Appendix

A. More details of Two New Datasets

A.1. Examples of the Isaac3D Dataset

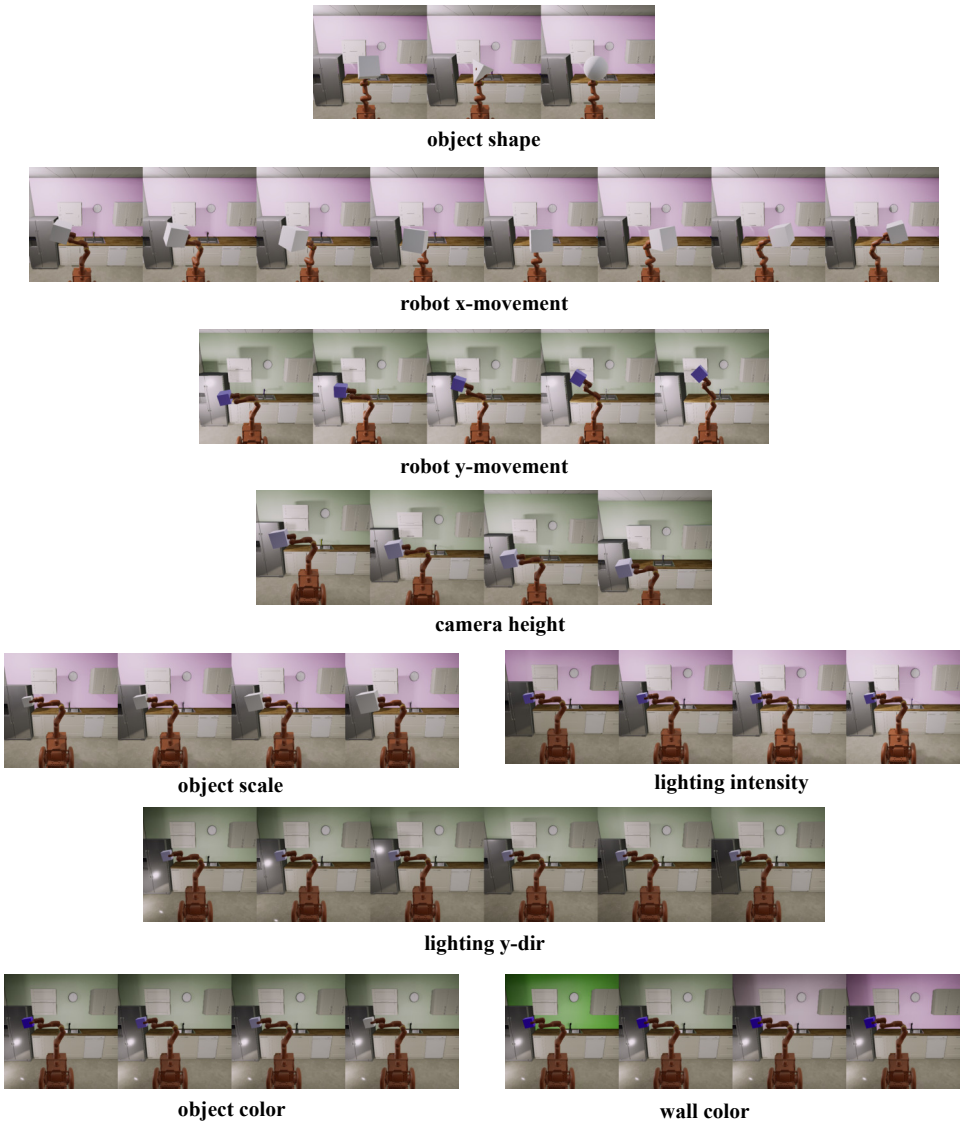


Figure 8. Examples of the Isaac3D dataset where we vary each factor of variation individually to see how each factor of variation changes with its corresponding ground-truth factor code.

A.2. Examples of the Falcor3D Dataset

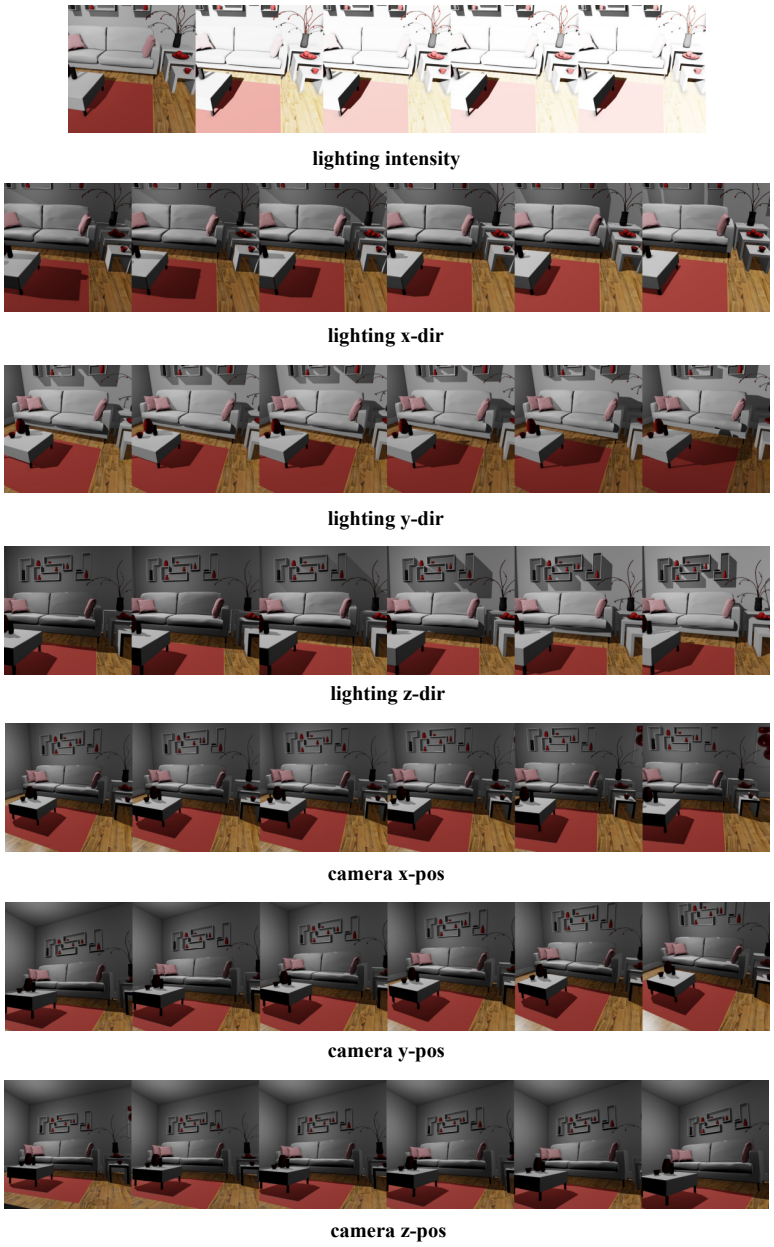


Figure 9. Examples in the Falcor3D dataset where we vary each factor of variation individually to see how each factor of variation changes with its corresponding ground-truth factor code.

B. More results of Info-StyleGAN

B.1. Progressive Training for Disentanglement Learning

Progressive growing has been shown to improve the image quality of GANs (Karras et al., 2017; 2019), however, its impact on disentanglement learning remains unknown so far. Thus, we compare the MIG scores of Info-StyleGAN with progressive and non-progressive growing, respectively, on both dSprites and Isaac3D, and the results are shown in Figure 10. We can see that with progressive growing, the disentanglement quality tends to be better on both two datasets. Besides, the gap of average MIG scores with different values of the hyperparameter γ is much smaller if the progressive growing is applied, which implies Info-StyleGAN with progressive growing seems to be less sensitive to hyperparameters. Therefore, unless stated otherwise, we use progressive growing for the GAN training in all the experiments.

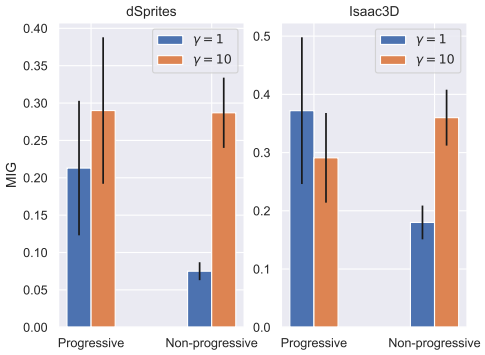


Figure 10. The impact of progressive training on the disentanglement learning with Info-StyleGAN. We can see that with progressive growing, the disentanglement quality tends to be better on both two datasets. Besides, the gap of average MIG scores with different values of the hyperparameter γ is much smaller if the progressive growing is applied, which implies Info-StyleGAN with progressive growing seems to be less sensitive to hyperparameters.

B.2. How to Get Info-StyleGAN* with Smaller Network Size

Mapping Network
(FC $\times$ n_{mp}) $f_{\text{mp}} \times f_{\text{mp}}$
Synthesis Network
(4×4 Conv) 3×3× $f_0 \times f_0$
(4×4 Conv) 3×3× $f_0 \times f_0$
(8×8 Conv) 3×3× $f_0 \times f_0$
(8×8 Conv) 3×3× $f_0 \times f_0$
(16×16 Conv) 3×3× $f_0 \times f_0$
(16×16 Conv) 3×3× $f_0 \times f_0$
(32×32 Conv) 3×3× $f_0 \times f_0$
(32×32 Conv) 3×3× $f_0 \times f_0$
(64×64 Conv) 3×3× $\frac{f_0}{2} \times \frac{f_0}{2}$
(64×64 Conv) 3×3× $\frac{f_0}{2} \times \frac{f_0}{2}$

(a) Generator

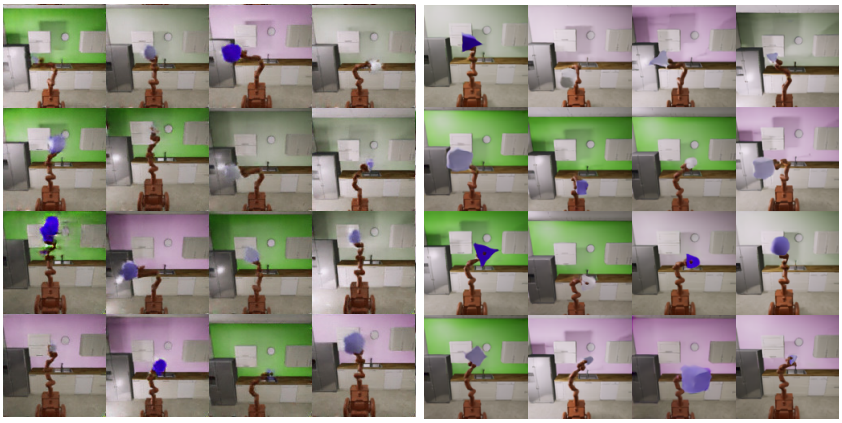
(64×64 Conv) 3×3× $\frac{f_0}{2} \times \frac{f_0}{2}$
(64×64 Conv) 3×3× $\frac{f_0}{2} \times \frac{f_0}{2}$
(32×32 Conv) 3×3×f_0×f_0
(32×32 Conv) 3×3×f_0×f_0
(16×16 Conv) 3×3×f_0×f_0
(16×16 Conv) 3×3×f_0×f_0
(8×8 Conv) 3×3×f_0×f_0
(8×8 Conv) 3×3×f_0×f_0
(4×4 Conv) 3×3×f_0×f_0
(4×4 FC) ($16 f_0$) × 64
(4×4 FC) 64×(1+ code_length)

(b) Discriminator / Encoder

Table 4. Generator and discriminator (or encoder) architectures in the implementation of Info-StylGAN for generating the image of resolution 128×128, where we use "FC $\times n_{\text{mp}}$" to denote that there are n_{mp} dense layers in the given block, and use "8×8 Conv" to denote the convolutional layer in the 8×8 resolution block. Note that there is not the last block (i.e., 64×64 Conv) in the generator and not the first block (i.e., 64×64 Conv) in the encoder if we want to generate the image of resolution 64×64. For the original Info-StyleGAN, we have $n_{\text{mp}} = 8$, $f_0 = 512$, $f_0 = 512$. For Info-StyleGAN* on dSprites, we set $n_{\text{mp}} = 3$, $f_{\text{mp}} = 64$, $f_0 = 64$ with 0.74M parameters in total. For Info-StyleGAN* on Downscaled Isaac3D, we set $n_{\text{mp}} = 3$, $f_{\text{mp}} = 256$, $f_0 = 128$ with 3.44M parameters in total.

B.3. Randomly Generated Samples of Baseline Models and Info-StyleGAN* (with Smaller Network Size)

(a) β-VAE (FID=120.3) (b) FactorVAE (FID=358.2) (c) β-TCVAE (FID=143.8)

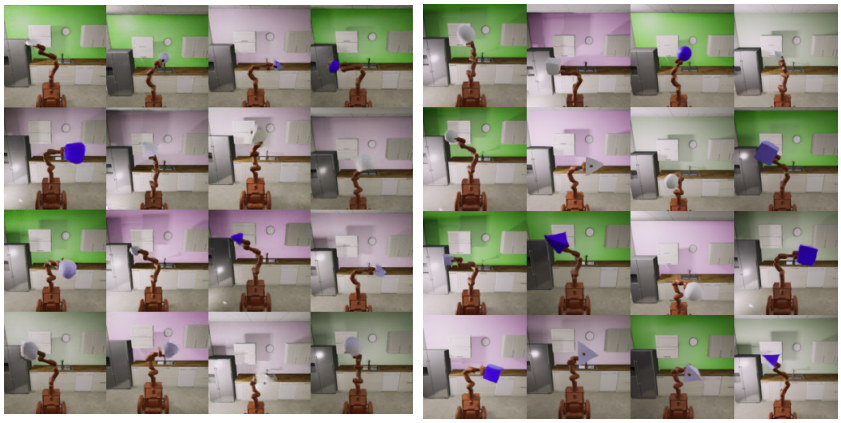


(d) InfoGAN-CR (FID=73.43) (e) Info-StyleGAN* (FID=8.38)

Figure 11. Randomly sampled images of baseline models and Info-StyleGAN* on (downscaled) Isaac3D of resolution 128x128. Note that for VAE-based models, we use the similar network architectures as in (Locatello et al., 2019a), and Info-StyleGAN* denotes the smaller version of Info-StyleGAN, in which its number of parameters is similar to baseline models. We can see that compared with Info-StyleGAN* (of the same network size), the generated images of VAE-based models (i) tend to be quite blurry and of low quality, and (ii) fail to cover all the variations in the dataset. As a strong GAN baseline, InfoGAN-CR is also significantly worse than Info-StyleGAN in terms of image quality. The results demonstrate that our proposed dataset can serve as a new challenging benchmark for disentanglement learning, in particular regarding the much higher resolution, and larger variation of factors.

B.4. Randomly Generated Samples of baseline models† (with Larger Network Size) and Info-StyleGAN

(a) β-VAE† (FID=60.71) (b) FactorVAE† (FID=60.67) (c) β-TCVAE† (FID=77.48)



(d) InfoGAN-CR (FID=30.41) (e) Info-StyleGAN (FID=2.50)

Figure 12. Randomly sampled images of baseline models† and Info-StyleGAN on (downscaled) Isaac3D of resolution 128x128. Note that for VAE-based models, we increase the number of featuremaps ($\times 8$) in each layer of the network architectures in (Locatello et al., 2019a), so that their number of parameters is similar to Info-StyleGAN. We also apply the same operations for InfoGAN-CR to match the network size of In-StyleGAN. We can see that (i) the image quality gets better after increasing the network size, (ii) compared with Info-StyleGAN (of the same network size), the generated images of VAE-based models still have issues with blurriness and failure in capturing all variations, (iii) the generated images of InfoGAN-CR are better than VAEs but still worse than Info-StyleGAN.

B.5. Other Experimental Settings

Our experiments are based on the StyleGAN implementation (Karras et al., 2019), where the GAN loss L_{GAN}, batch sizes, learning rates for both generator and discriminator, and other hyperparameters in the Adam optimizer and weights in each resolution black are all kept the same with (Karras et al., 2019), unless stated otherwise. Different from the original StyleGAN implementation, we do not apply truncation tricks. We also do not add noise inputs to introduce another randomness, as we consider the case where the factor code and latent z will capture all the factors in the data. For all the quantitative results in the paper, we report the error bars by taking the mean and standard deviation of four runs with random seeds. For the implementation of evaluation metrics, we use 50K random sampled real images and fake images to calculate the FID score. We use 5K ground-truth observation-code pairs as training samples and 2K ground-truth observation-code pairs as test samples to evaluate the Factor score. We use 10K ground-truth observation-code pairs and 10K generated observation-code pairs to calculate the MIG and MIG-gen scores, respectively. We also use 1K ground-truth observation-code pairs and 1K generated observation-code pairs to calculate the L2 and L2-gen scores, respectively.

C. More Results of Semi-StyleGAN

C.1. Semi-StyleGAN with 0.5% of Labeled Data on Isaac3D with Resolution 512x512

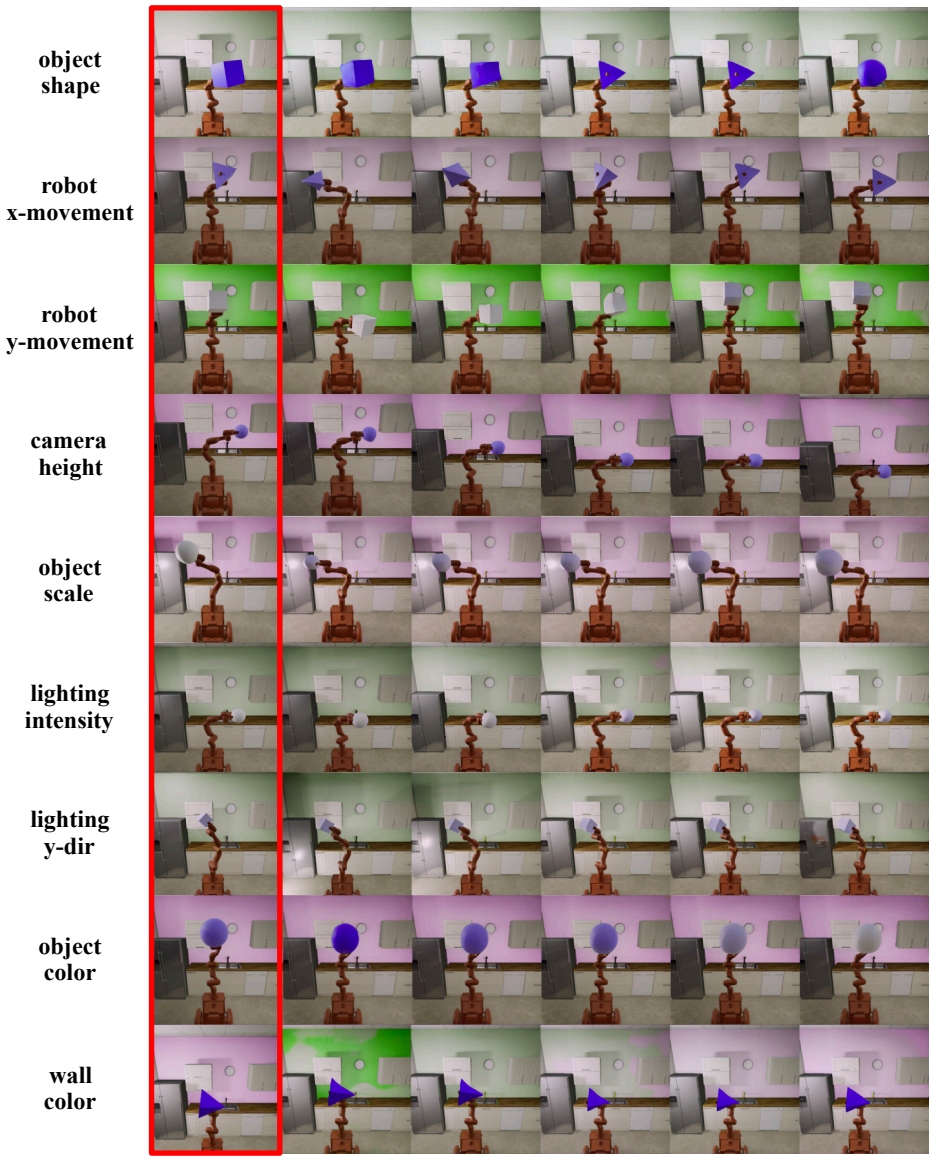


Figure 13. Latent traversal of Semi-StyleGAN on Isaac3D by using 0.5% of the labeled data. Images in the first column (marked by red box) are randomly sampled real images of resolution 512x512 and the rest images in each row are their interpolations, respectively, by uniformly varying the given factor from 0 to 1. We can see that each factor changes smoothly during its interpolation without affecting other factors, and the interpolated images in each row visually look almost the same with their input image except the considered varying factor. Also, the image quality does not get worse during the interpolations.

C.2. Semi-StyleGAN with 1% of Labeled Data on Falcor3D with Resolution 512x512

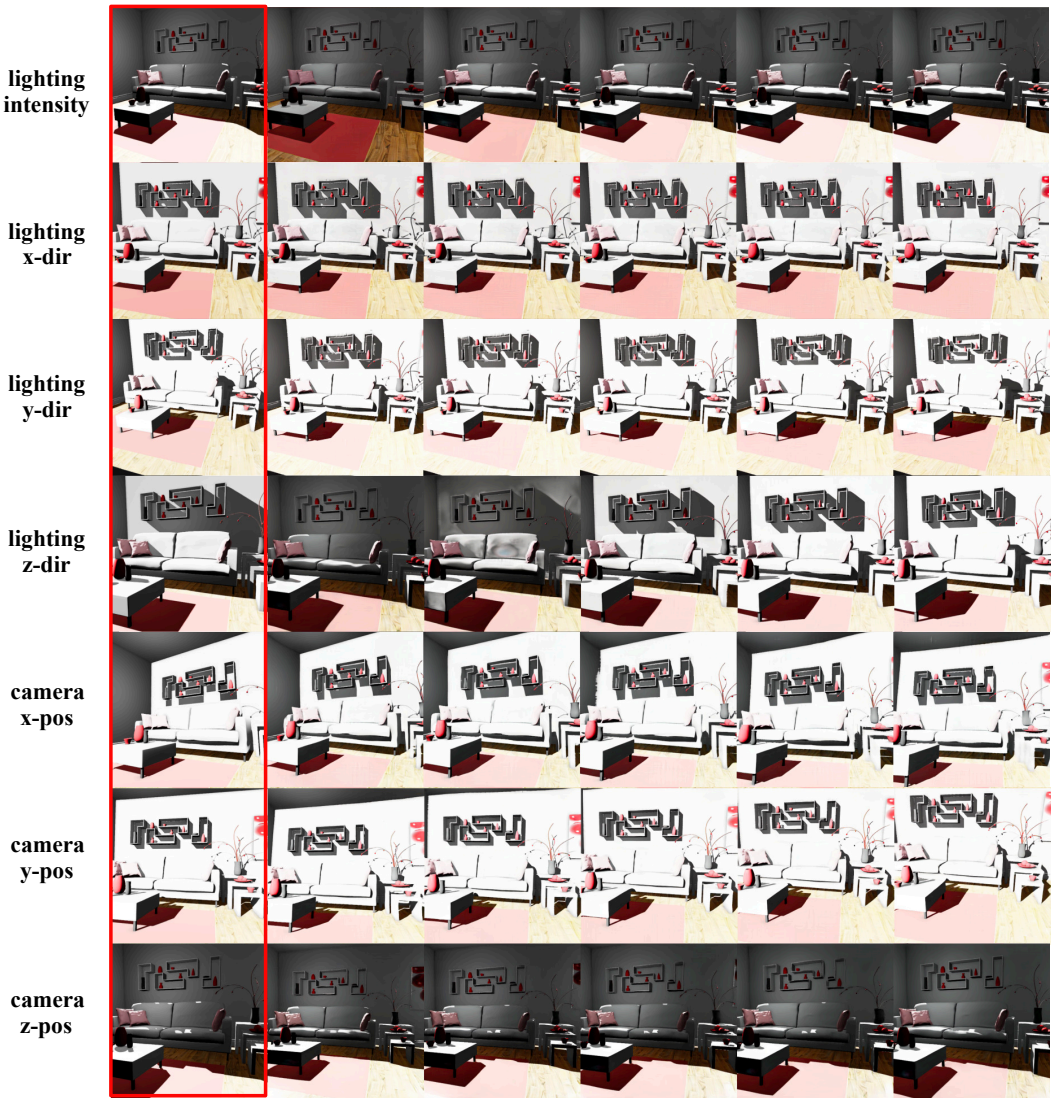


Figure 14. Latent traversal of Semi-StyleGAN on Falcor3D by using 0.5% of the labeled data. Images in the first column (marked by red box) are randomly sampled real images of resolution 512x512 and the rest images in each row are their interpolations, respectively, by uniformly varying the given factor from 0 to 1. We can see that each factor changes smoothly during its interpolation without affecting other factors, and the interpolated images in each row visually look almost the same with their input image except the considered varying factor. Also, the image quality does not get worse during the interpolations.

C.3. Semi-StyleGAN with 0.5% of Labeled Data on CelebA with Resolution 256x256

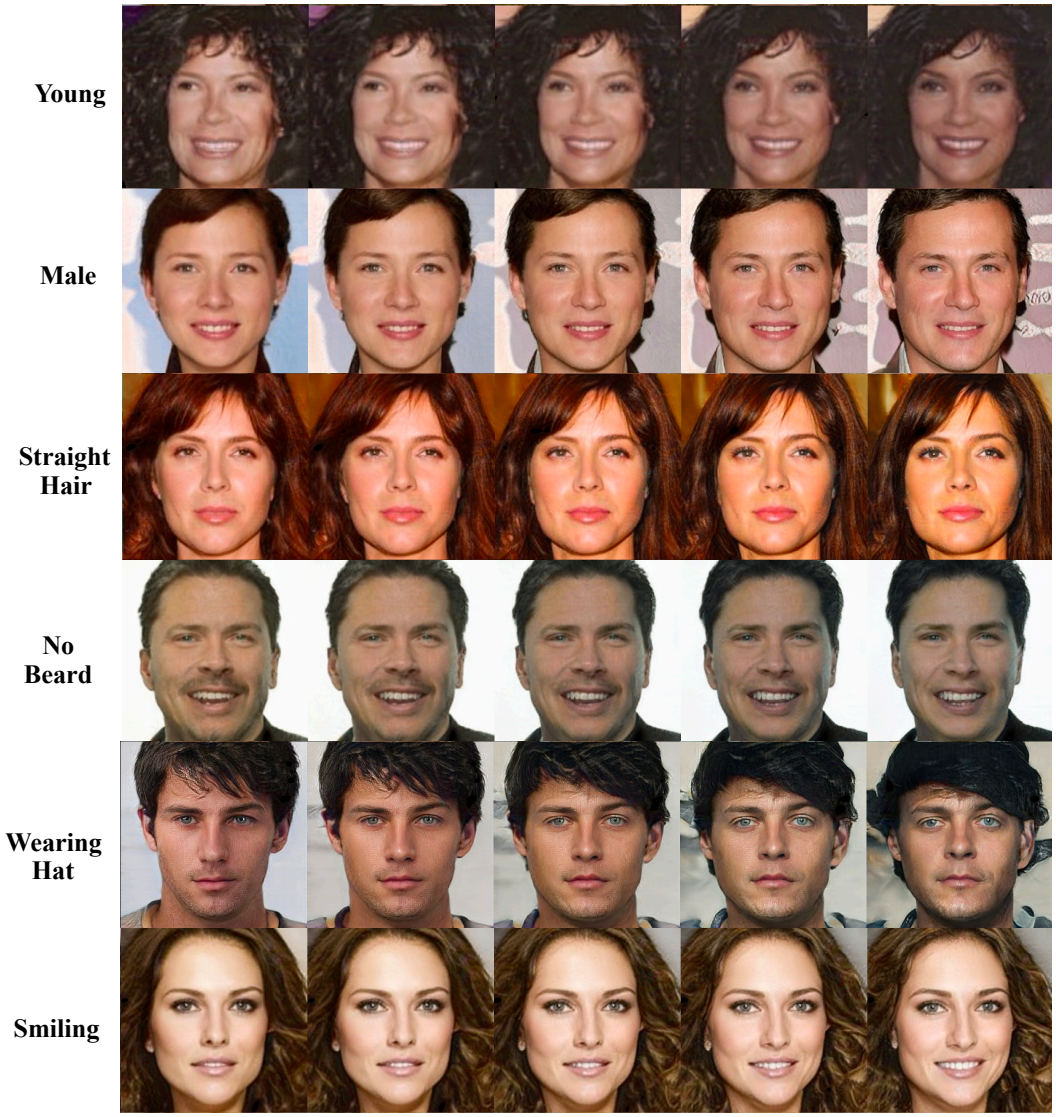


Figure 15. More latent traversal results of Semi-StyleGAN on CelebA with resolution 256x256 by using 0.5% of the labeled data, where we control all 40 binary attributes at the same time. We can see that Semi-StyleGAN with only 0.5% of the labeled data is capable of controlling the considered attributes. We note that the image background may also change slightly over interpolations of some attributions. We argue that it is because the other nuisance factors (i.e., those not in the set of considered 40 attributes) including background strongly confound the observed factors, which has been a common and difficult problem in high-dimensional partially observed latent variables models. We leave the investigation into how to further alleviate this confounding issue as the future work.

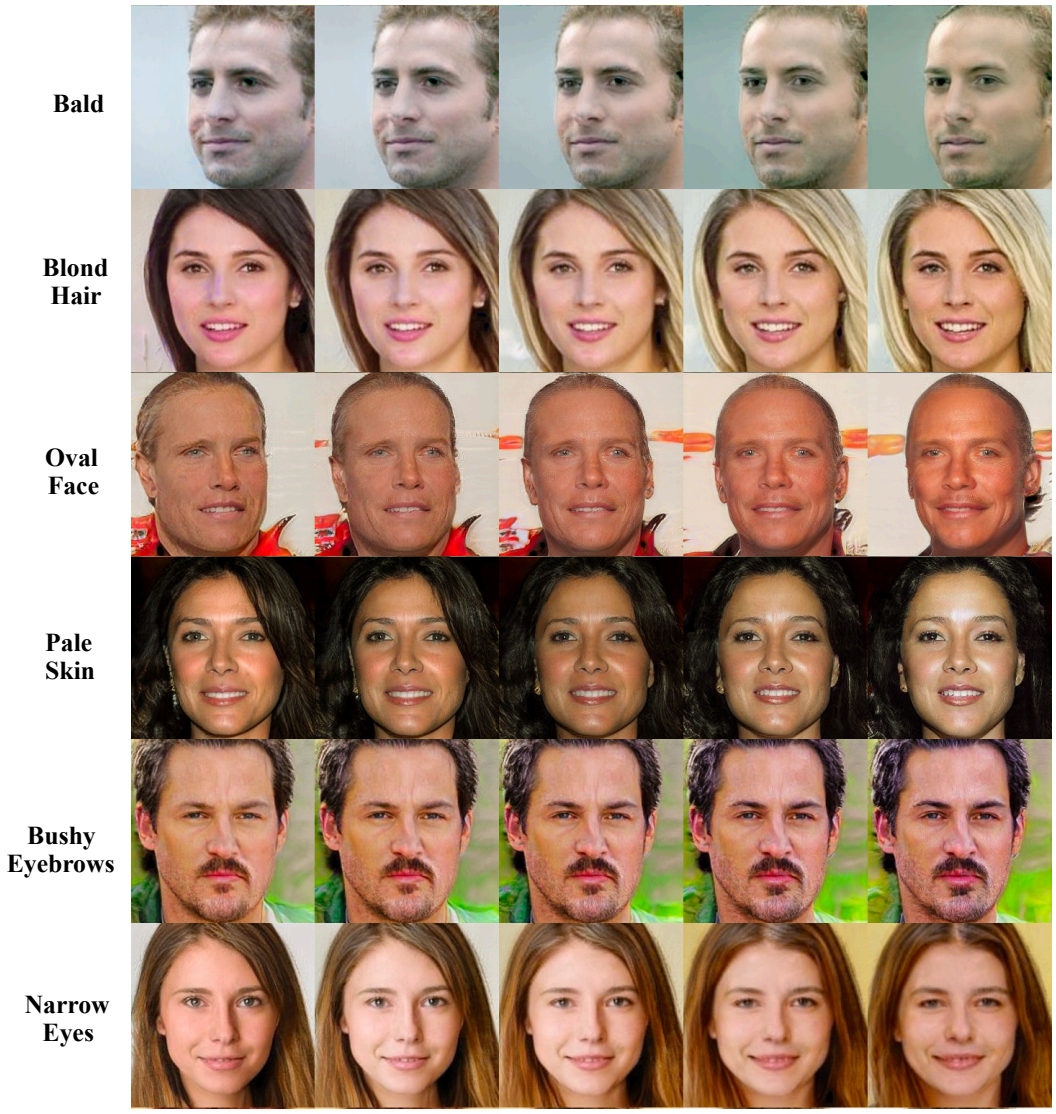


Figure 16. Latent traversal of Semi-StyleGAN on CelebA with resolution 256x256 by using 0.5% of the labeled data, where we control all 40 binary attributes at the same time. We can see that Semi-StyleGAN with only 0.5% of the labeled data is capable of controlling the considered attributes. We note that the image background may also change slightly over interpolations of some attributions. We argue that it is because the other nuisance factors (i.e., those not in the set of considered 40 attributes) including background strongly confound the observed factors, which has been a common and difficult problem in high-dimensional partially observed latent variables models. We leave the investigation into how to further alleviate this confounding issue as the future work.

D. More results on Semi-StyleGAN-*fine*

D.1. Semi-StyleGAN-*fine* with 1% of Labeled Data on Isaac3D Novel Images with Resolution 512x512

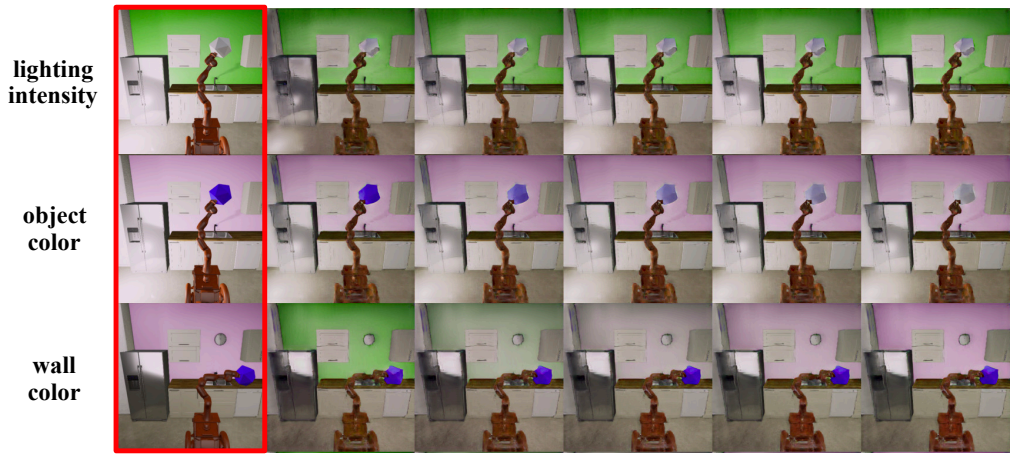


Figure 17. Generalization of Semi-StyleGAN-*fine* with 1% of the labeled data where we set $\phi = 64$ and interpolate three fine styles: (lighting intensity, object color, wall color). In the test image, we shift the position of the robot arm to the right side, and also attach it with an unseen object (i.e., octahedron). Images in the first column (marked by red box) are real novel images of resolution 512×512 and the rest images in each row are their interpolations, respectively, by uniformly varying the given factor from 0 to 1. We can see that Semi-StyleGAN-*fine* with only 1% of the labeled data is capable of controlling the considered fine-grained attributes without affecting the coarse-grained factors.

D.2. Semi-StyleGAN-*fine* with 1% of Labeled Data on CelebA Novel Images with Resolution 256x256

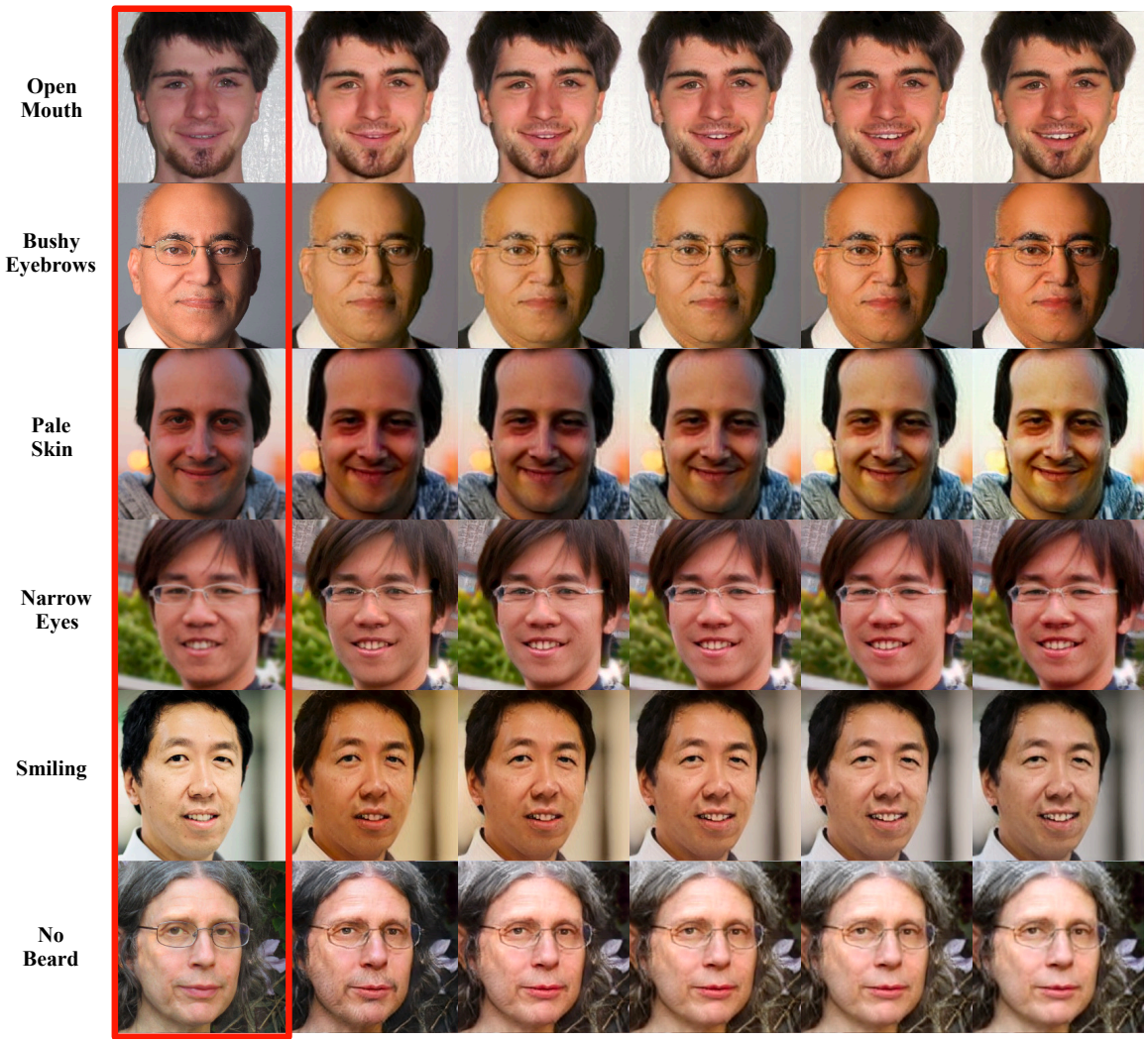


Figure 18. Generalized latent traversal results of Semi-StyleGAN-*fine* trained on CelebA with 1% of labeled data where we set $\phi = 64$ and control the shown fine styles. Images in the first column (marked by red box) are real novel images of resolution 256×256 and the rest images in each row are their interpolations, respectively, by uniformly varying the given factor from 0 to 1. We can see that Semi-StyleGAN-*fine* with only 1% of the labeled data is capable of controlling the considered fine-grained attributes without affecting the coarse-grained factors, in particular the personal identity.