Rethinking Content and Style: Exploring Bias for Unsupervised Disentanglement
Appendix

A. Implementation Details

All the models are trained on an Nvidia Tesla V100 GPU. The model is implemented in PyTorch (Paszke et al., 2017). For Car3D, Chairs, and Celeba, we set the size of the style embedding d_s to 256, the size of the content embedding d_c to 128, and the epoch to 200. All the images are resized to 64×64. For the hyperparameters, $w_P = 5, w_{IB} = 1, w_{ID} = 1$. We used Adam (Kingma & Ba, 2015) with a learning rate of 0.0003 for the models and 0.003 for the latent spaces.

A.1. Experimental Details

Content Transfer Metric. For Car3D and Chairs datasets, the content labels (ground truth) are available. For images I_i and I_j sampled from the testing set randomly, we compute L-PIPS (Zhang et al., 2018) between $G_\theta(c_i, s_j)$ and the corresponding ground truth from the same class of I_j. For CelebA, we randomly sample two images I_i and I_j with the same identity, and we retrieve an image I_k that has the most similar pose to I_j from the test set, i.e., the nearest neighbor in the 68 facial-landmarks space. We measure the similarity between I_j and $G_\theta(c_k, s_i)$.

Classification Metric. Following Jha et al. (2018), we train two models of a single fully-connected layer to classify content labels from style embeddings and classify style labels from content embeddings.

A.2. Network Structure

Our `Single C-S DisMo framework` and `Multiple C-S DisMo framework` are shown in Figure 1. For the details of the reparametric module R, please refer to Appendix C.4.

B. Baseline Details

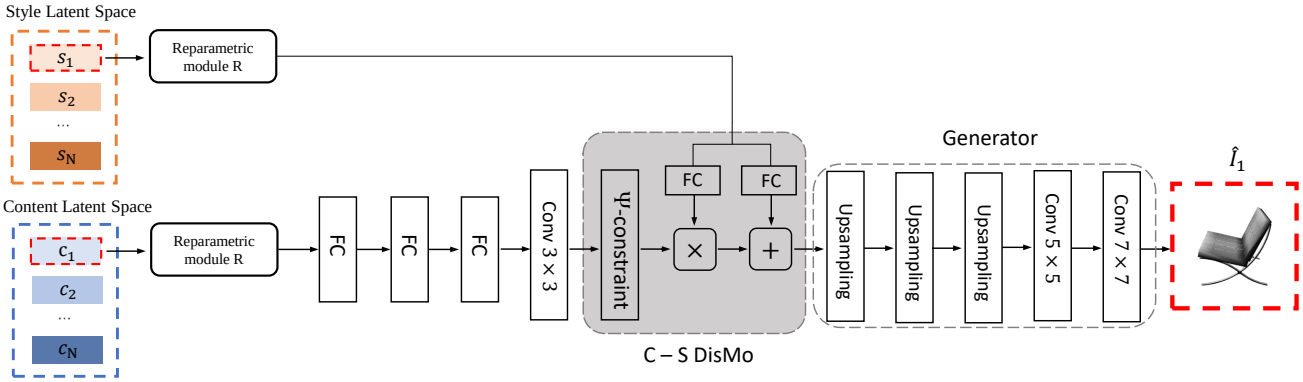
For the datasets in the main paper, Car3D contains 183 car models, each rendered from 96 poses. Chairs consists of 1393 chair models, each rendered from 62 poses. CelebA contains 202,599 facial images of 10,177 celebrities.

For the baselines, we use open-source implementations for Cycle-VAE (Jha et al., 2018) [1], DrNet (Denton & Birodkar, 2017) [2], Lord (Gabbay & Hoshen, 2020) [3] and FactorVAE (Kim & Mnih, 2018) [4].

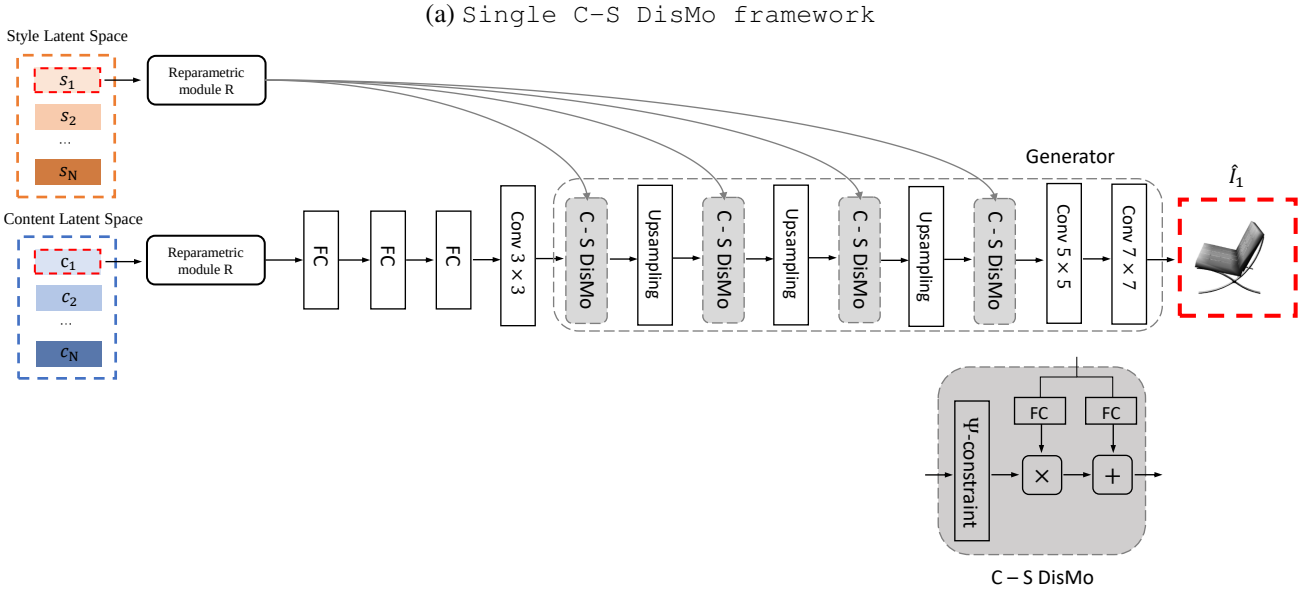
For FactorVAE, we traverse the latent space to select the dimensions related to pose as content embedding and treat the other dimensions as style embedding. For Wu et al. (2019b), there is no open-source implementation. We use the code from `https://github.com/CompVis/vunet`, which uses ground truth landmarks as input instead of learning the landmarks unsupervisedly. To achieve the pseudo ground truth landmarks, we use the face detection library (Bulat & Tzimiropoulos, 2017) for Celeba. We try to use the L1 and perceptual loss for all the baselines and select the best.

We split the datasets into training and testing sets. For Celeba, we randomly select 1000 among 10177 celebrities for testing. For Car3D, we randomly select 20 among 183 CAD models for testing. For Chairs, we randomly select 100 among 1393 models for testing. For baselines with group supervision, only the training sets are used for training. For unsupervised baselines and our method, all the datasets are used for training.

[1] `https://github.com/ananyahjha93/cycle-consistent-vae`
[2] `https://github.com/ap229997/DRNET`
[3] `https://github.com/avivga/lord-pytorch`
[4] `https://github.com/1Konny/FactorVAE`



(a) Single C-S DisMo framework



(b) Multiple C-S DisMo framework

Figure 1. Details of network structure. For every upsampling layer, there is a 3×3 convolutional layer following it. The embedding s_1 and c_1 from style and content embedding space respectively are first processed by the reparametric model R, then fed into the network in different ways. Then the image $\hat{I}_1$ is generated. The style and content latent space and network are jointly optimized under the supervision of the reconstruction loss between synthesized image $\hat{I}_1$ and ground truth image I_1 from the dataset.

C. Technical Components

Here we present three technical components that are helpful to the C-S disentanglement.

C.1. Instance Discrimination

We first pretrain a ResNet-18 (He et al., 2016) Φ unsupervisedly with the method in Wu et al. (2018) and define a collection of layers of Φ as $\{\Phi_l\}$. Given two images I_i and I_j, we mix the embeddings to generate $u = G_\theta(s_i, c_j)$ and $v = G_\theta(s_j, c_i)$. For samples sharing the same style embedding, we enforce the feature distance in Φ between them to be close. For samples with different style embeddings, we enforce the feature distance in Φ between them to be large. This loss term can be written as

$$\mathcal{L}_{\mathcal{ID}} = \frac{\sum_l \lambda_l(\|\Phi_l(u) - \Phi_l(x)\|_1 + \|\Phi_l(v) - \Phi_l(y)\|_1)}{\sum_l \lambda_l(\|\Phi_l(u) - \Phi_l(y)\|_1 + \|\Phi_l(v) - \Phi_l(x)\|_1)}, \tag{1}$$

where $x = G_\theta(s_i, c_i)$ and $y = G_\theta(s_j, c_j)$. The hyperparameters $\{\lambda_l\}$ balance the contribution of each layer l to the loss. $\{\lambda_l\}$ are set to be $[1, 1, 1, 1, 1]$

C.2. Information Bottleneck

Similar to Anneal VAE (Burgess et al., 2018), we introduce a information bottleneck given by

$$\mathcal{L}_{\mathcal{IB}} = \gamma_s\|s^2 - C_s\|_1 + \gamma_c\|c^2 - C_c\|_1 \tag{2}$$

where C_s and C_c are the information capacity controlling the amount of information of the content and style respectively. During training, C_s and C_c increase linearly. The rate of increase is controlled by the increase steps and the maximum value. By controlling the increase rate, the content is forced to encode information first, so that the learning process is more consistent with our assumptions.

For the information bottleneck, we determine the increase steps and the maximum of the information capacity C_c and C_s by taking the training process of the model without the information bottleneck as a reference. We can enhance the model inductive bias by tuning these parameters. For Chairs, we set the maximum of C_c to 5, the start value of C_c to 2, the increase steps of C_c to 1.4×10^5, γ_c to 1 and γ_s to 0. Note that our model achieves state-of-the-art performance on Chairs even without information bottleneck.

C.3. Latent Optimization.

In the C-S disentanglement literature, it is common to use encoders to predict embeddings, while latent optimization (Bojanowski et al., 2018; Gabbay & Hoshen, 2020) directly optimizes the embeddings via back-propagating without using encoders. Encoders have a large number of parameters and require a lot more extra effort for training. Therefore, we adopt the latent optimization approach to update the latent spaces directly.

C.4. Reparametric Module

Inspired by VAE (Kingma & Welling, 2014), we design a reparametric module to force the latent space to be continuous. Thus, the embeddings encoding similar information will get closer in the latent space. Assume we have a mean embedding μ with a standard deviation σ, the reparametrized output is $\sigma X + \mu$, where $X \sim \mathcal{N}(0,\ I)$. To further simplify the problem, we set $\sigma = 1$ following Wu et al. (2019b) and Gabbay & Hoshen (2020). The mean embedding is the input style or content embedding. The reparametric module can make the latent space continuous, which is helpful for backpropagation.

D. More Results

In this section, we demonstrate more qualitative comparison and more qualitative results (including more datasets).

D.1. More qualitative experiments

In the main paper, for unsupervised baselines, we only compare our method with FactorVAE (Kim & Mnih, 2018) limited to space. As shown in Figure 2, we also outperform Wu et al. (2019b). For Wu et al. (2019b), the disentanglement is poor, such that the content embeddings control almost all the factors while the style embeddings control the tone.

For datsets in the main paper, We provide more qualitative results in Fig. 15, 16, 17, 18 and 19. Moreover, we also apply our method on higher resolution images and achieve good performance, as shown in Figure 3.

Wu et al. Ours

Figure 2. Comparison between Wu et al. and our method. For Wu et al., the images are mainly determined by content embeddings, while style embeddings only change the tone.

Figure 3. Results on Celeba with 128×128 resolution. Zoom in for better view.

D.2. More datasets

Besides the datasets introduced in the main paper, we make additional experiments on other datasets: such as **MNIST** (LeCun et al., 2010), **Cat** (Parkhi et al., 2012; Zhang et al., 2008), **Anime** (Chao, 2019) and **Market-1501** (Zheng et al., 2015). MNIST has 70k examples for 10 handwritten digits. Cat has 1.2k cat head images. Anime contains 63,632 anime faces. Market-1501 have 25,259 images. The results are shown in Figure 4, 5 and 6. Furthermore, we show our results on the Market-1501 dataset in Figure 7, which demonstrates our method can disentangle the human pose and the appearance even though the skeletons have large variances.

Figure 4. Results on the Cat dataset. Content indicates pose, and style indicates identity. The result further qualitatively demonstrates the ability of our method to disentangle on real-world data.

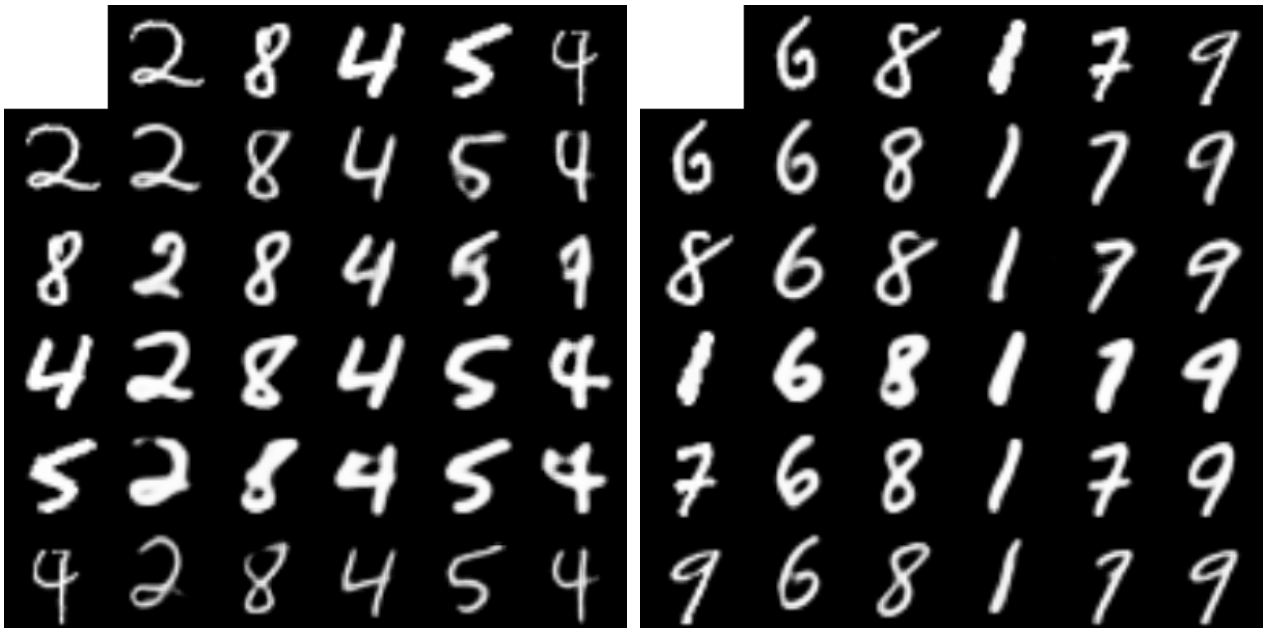


Figure 5. Results on the MNIST dataset. Content indicates geometric attributes, and style indicates texture.

Figure 6. Results on the Anime dataset. Content indicates pose, and style indicates identity.

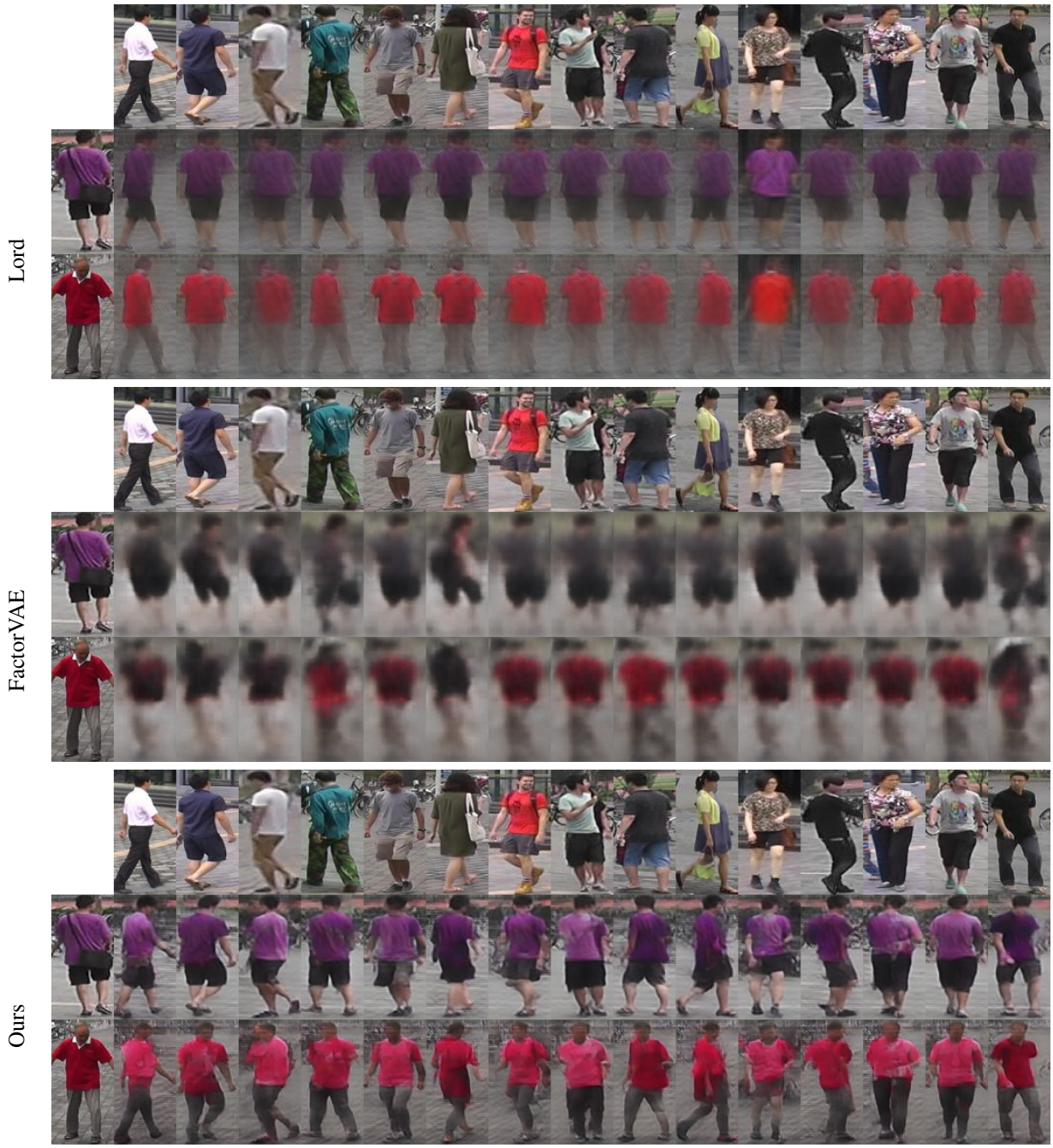


Figure 7. Comparison of visual analogy results on Market-1501 dataset. Our method outperforms supervised method Lord (Gabbay & Hoshen, 2020) and unsupervised method FactorVAE (Kim & Mnih, 2018) significantly.

E. More Ablation Study

Here we perform more ablation study for the technical modules.

If we use an amortized scheme instead of a latent optimization scheme, there are leaks between style and content latent space, and the result is worse than latent optimization, as shown in Figure 8 (a) and (c). Furthermore, if we do not use a reparametric module, we find the reconstruction performance is worse, as shown in Figure 8 (b). For the instance discrimination loss, the comparison is shown in Table 2. The disentanglement is better with an instance discrimination loss. For the information bottleneck, as shown in Table 1, the result with an information bottleneck is much better than the one without it.

We also conduct experiments on the influence of the size of embeddings. As mentioned before, we empirically set the size of style embedding d_s to 256 and the size of content embedding d_c to 128, which achieves good performance on all the datasets. Here, we adjust the size of the embeddings to study the influence of it, as shown in Figure 9.

(a) Amortized (b) w/o R (c) Ours

Figure 8. Ablation study. R indicates reparametric module.

(a) $d_s = 128, d_c = 256$ (b) $d_s = 256, d_c = 256$ (c) $d_s = 256, d_c = 128$

Figure 9. Study on the influence of size of embeddings. d_s is the size of style embedding and d_c is the size of content embedding. For (a), the content embeddings contain shape of face, facial expression and pose. For (b), the content embeddings contain shape of face and facial expression. For (c), which is the setting used in our paper, the content embeddings contain pose.

Table 1. Ablation study for infomation bottleneck on Chairs dataset. Lower is better.

Method	Content transfer
Ours (w/o Information Bottleneck)	0.280
Ours	0.190

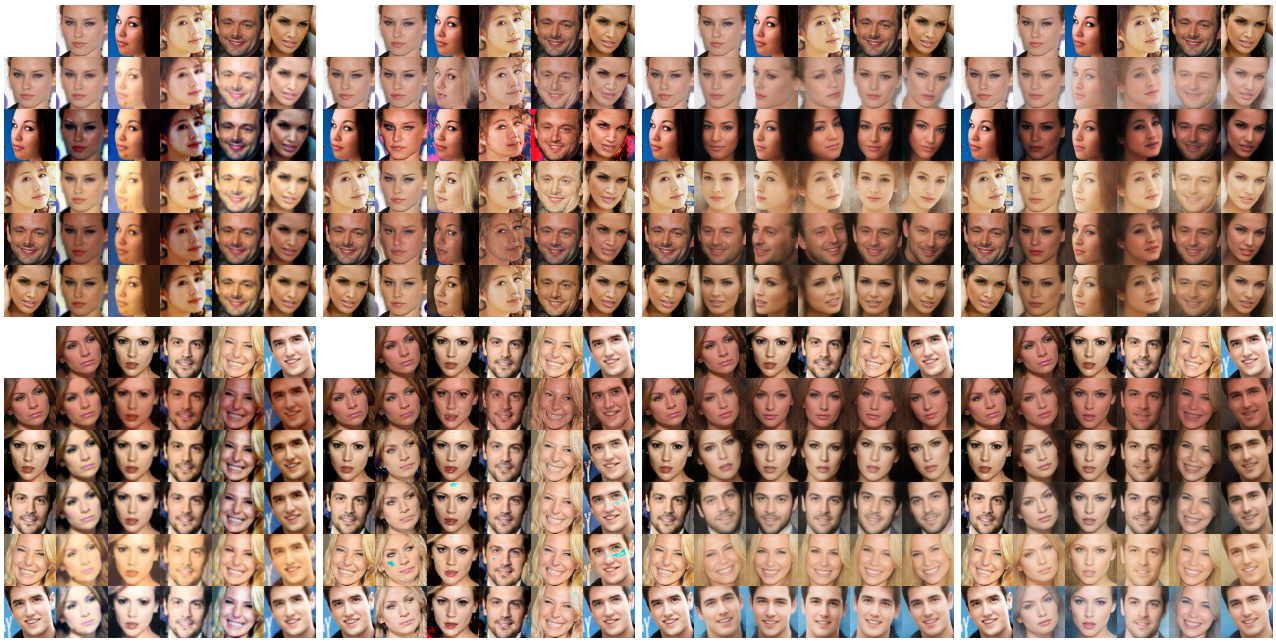
Table 2. Ablation study for instance discrimination on Celeba dataset. Lower is better.

Method	Content transfer
Ours (w/o Instance Discrimination)	0.165
Ours	0.161

F. Comparison with Selected Related Work

Comparison with StyleGAN. In our framework, the optimized content (conv) and style embeddings are disentangled representations of corresponding images. While StyleGAN (Karras et al., 2019) keeps the input of the convolution branch as a learnt constant for the whole dataset and finds the feature space of the "style" branch has disentanglement ability. For StyleGAN2 (Karras et al., 2020) [5], we select the subset of "style", which represents pose, as the content embedding and the rest subset as the style embedding. As shown in Figure 11, StyleGAN2 entangled pose with other semantic attributes, such as hair and glasses. As shown in Figure 19, the content of our method on human faces is pose attribute without entanglement.

Comparsion with MUNIT & Park et al. (2020). Besides the comparison in the main paper, we provide more qualitative comparison in Figure 10.



(a) MUNIT (b) Park et al. (2020) (c) Ours (d) Our fine

Figure 10. Comparison with MUNIT (Huang et al., 2018) and Park et al. (2020). MUNIT (Huang et al., 2018) and Park et al. (2020) learn the texture information which is different from Ours (c). Our fine (d) is that we only exchange the fine styles.

[5] We use the implementation from https://github.com/rosinality/stylegan2-pytorch.

StyleGAN2

Figure 11. Performance of StyleGAN2 (Karras et al., 2020) on human faces. For StyleGAN2, the content contains entangled semantic attribute, such as pose, hair and glasses. In our case, the content is pose, which is a high-level semantic attribute of the object.

G. More 3D Reconstruction

Our setting treats every image as a single identity (style) without ambiguity for augmenting single-view images. On Celeba, we use MVF-Net (Wu et al., 2019a) based on multi-view to reconstruct 3D facial shapes. For a given image, we can get the corresponding style embedding content embedding. Then we can get the front, left, and right view of this image combining the extracted style embedding and prepared content embeddings [6]. As shown in Figure 12, our augmented multi-view images are consistent, and the 3D meshes based on our method are more accurate than those based on Lord. As shown in Figure 13, we also provide additional results for Chairs.

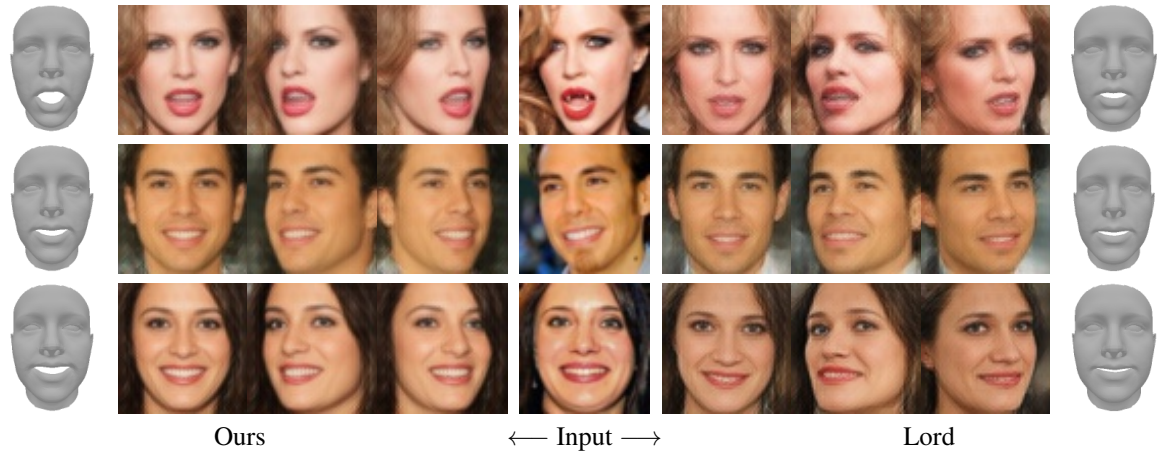


Figure 12. 3D face reconstruction. Given an image, we first generate multi-view images and then use them as augmented input.

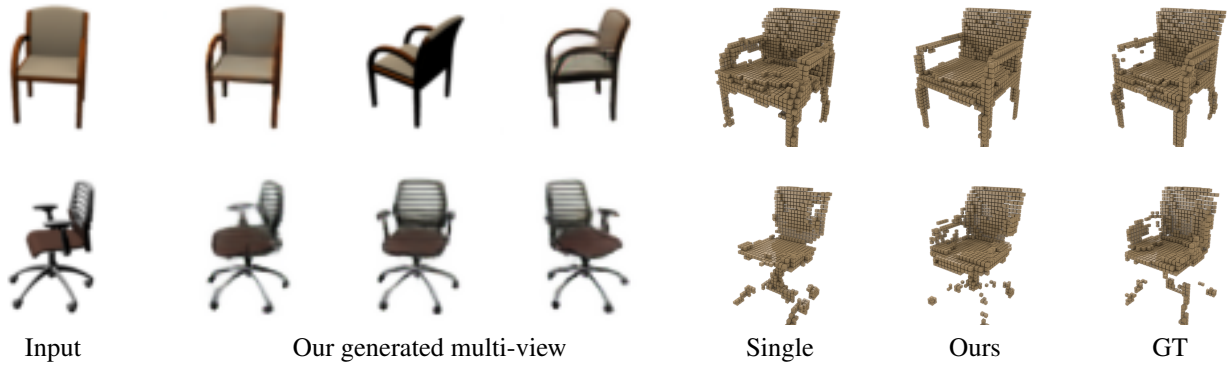


Figure 13. 3D reconstruction results on Chairs. We generate multi-view from Input. Single: the object reconstructed by only Input. Ours: the object reconstructed by multi-view inputs. GT: the object reconstructed by the ground truth of multi-view inputs.

[6] We retrieve the nearest neighborhood (facial landmarks space) of suggested inputs of MVF-Net and extract content embedding.

H. Cross-domain Application

As shown in the main paper, the content and style are disentangled in a single domain, and we demonstrate the domain transfer application without domain labels using our method. Furthermore, once the domain labels are given, we can disentangle and align the cross-domain dataset. This experiment may be helpful for domain transfer and domain adaptation. We train our model on the dataset that consists of Celeba and Anime. The model needs to be modified for learning cross-domain data: concatenate the domain embedding sharing in the corresponding domain and the image-specific style embedding, take it as the style embedding in the original model, and optimize all the embeddings during latent optimization. The results are shown in Figure 14. The learned poses are well aligned both in the animation and reality domain.

Figure 14. Results of modified model on merged cross-domain dataset based on Celeba and Anime. The learned content embedding are well aligned both in the animation and reality domain.

I. Proof

For the first term of Eq. 1 in the paper, we have

$$\max_{\theta,c_i,s_i} \sum_{i=1}^{N} \log \hat{P}_{\theta,s_i}(\boldsymbol{x} = I_i|\boldsymbol{c} = c_i), \tag{3}$$

Here we assume $\hat{P}_{\theta,s_i}$ is a Gaussian distribution,

$$\hat{P}_{\theta,s_i}(\boldsymbol{x}|\boldsymbol{c} = c_i) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2\sigma^2}\|x - G_\theta(s_i, c_i)\|_2^2\right). \tag{4}$$

Combining Eq. 3 and Eq. 4, we have

$$\max_{\theta,c_i,s_i} \sum_{i=1}^{N} \left(-\frac{1}{2\sigma^2}\|I_i - G_\theta(s_i, c_i)\|_2^2\right). \tag{5}$$

Consequently, the final optimization target of the first term is

$$\min_{\theta,c_i,s_i} \sum_{i=1}^{N} \|I_i - G_\theta(z_i)\|_2^2. \tag{6}$$

Q.E.D.

Figure 15. More visual anology of our method on Car3D.

Figure 16. More visual anology of our method on Car3D.

Figure 17. More visual anology of our method on Chairs.

Figure 18. More visual anology of our method on Chairs.

Figure 19. More visual anology of our method on Celeba.

References

Bojanowski, P., Joulin, A., Lopez-Paz, D., and Szlam, A. Optimizing the latent space of generative networks. In *ICML*, 2018.

Bulat, A. and Tzimiropoulos, G. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *ICCV*, 2017.

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

Chao, B. A collection of high-quality anime faces. https://github.com/bchao1/Anime-Face-Dataset, 2019.

Denton, E. L. and Birodkar, V. Unsupervised learning of disentangled representations from video. In *NeurIPS*, 2017.

Gabbay, A. and Hoshen, Y. Demystifying inter-class disentanglement. In *ICLR*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Huang, X., Liu, M., Belongie, S. J., and Kautz, J. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

Jha, A. H., Anand, S., Singh, M., and Veeravasarapu, V. S. R. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *ECCV*, 2018.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020.

Kim, H. and Mnih, A. Disentangling by factorising. In *ICML*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *ICLR*, 2014.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Park, T., Zhu, J., Wang, O., Lu, J., Shechtman, E., Efros, A. A., and Zhang, R. Swapping autoencoder for deep image manipulation. In *NeurIPS*, 2020.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *CVPR*, 2012.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Wu, F., Bao, L., Chen, Y., Ling, Y., Song, Y., Li, S., Ngan, K. N., and Liu, W. Mvf-net: Multi-view 3d face morphable model regression. In *CVPR*, 2019a.

Wu, W., Cao, K., Li, C., Qian, C., and Loy, C. C. Disentangling content and style via unsupervised geometry distillation. In *ICLRW*, 2019b.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Zhang, W., Sun, J., and Tang, X. Cat head detection - how to effectively exploit shape and texture features. In *ECCV*, 2008.

Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., and Tian, Q. Scalable person re-identification: A benchmark. In *ICCV*, 2015.