanglement and use our metric to provide self-supervision information in the learning framework. Our metric is robust to hyper-parameters and experiments show it correlates well with existed ground-truth-required metrics. We evaluate our framework on benchmark datasets and the results show that it outperforms baselines on correlated data. We also conduct experiments on a real-world dataset to verify that our framework can disentangle semantic factors compared to baselines.

REFERENCES

Tameem Adel, Zoubin Ghahramani, and Adrian Weller. Discovering interpretable representations for both deep generative and discriminative models. In *International Conference on Machine Learning*, pp. 50–59, 2018.

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

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 2610–2620, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pp. 710–720, 2018.

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Irina Higgins, Loic 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. 2016.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1480–1490, 2017.

Qiyang Hu, Attila Szabó, Tiziano Portenier, Paolo Favaro, and Matthias Zwicker. Disentangling factors of variation by mixing them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3399–3407, 2018.

Insu Jeon, Wonkwang Lee, and Gunhee Kim. Ib-gan: Disentangled representation learning with information bottleneck gan. 2018.

Yeonwoo Jeong and Hyun Oh Song. Learning discrete and continuous factors of data via alternating disentanglement. In *International Conference on Machine Learning*, pp. 3091–3099, 2019.

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

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

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

Zinan Lin, Kiran K Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. ICML, 2020.

Bingchen Liu, Yizhe Zhu, Zuohui Fu, Gerard de Melo, and Ahmed Elgammal. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization. In *AAAI*, pp. 4836–4843, 2020.

Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pp. 6056–6065. PMLR, 2019.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Anirudh Goyal, Francesco Locatello, Bernhard Schölkopf, and Stefan Bauer. Is independence all you need? on the generalization of representations learned from correlated data. *arXiv preprint arXiv:2006.07886*, 2020.

Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? In *Advances in Neural Information Processing Systems*, pp. 14245–14258, 2019.

Xinqi Zhu, Chang Xu, and Dacheng Tao. Learning disentangled representations with latent variation predictability. ECCV, 2020.

A Proof of Proposition 2

First, we can have the following lemma under Assumption 1:

Lemma 1. $\forall \boldsymbol{x} \in \mathrm{supp}(p_{\boldsymbol{X}}) : \dot{p}_{\boldsymbol{X}}(\boldsymbol{x}) > 0,$

where $\mathrm{supp}(p_{\boldsymbol{X}})$ means its support set. From the assumption 1, $\mathrm{supp}(p_{\boldsymbol{X}})$ is a subset of the image distribution after intervention. The proof of it is:

Proof. We have:

$$\begin{aligned}
\dot{p}_{\boldsymbol{X}}(\boldsymbol{x}) &= \int p(\boldsymbol{x}|\boldsymbol{g})\dot{p}_{\boldsymbol{G}}(\boldsymbol{g})d\boldsymbol{g} \\
&= \int p(\boldsymbol{x}|\boldsymbol{g})(\frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{g}_{\setminus k})p_{G_k}(g_k))d\boldsymbol{g} \\
&= \frac{1}{K}\sum_{k=1}^{K}\int p(\boldsymbol{x}|\boldsymbol{g})p(\boldsymbol{g}_{\setminus k})p_{G_k}(g_k)d\boldsymbol{g}
\end{aligned} \tag{4}$$

Suppose $\exists \boldsymbol{x'} \in \mathrm{supp}(p_{\boldsymbol{X}}), \dot{p}_{\boldsymbol{X}}(\boldsymbol{x'}) = 0$. Then $\forall k, \int p(\boldsymbol{x}'|\boldsymbol{g})p(\boldsymbol{g}_{\setminus k})p_{G_k}(g_k)d\boldsymbol{g} = 0$.

According to assumption 1, $\exists k', \forall x$, $p(g_{k'}|\boldsymbol{g}_{\setminus k'})$ and $p(\boldsymbol{g}_{\setminus k'}|g_{k'})$ are finite, we have when $p(g_{k'}) = 0, p(\boldsymbol{g}) = p(g_{k'})p(g_{k'}|\boldsymbol{g}_{\setminus k'}) = 0$ and when $p(\boldsymbol{g}_{\setminus k'}) = 0, p(\boldsymbol{g}) = 0$

We further have: if $p(\boldsymbol{g}_{\setminus k'}) > 0$ and $p(g_{k'}) > 0$, $\frac{p(\boldsymbol{g})}{p(\boldsymbol{g}_{\setminus k'})p(g_{k^{\text{`}}})} \leq C$. We have

$$\begin{aligned}
p_{\boldsymbol{X}}(\boldsymbol{x'}) &= \int p(\boldsymbol{x'}|\boldsymbol{g})p_{\boldsymbol{G}}(\boldsymbol{g})d\boldsymbol{g} \\
&= \int_{p(\boldsymbol{g}_{\setminus k'})>0, p(g_{k'})>0} p(x|\boldsymbol{g})p_{\boldsymbol{G}}(\boldsymbol{g})d\boldsymbol{g} \\
&\leq C\int_{p(\boldsymbol{g}_{\setminus k'})>0, p(g_{k'})>0} p(\boldsymbol{x}'|\boldsymbol{g})p(\boldsymbol{g}_{\setminus k'})p(g_{k'})d\boldsymbol{g} \\
&\leq C\int p(\boldsymbol{x}'|\boldsymbol{g})p(\boldsymbol{g}_{\setminus k})p_{G_k}(g_k)d\boldsymbol{g} = 0
\end{aligned} \tag{5}$$

which is contradict with $\boldsymbol{x'} \in \mathrm{supp}(p_{\boldsymbol{X}})$. So we can have $\forall \boldsymbol{x'} \in \mathrm{supp}(p_{\boldsymbol{X}}), \dot{p}_{\boldsymbol{X}}(\boldsymbol{x'}) > 0$.

With Lemma 1, let $w = W(\boldsymbol{x}) = \frac{p_{\boldsymbol{X}}(\boldsymbol{x})}{\dot{p}_{\boldsymbol{X}}(\boldsymbol{x})}$, we will have $w \circ \dot{p}_{\boldsymbol{X}}(\boldsymbol{x}) = p_{\boldsymbol{X}}(\boldsymbol{x})$.

□

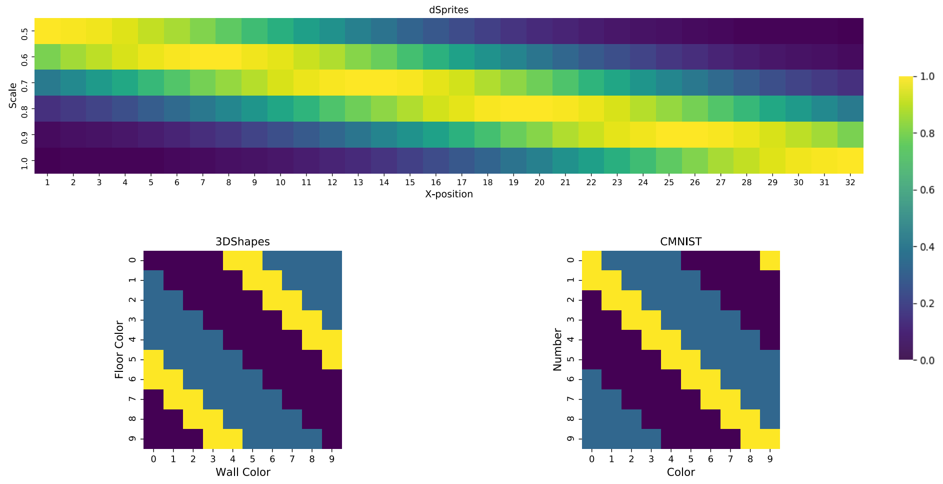

Figure 6: Joint distribution of correlated factor on each dataset. We normalize the highest probability as 1.

B Data Processing

For dSprites dataset, we first hold out a test set with 50,000 samples. Then we resample on the rest of data with joint distribution of x-position and scale as the top of Fig. (6) to construct a training set with correlated factors. The size of this training set is 200,000. We also resample a training set whose size is 200,000 with independence factors. Similar operations have also been conducted on 3DShapes. The size of training set and test set are 100,000 and 50,000. For CMNIST, we paint on the original training and test set of MNIST dataset respectively with ten different colors. Each kind of color is corresponding to that in 3DShapes dataset. The joint distribution of correlated factors are shown in Fig (6)

C Network Structure

In almost all experiments, we used a convolutional neural network for the encoders of all autoencoder models (Factor VAE, β-TCVAE and our model), discriminators of InfoGAN-CR model and our model, weight network of our model. We used a deconvolutional neural network for the decoders of all autoencoder models (Factor VAE, β-TCVAE and our model) and generator of InfoGAN-CR model. We used a multi-layer perceptron for total correlation discriminator of FactorVAE. One exception is that for dSprites dataset, we use multi-layer perceptrons for encoder and decoder of β-TCVAE, according to the authors' public implementation. All network structure of our model are shown from Table 4 to 9.

Encoder	**Decoder**
Input 64 $\times$ 64 $\times$ n_c image	Input $\in \mathbb{R}^{10}$
4 $\times$ 4 conv. 32 ReLU. stride 2	FC. 128 ReLU.
4 $\times$ 4 conv. 32 ReLU. stride 2.	FC. 4 $\times$ 4 $\times$ 64 ReLU.
4 $\times$ 4 conv. 64 ReLU. stride 2.	4 $\times$ 4 upconv. 64 ReLU. stride 2.
4 $\times$ 4 conv. 64 ReLU. stride 2.	4 $\times$ 4 upconv. 32 ReLU. stride 2.
FC. 128.	4 $\times$ 4 upconv. 32 ReLU. stride 2.
FC. 10.	4 $\times$ 4 upconv. $\times$ n_c. stride 2 Sigmoid

Table 4: Encoder and Decoder network architectures of our model for dSprites($n_c = 1$) and 3DShapes ($n_c = 3$) experiments.

We used the Adam optimizer for all updates. For baselines, learning rates are selected according to the original papers. Since our dataset is constructed with sampling, the total number of training

Discriminator	**Weight Network**
Input $64 \times 64 \times n_c$ image	Input $64 \times 64 \times n_c$ image
4×4 conv. 32 ReLU. stride 2	4×4 conv. 32 ReLU. stride 2
4×4 conv. 32 ReLU. stride 2.	4×4 conv. 32 ReLU. stride 2.
4×4 conv. 64 ReLU. stride 2.	4×4 conv. 64 ReLU. stride 2.
4×4 conv. 64 ReLU. stride 2.	4×4 conv. 64 ReLU. stride 2.
FC. 128.	FC. 128.
FC. 1 Sigmoid.	FC. 1 Exp.

Table 5: Discriminator and Weight Network architectures of our model for dSprites($n_c = 1$) and 3DShapes ($n_c = 3$) experiments.

Encoder	**Decoder**
Input $28 \times 28 \times 3$ image	Input $\in \mathbb{R}^{32}$
4×4 conv. 64 ReLU. stride 2	FC. $4 \times 4 \times 64$ ReLU.
4×4 conv. 128 ReLU. stride 2.	4×4 upconv. 64 ReLU. stride 2.
3×3 conv. 128 ReLU. stride 1.	4×4 upconv. $\times$ 3. stride 2 Sigmoid
FC. 32.	

Table 6: Encoder and Decoder network architectures of our model for CMNIST experiments.

samples are less than that of the whole dataset. We run each baseline within a closed iteration to make sure the algorithm converages. We use a batch size of 64 for all experiments except for β-TCVAE. Since it need a large batch size for estimation of total correlation, we use a batch size of 2048 as the authors used. The learning rate of Encoder, Decoder and Discriminator of our model is 0.0001. The learning rate of Weight Network is 0.00001. We fix λ_3 to be 1. First, we do not include L_{diff} in the objective function and choose λ_1 and λ_2 with L_{diff} as metric. We get $\lambda_1 = 0.25, \lambda_2 = 10.0$ for dSprites and 3DShapes dataset, $\lambda_1 = 1.0, \lambda_2 = 10.0$ for CMNIST and $\lambda_1 = 1.0, \lambda_2 = 100.0$ for OxfordFlowers102. After determing λ_1 and λ_2, we train a model with the objective function with L_{diff}.

D Disentanglement Evaluation

The details of calculation of disentanglement metrics used in this paper can be referred to their original works. We clarify the selection of some involved models and hyperparameters in this paper.

In the calculation of SAP, we use a logistic regression model as the classifier of discrete factors.

When we compute the disentanglement metrics of (Eastwood & Williams (2018)) (DCI), we follow the work of (Lin et al. (2020)). Random forest regressor implemented in the scikit-learn library is chosen as the regressor. For dSprites experiments, default values for all parameters are used, except for the max-depth parameter. We use the values: 4, 2, 4, 2, and 2, for the latent factors: shape, scale, rotation, x-position, and y-position respectively, as used by the IB-GAN paper (Dupont (2018)). For 3DShapes experiments, we use the cross-validation strategy to search for the max-depth of random forest regressor.

Discriminator	**Weight Network**
Input $28 \times 28 \times 3$ image	Input $28 \times 28 \times 3$ image
4×4 conv. 64 ReLU. stride 2	4×4 conv. 64 ReLU. stride 2
4×4 conv. 128 ReLU. stride 2.	4×4 conv. 128 ReLU. stride 2.
3×3 conv. 128 ReLU. stride 1.	3×3 conv. 128 ReLU. stride 1.
FC. 128.	FC. 128.
FC. 1 Sigmoid.	FC. 1 Exp.

Table 7: Discriminator and Weight Network architectures of our model for CMNIST experiments.

Encoder	**Decoder**
Input $64 \times 64 \times 3$ image	Input $\in \mathbb{R}^{48}$
4×4 conv. 128 ReLU. stride 2	FC. 512 ReLU.
4×4 conv. 128 ReLU. stride 2.	FC. $4 \times 4 \times 256$ ReLU.
4×4 conv. 256 ReLU. stride 2.	4×4 upconv. 256 ReLU. stride 2.
4×4 conv. 256 ReLU. stride 2.	4×4 upconv. 128 ReLU. stride 2.
FC. 512.	4×4 upconv. 128 ReLU. stride 2.
FC. 48.	4×4 upconv. $\times$ 3. stride 2 Sigmoid

Table 8: Encoder and Decoder network architectures of our model for OxfordFlowers10 experiment.

Discriminator	**Weight Network**
Input $64 \times 64 \times 3$ image	Input $64 \times 64 \times n_c$ image
4×4 conv. 64 LReLU. stride 2	4×4 conv. 64 LReLU. stride 2
4×4 conv. 64 LReLU. stride 2.	4×4 conv. 64 LReLU. stride 2.
4×4 conv. 128 LReLU. stride 2.	4×4 conv. 128 LReLU. stride 2.
4×4 conv. 128 LReLU. stride 2.	4×4 conv. 128 LReLU. stride 2.
FC. 256.	FC. 256.
FC. 256.	FC. 256.
FC. 1 Sigmoid.	FC. 1 Exp.

Table 9: Discriminator and Weight Network architectures of our model for OxfordFlowers10 experiment.