Do Generative Models Know Disentanglement? Contrastive Learning is All You Need
Appendix

A. Latent Traversals

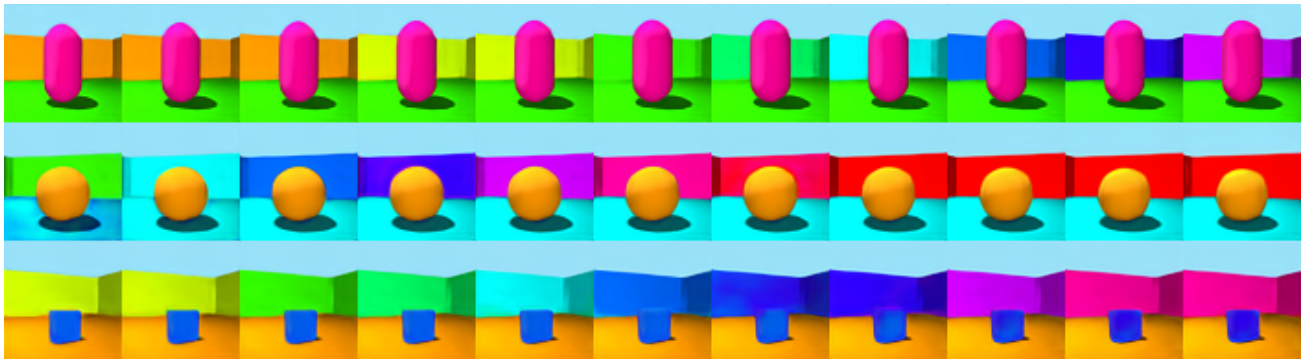
In this section, we visualize the disentangled directions of the latent space discovered by `DisCo` on each dataset.

(a) StyleGAN2 Cars3D – Azimuth

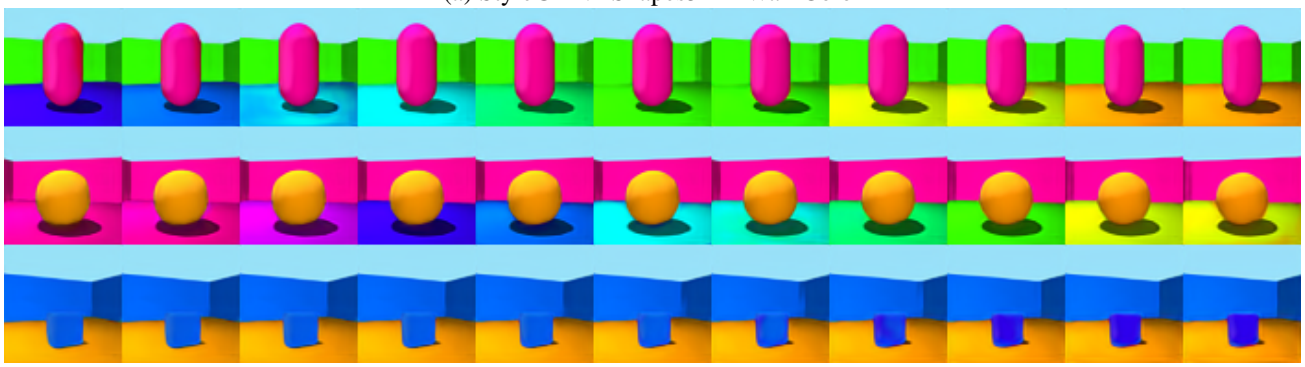
(b) StyleGAN2 Cars3D – Yaw

(c) StyleGAN2 Cars3D – Type

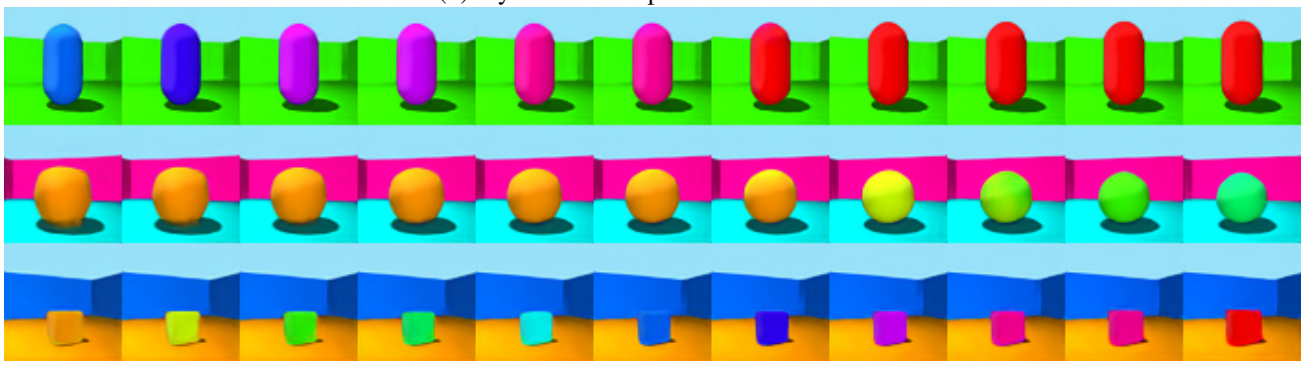
Figure 1. Examples of disentangled directions for StyleGAN2 on Cars3D discovered by `DisCo`.



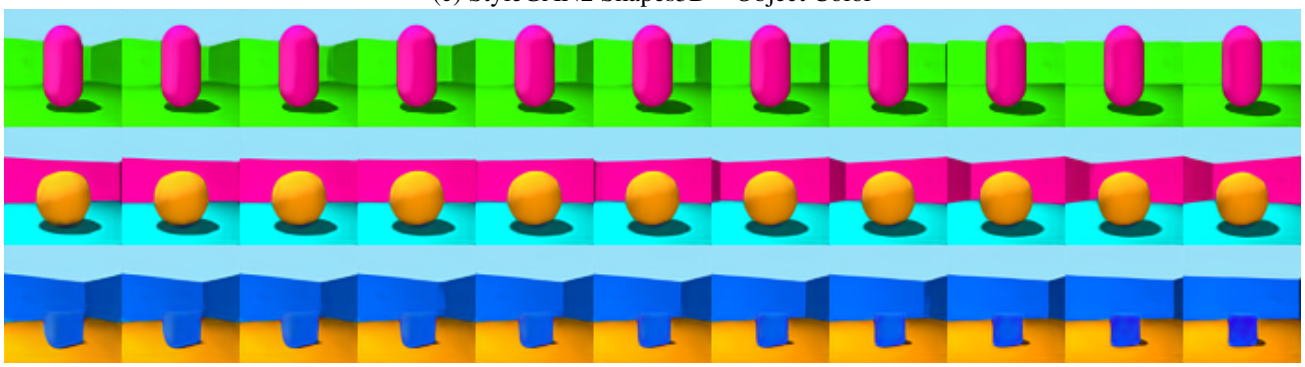
(a) StyleGAN2 Shapes3D – Wall Color



(b) StyleGAN2 Shapes3D – Floor Color



(c) StyleGAN2 Shapes3D – Object Color

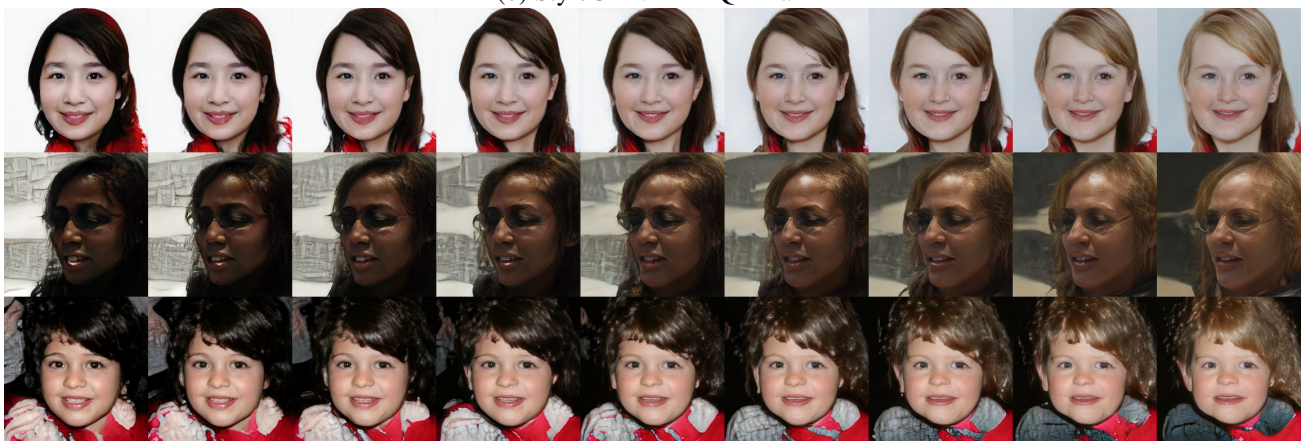


(d) StyleGAN2 Shapes3D – Pose

Figure 2. Examples of disentangled directions for StyleGAN2 on Shapes3D discovered by `DisCo`. As shown in (b), the latent space have local semantic.

(a) StyleGAN2 FFHQ – Oldness

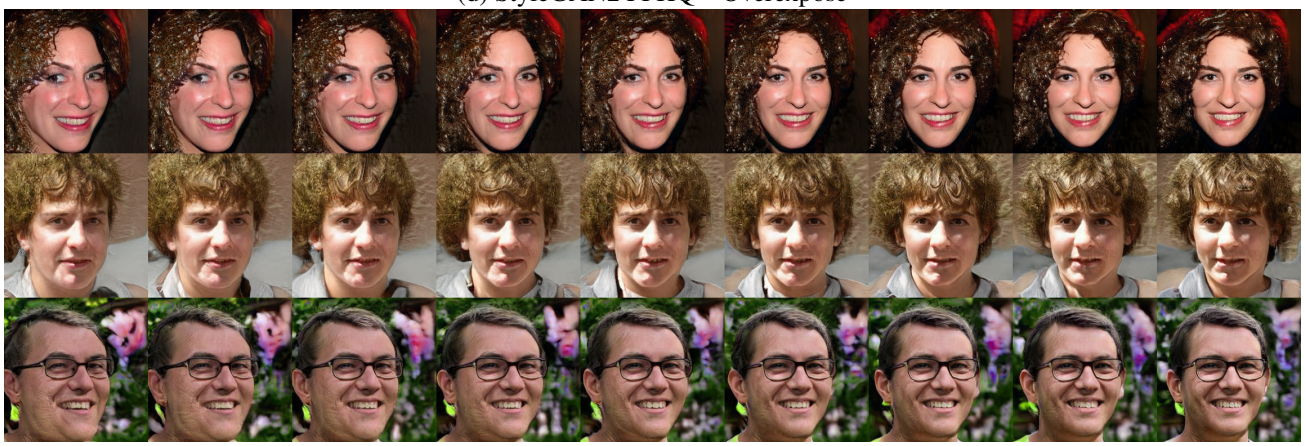
(b) StyleGAN2 FFHQ – Hair



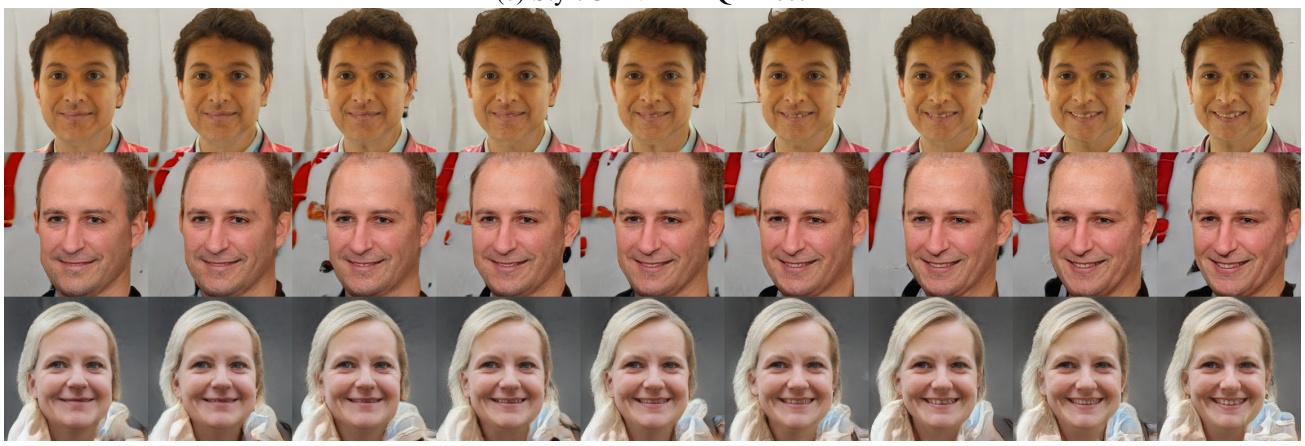
(c) StyleGAN2 FFHQ – Race

Figure 3. Examples of disentangled directions for StyleGAN2 on FFHQ discovered by `DisCo`.

(d) StyleGAN2 FFHQ – Overexpose



(e) StyleGAN2 FFHQ – Pose



(f) StyleGAN2 FFHQ – Smile

Figure 4. Examples of disentangled directions for StyleGAN2 on FFHQ discovered by `DisCo`.

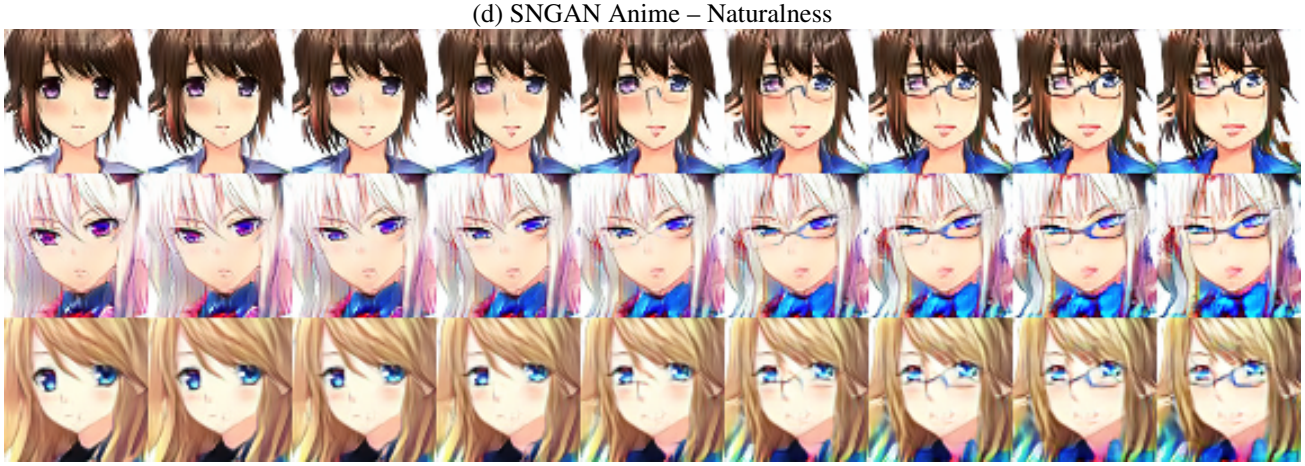
(a) SNGAN Anime – Tone

(b) SNGAN Anime – Skin

(c) SNGAN Anime – Pose

Figure 5. Examples of disentangled directions for SNGAN on Anime discoverd by `DisCo`.

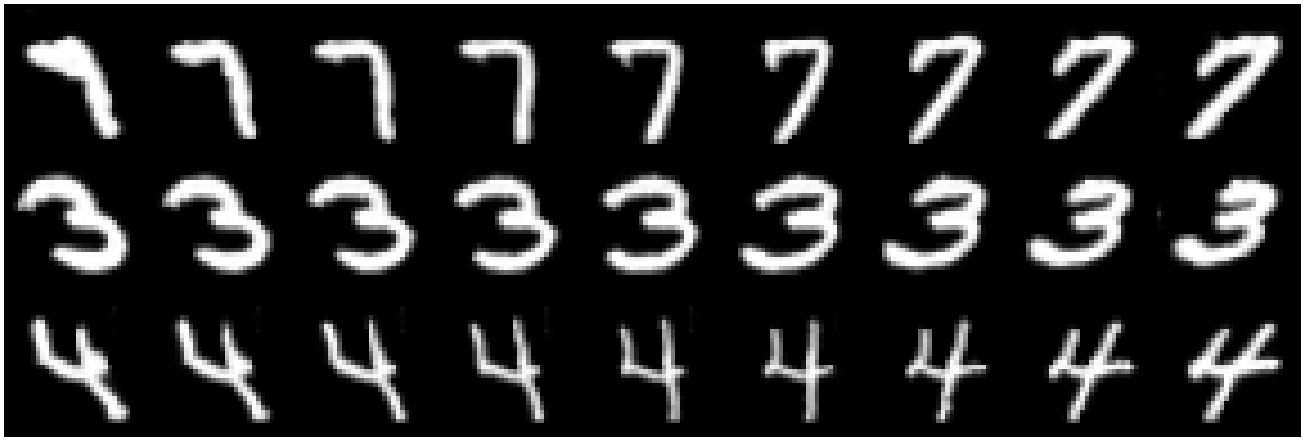
(d) SNGAN Anime – Naturalness



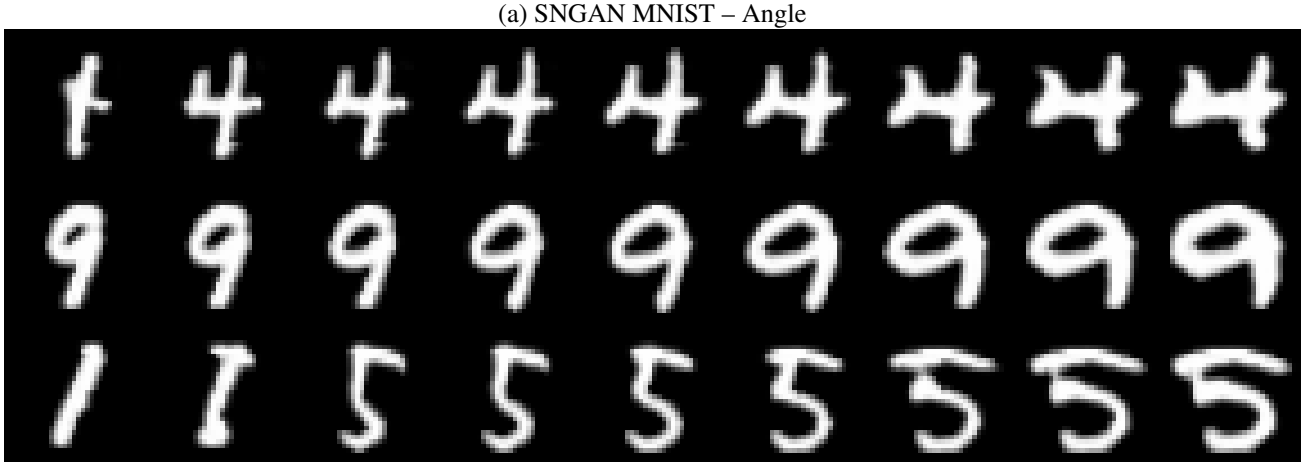
(e) SNGAN Anime – Glass

(f) SNGAN Anime – Whiteness

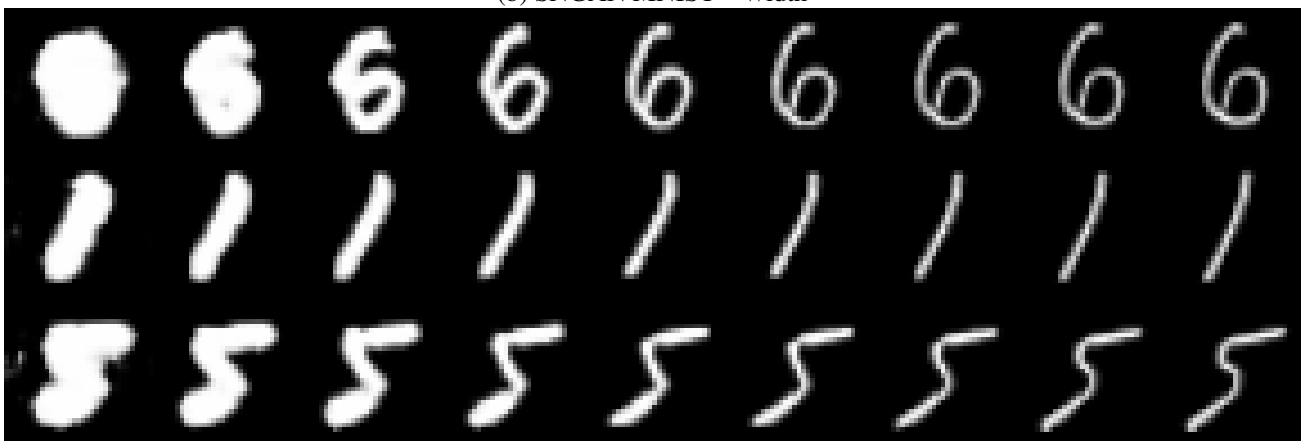
Figure 6. Examples of disentangled directions for SNGAN on Anime discovered by `DisCo`.



(a) SNGAN MNIST – Angle

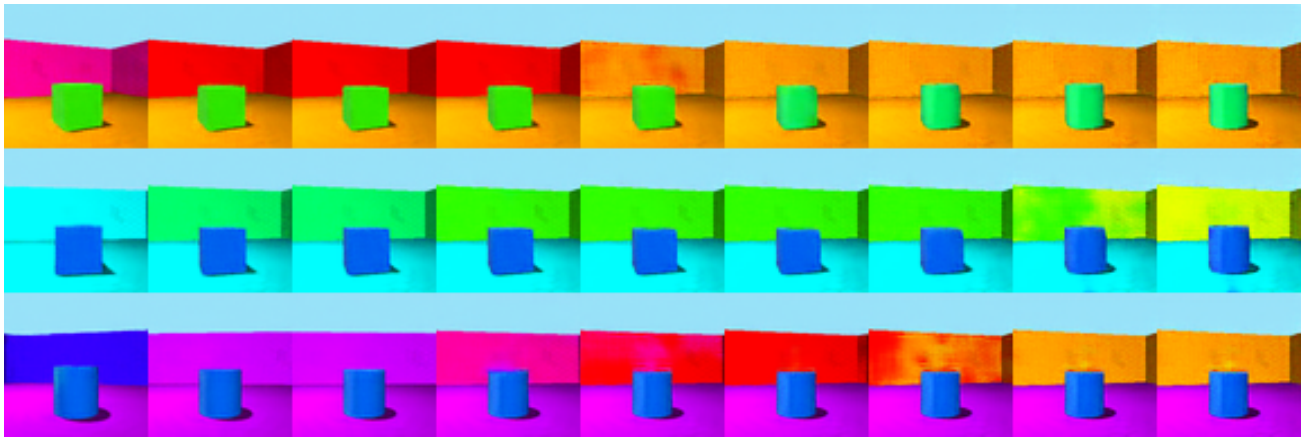


(b) SNGAN MNIST – Width

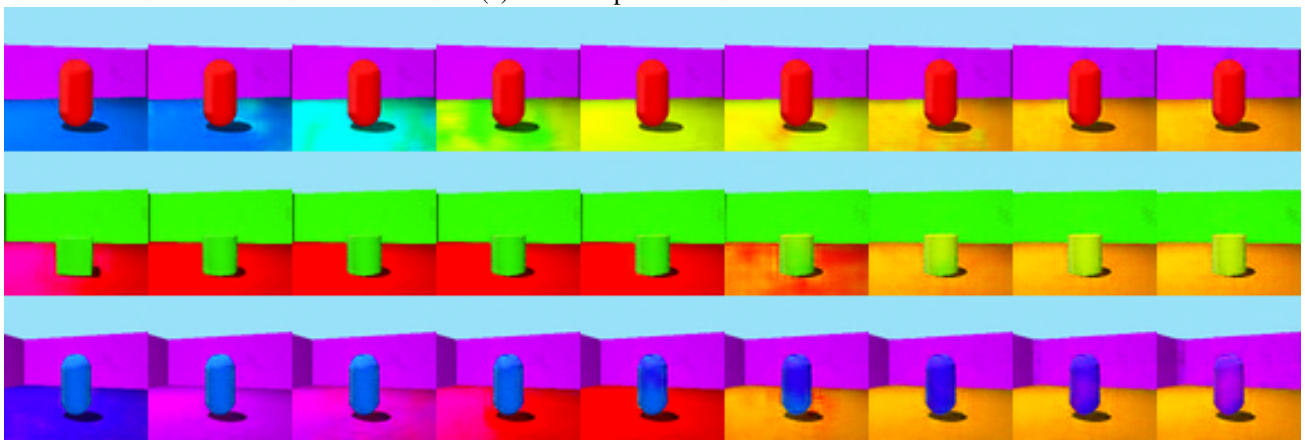


(c) SNGAN MNIST – Thickness

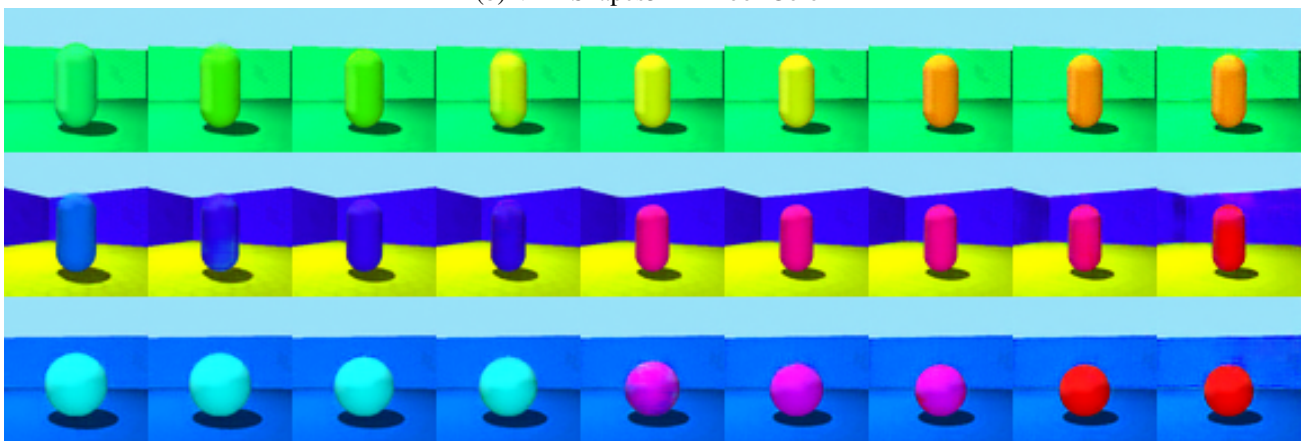
Figure 7. Examples of disentangled directions for SNGAN on MNIST discovered by `DisCo`.



(a) VAE Shapes3D – Wall Color

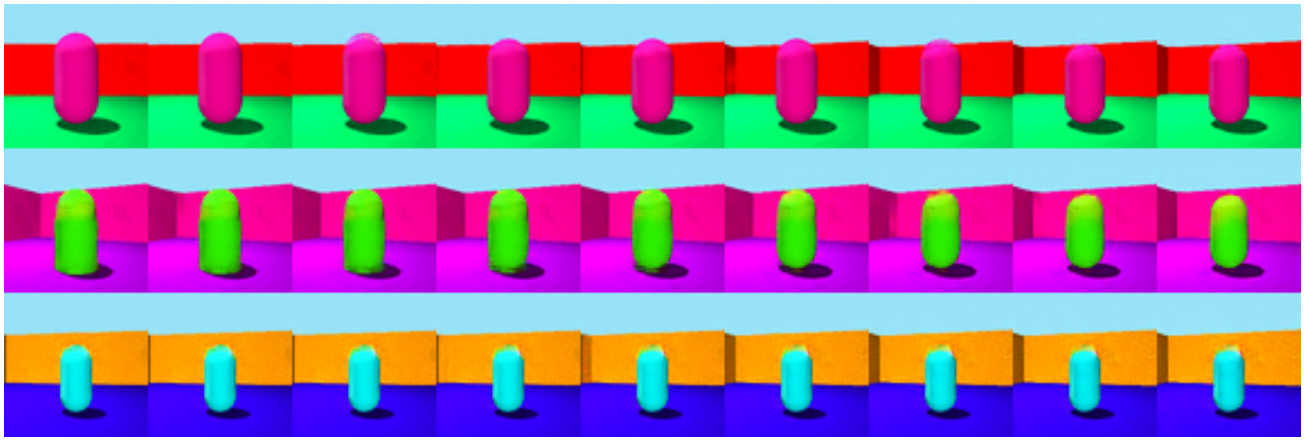


(b) VAE Shapes3D – Floor Color

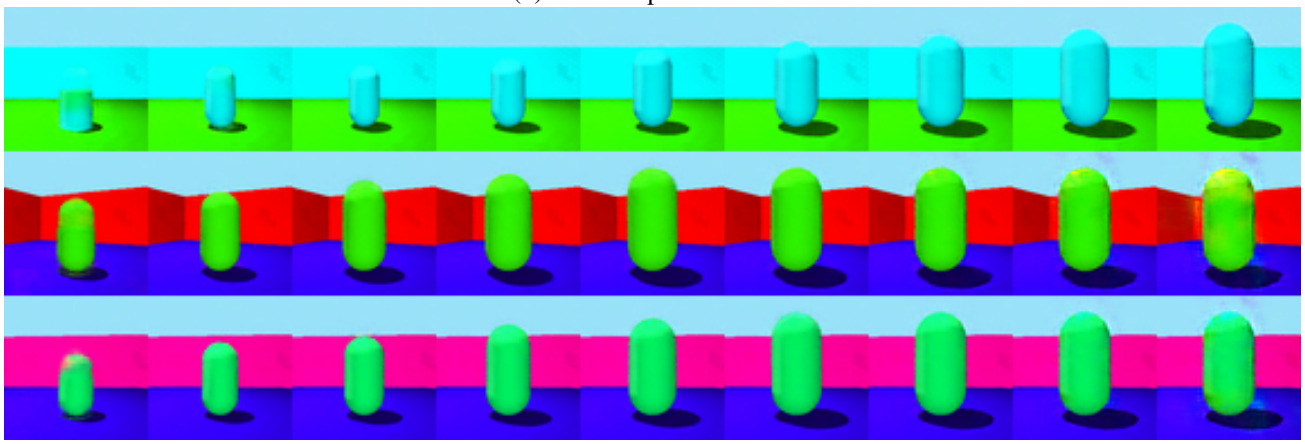


(c) VAE Shapes3D – Object Color

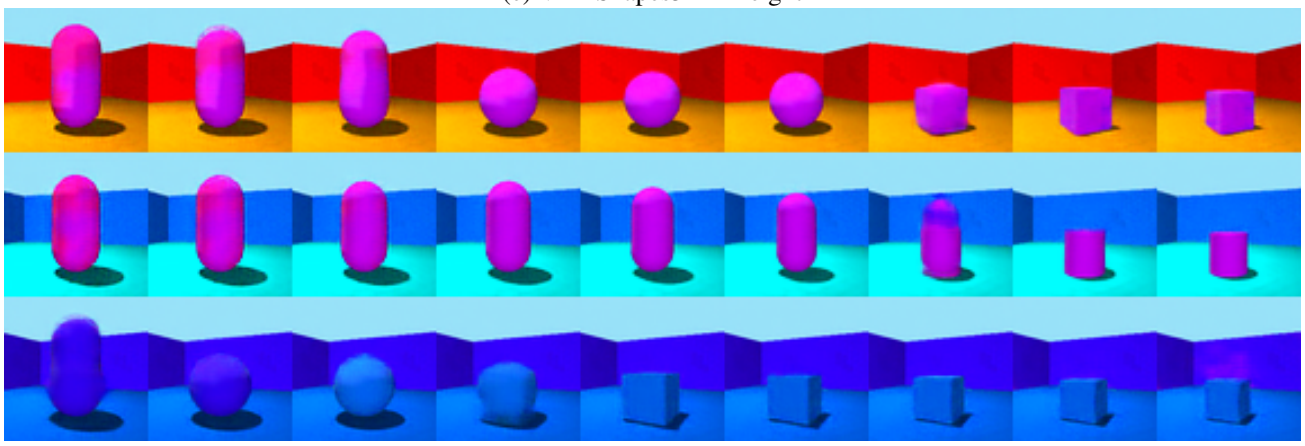
Figure 8. Examples of disentangled directions for VAE on Shapes3D discovered by `DisCo`.



(d) VAE Shapes3D – Pose

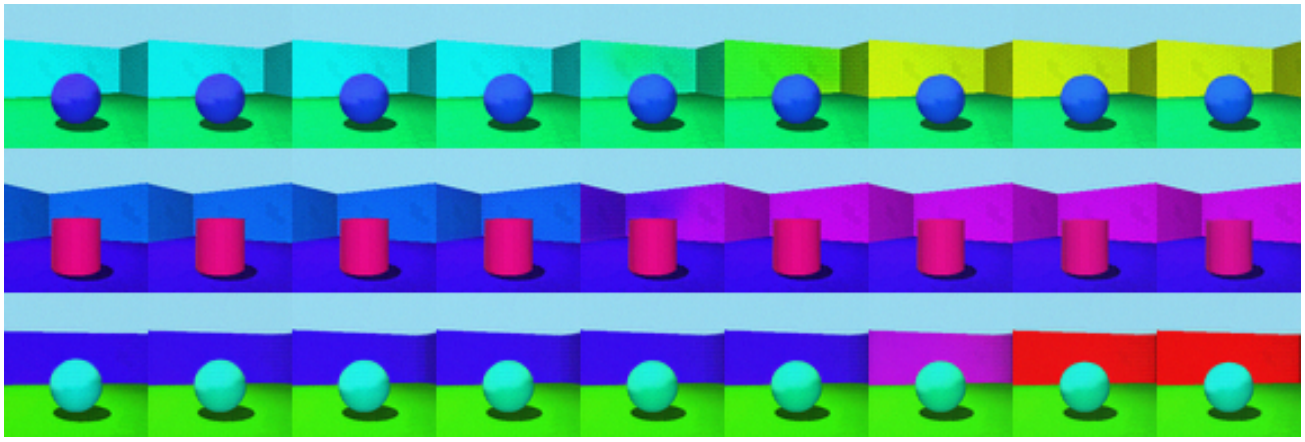


(e) VAE Shapes3D – Height

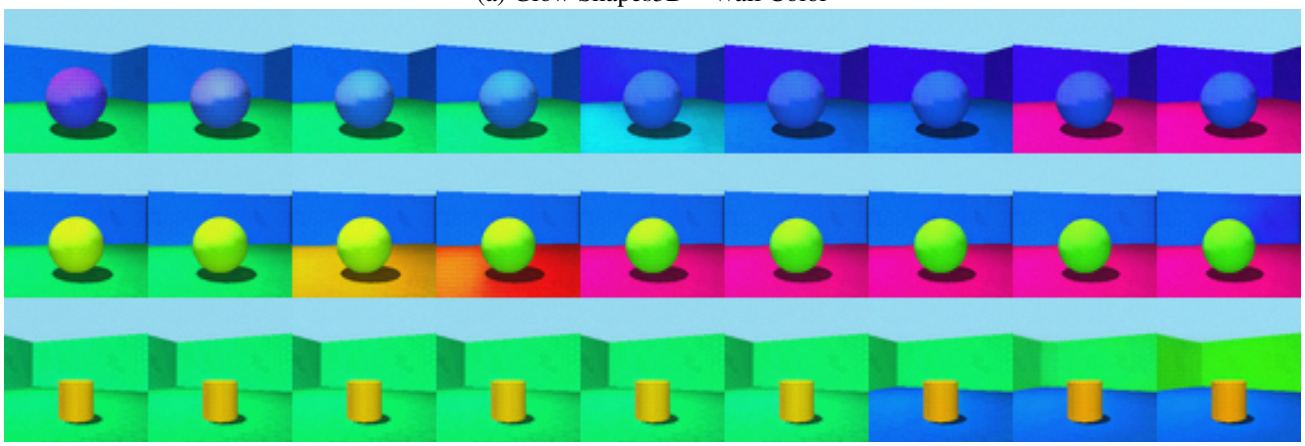


(f) VAE Shapes3D – Object Shape

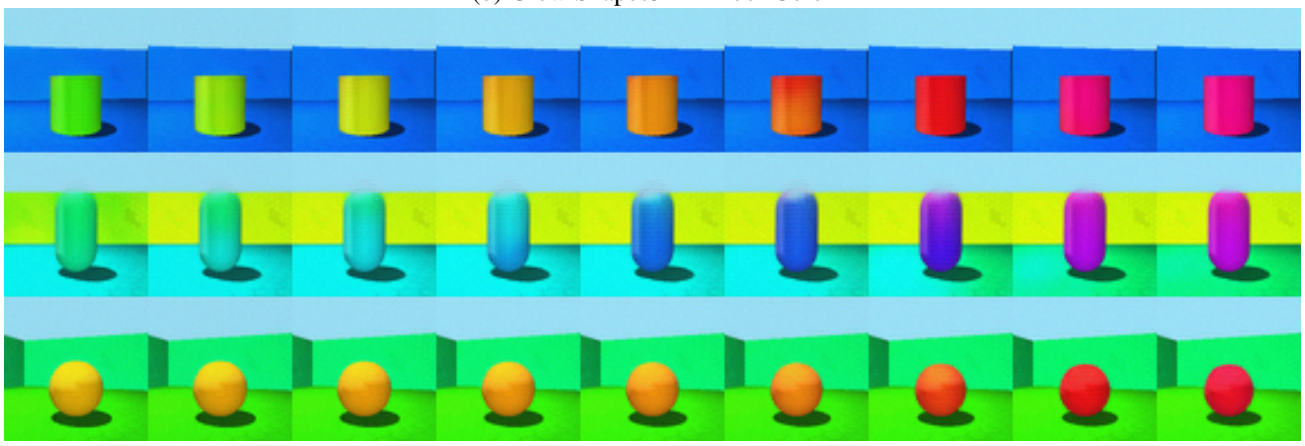
Figure 9. Examples of disentangled directions for VAE on Shapes3D discovered by `DisCo`.



(a) Glow Shapes3D – Wall Color



(b) Glow Shapes3D – Floor Color



(c) Glow Shapes3D – Object Color

Figure 10. Examples of disentangled directions for Glow on Shapes3D discovered by `DisCo`.

B. Implementation Details

B.1. Setting for baselines

VAE-based methods. We choose FactorVAE and β-TCVAE as the SOTA VAE-based methods, we follow Locatello et al. (2019) to use the same architecture of encoder and decoder and set the latent dimension of representation to 10. For FactorVAE, we set the hyperparameter γ to 10. For β-TCVAE, we set the hyperparameter β to 6. We run 25 times with different random seeds for each model.

InfoGAN-based methods. We choose InfoGAN-CR as a baseline. We use the official implementation [1] with the default settings.

GAN-based methods. We follow Khrulkov et al. (2021) to use the same settings for the following four baselines: LD (GAN), CF, GS, and DS. We take the top-10 directions for 5 different random seeds for GAN and 5 different random seeds for the additional encoder to learn disentangled representations.

LD (VAE) & LD (Flow). We follow LD (GAN) to use the same settings and substitute the GAN with VAE / Glow. The only exception is the randomness for LD (Flow). We only run one random seed to pretrain the Glow and use one random seed for the encoder.

B.2. Setting for `DisCo`

We set the hyperparameters temperature τ to 1, threshold T to 0.95, batch size B to 32, the number of positives N to 32, the number of negatives K to 64, λ to 1, learning rate lr to $1e-5$, the number of directions D to 64 and the dimension of the representation n to 32. We use an Adam optimizer (Kingma & Ba, 2015) in the training process, as shown in Table 1. We follow Khrulkov et al. (2021) to run 5 random seeds to pretrain the GAN and 5 random seeds for training `DisCo`. We have the same setting for `DisCo` on GAN, VAE, and Flow on all three datasets. The only exception is the randomness for Flow. We only use one random seed to pretrain the Glow and use one random seed for `DisCo`.

B.3. Domain gap problem

Please note that there exists a domain gap between the generated images of pretrained generative models and the real images. However, the experiments show that the domain gap has limited influence on the performance of `DisCo`.

Table 1. Optimizer for `DisCo`

Parameter	Values
Optimizer	Adam
Adam: beta1	0.9
Adam: beta2	0.999
Adam: epsilon	1.00e-08
Adam: learning rate	0.00001

[1] https://github.com/fjxmlzn/InfoGAN-CR

B.4. Architecture

Here, we provide the model architectures in our work. For the architecture of StyleGAN2, we follow Khrulkov et al. (2021). For the architecture of Glow, we use the open-source implementation [2].

Table 2. Encoder E architecture used in `DisCo`. n is 64 for Shapes3D, MPI3D and Car3D.

Conv $7 \times 7 \times 3 \times 64$, $\texttt{stride} = 1$
ReLu
Conv $4 \times 4 \times 64 \times 128$, $\texttt{stride} = 2$
ReLu
Conv $4 \times 4 \times 128 \times 256$, $\texttt{stride} = 2$
ReLu
Conv $4 \times 4 \times 256 \times 256$, $\texttt{stride} = 2$
ReLu
Conv $4 \times 4 \times 256 \times 256$, $\texttt{stride} = 2$
ReLu
FC 4096×256
ReLu
FC 256×256
ReLu
FC $256 \times n$

Table 3. VAE's decoder architecture. Its encoder is the same with the encoder in `DisCo`. In our work, n is 64 .

FC $n \times 256$
ReLu
FC 256×256
ReLu
FC 256×4096
ConvTranspose $4 \times 4 \times 256 \times 256$, $\texttt{stride} = 2$
ReLu
ConvTranspose $4 \times 4 \times 256 \times 256$, $\texttt{stride} = 2$
ReLu
ConvTranspose $4 \times 4 \times 256 \times 128$, $\texttt{stride} = 2$
ReLu
ConvTranspose $4 \times 4 \times 128 \times 64$, $\texttt{stride} = 2$
ReLu
ConvTranspose $7 \times 7 \times 64 \times 3$, $\texttt{stride} = 1$

[2] https://github.com/rosinality/glow-pytorch

C. More Experiments

C.1. More quantitative comparison

We provide additional quantitative comparisons in terms of β-VAE score and FactorVAE score. `DisCo` on pretrained GAN is comparable to GAN-based baselines in terms of β-VAE score and FactorVAE score, suggesting that some disagreement between these two scores and MIG/ DCI. However, note that the qualitative evaluation in Figure 11 and Figure 12 shows that the disentanglement ability of `DisCo` is better than all the baselines on Shapes3D dataset.

Table 4. Comparisons of the β-VAE and FactorVAE scores on the Shapes3D dataset (mean $\pm$ variance). A higher mean indicates a better performance.

Method	Cars3D		Shapes3D		MPI3D	
	β-VAE score	FactorVAE score	β-VAE score	FactorVAE score	β-VAE score	FactorVAE score
Typical disentanglement baselines:						
FactorVAE	1.00 ± 0.00	0.906 ± 0.052	0.892 ± 0.064	0.840 ± 0.066	0.339 ± 0.029	0.152 ± 0.025
β-TCVAE	$0.999 \pm 1.0e-4$	0.855 ± 0.082	0.978 ± 0.036	0.873 ± 0.074	0.348 ± 0.012	0.179 ± 0.017
InfoGAN-CR	0.450 ± 0.022	0.411 ± 0.013	0.837 ± 0.039	0.587 ± 0.058	0.672 ± 0.101	0.439 ± 0.061
Methods on pretrained GAN:						
LD	$0.999 \pm 2.54e-4$	0.852 ± 0.039	0.913 ± 0.063	0.805 ± 0.064	0.535 ± 0.057	0.391 ± 0.039
CF	1.00 ± 0.00	0.873 ± 0.036	0.999 ± 0.001	0.951 ± 0.021	0.669 ± 0.033	0.523 ± 0.056
GS	1.00 ± 0.00	0.932 ± 0.018	0.944 ± 0.044	0.788 ± 0.091	0.605 ± 0.061	0.465 ± 0.036
DS	1.00 ± 0.00	0.871 ± 0.047	0.991 ± 0.022	0.929 ± 0.065	0.651 ± 0.043	0.502 ± 0.042
`DisCo` (ours)	$0.999 \pm 6.86e-5$	0.855 ± 0.074	0.987 ± 0.028	0.877 ± 0.031	0.530 ± 0.015	0.371 ± 0.030
Methods on pretrained VAE:						
LD	0.951 ± 0.074	0.711 ± 0.085	0.602 ± 0.196	0.437 ± 0.188	0.266 ± 0.068	0.242 ± 0.010
`DisCo` (ours)	$0.999 \pm 5.42e-5$	0.761 ± 0.114	$0.999 \pm 8.9e-4$	0.956 ± 0.041	0.411 ± 0.034	0.391 ± 0.075
Methods on pretrained Flow:						
LD	0.922 ± 0.000	0.633 ± 0.000	0.699 ± 0.000	0.597 ± 0.000	0.266 ± 0.000	0.242 ± 0.000
`DisCo` (ours)	1.00 ± 0.000	0.880 ± 0.000	0.860 ± 0.000	0.854 ± 0.000	0.538 ± 0.000	0.486 ± 0.000

C.2. Qualitative comparison

We provide some examples for qualitative comparison. As shown in Figure 11 and Figure 12, VAE-based methods suffer from poor image quality, and GAN-based methods tend to entangle with other attributes.

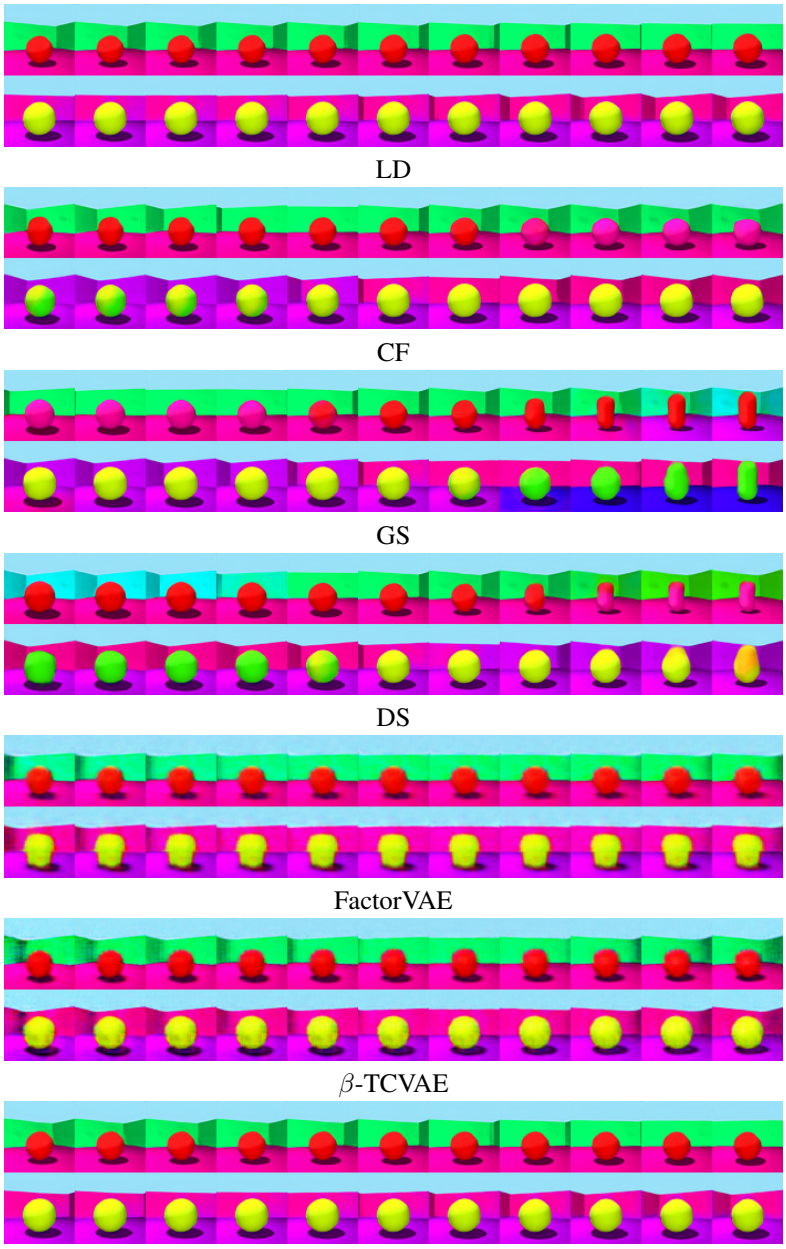


Figure 11. Comparison with baselines on Shapes3D dataset with *Pose* attribute.

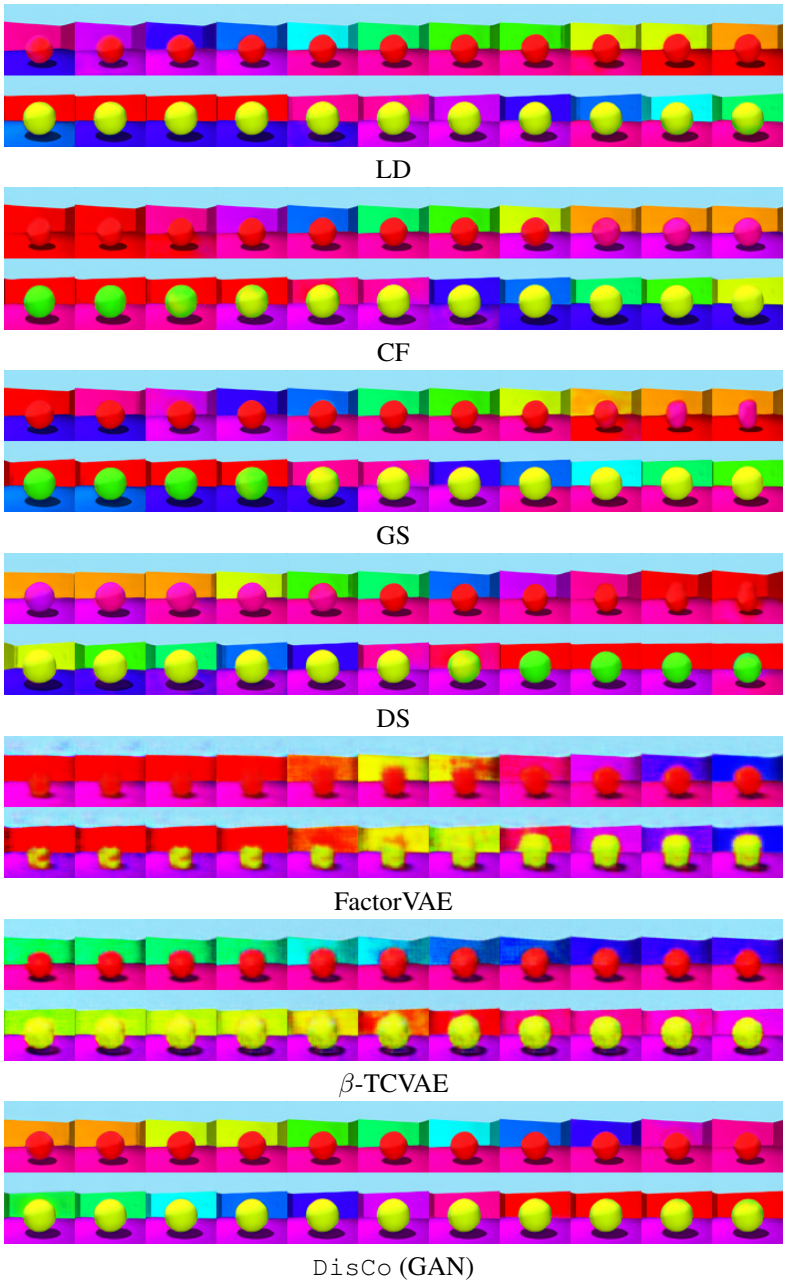


Figure 12. Comparison with baselines on Shapes3D dataset with *Wall Color* attribute. VAE-based methods suffer from poor image quality. GAN-based methods tend to entangle with other attributes.

D. Extension: Bridge the pretrained VAE and pretrained GAN

Researchers are recently interested in improving image quality given the disentangled representation generated by typical disentanglement methods. Lee et al. (2020) propose a post-processing stage using a Generative Adversarial Network based on disentangled representations learned by VAE-based disentanglement models. This method scarifies a little generation ability due to an additional constraint. Similarly, Srivastava et al. (2020) propose to use a deep generative model with AdaIN (Huang & Belongie, 2017) as a post-processing stage to improve reconstruction ability. Following this setting, we can replace the encoder in DisCo with an encoder pretrained by VAE-based disentangled baselines. In this way, we can bridge the pretrained disentangled VAE and pretrained GAN, as shown in Figure 13. Compared to previous methods, our method can fully utilize the state-of-the-art GAN and the state-of-the-art VAE-based method and does not need to train an deep generative model from scratch.

Figure 13. DisCo with a pretrained encoder allows synthesizing high-quality images by bridging pretrained β-TCVAE and pretrained StyleGAN2.

References

Huang, X. and Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

Khrulkov, V., Mirvakhabova, L., Oseledets, I., and Babenko, A. On disentangled representations extracted from pretrained gans, 2021. URL https://openreview.net/forum?id=VCAXR34cp59.

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

Lee, W., Kim, D., Hong, S., and Lee, H. High-fidelity synthesis with disentangled representation. In *ECCV*, 2020.

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019.

Srivastava, A., Bansal, Y., Ding, Y., Hurwitz, C. L., Xu, K., Egger, B., Sattigeri, P., Tenenbaum, J., Cox, D. D., and Gutfreund, D. Improving the reconstruction of disentangled representation learners via multi-stage modelling. *CoRR*, abs/2010.13187, 2020.