Injecting 3D Perception of Controllable NeRF-GAN into StyleGAN for Editable Portrait Image Synthesis -Supplementary material-

Jeong-gi Kwak[†] Yuanming Li[†] Dongsik Yoon[†] Donghyeon Kim[†]
David Han[‡] Hanseok Ko[†]

[†] Korea University [‡]Drexel Univiersity

6 Appendix: SURF-GAN

In this section, we supplement contents that are not covered in the main paper, i.e., additional details, experiments and discussion about the proposed SURF-GAN.

6.1 Additional implementation details

Training details. The maximum resolution of SURF-GAN, as well as π-GAN [3], which allows stable learning is 64^2 in our setting, it is because 3D-aware GANs (especially pure NeRF-based GAN) require computationally expensive resources for training. Following π-GAN, we adopt the progressive growing strategy that the size of the generated image increases progressively. Unlike 2D GANs, the generator does not actually "grow". Instead, the number of sampled rays increases. Because NeRF-based model can be seen as an implicit continuous function, thus it is theoretically possible to generate arbitrary resolution images. Therefore, only the discriminator adds new layers at each stage to handle higher resolutions. We start training at 32^2 and it is doubled at the next stage. In training phase, the control parameter $\mathbf{z}$ is sampled from the standard normal distribution. The camera pose (pitch and yaw) are sampled from the approximated distribution (normal distribution) of dataset. We assume a perspective pinhole camera where the field of view (FOV) is $12°$. The number of sampled points in each ray is 24 (12 from coarse sampling and 12 from hierarchical sampling). We exploit non-saturating GAN loss with R1 penalty [16] following π-GAN. In addition, there is orthogonal regularization of basis ($\mathcal{L}_{\text{reg}}$) as explained in Sec. 3.1. Finally, pose loss is adopted optionally on different purposes we discuss it in Sec. 6.4. We adopt ADAM [14] optimizer with $\beta_1 = 0$ and $\beta_2 = 0.99$, and the learning rate is set initially to 0.0001 and it is halved in the next stage.

SURF-GAN architecture. The architecture of SURF-GAN generator is illustrated in Fig. 1. The discriminator is same with that of π-GAN [3] except the last layer. Besides the adversarial term that distinguishes real or fake, there is an

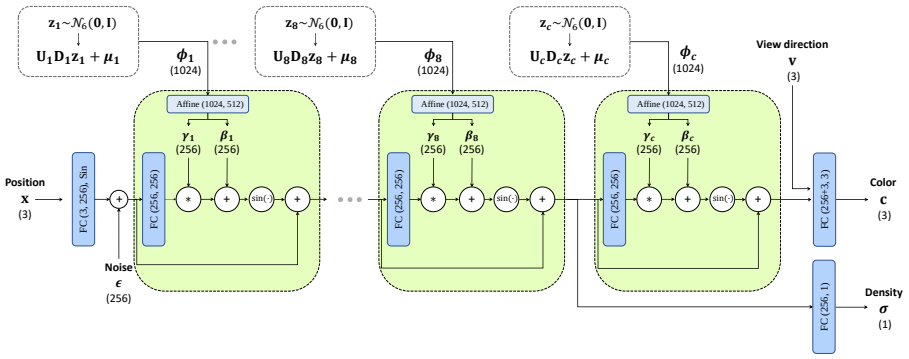


Fig. 1: Details of SURF-GAN generator.

Table 1: Quantitative comparison of SURF-GAN and π-GAN. Each method is trained on 64^2 images and the test is conducted on rendered 128^2 images.

Method	π-GAN		SURF-GAN	
Dataset	CelebA	FFHQ	CelebA	FFHQ
FID (↓)	29.54	47.12	**28.88**	**44.56**
Pose err.($\times 10^{-2}$) (↓)	**5.81**	3.35	6.10	**2.69**
ID (↑)	0.65	0.63	**0.68**	**0.66**
Runtime (↓)	**0.10**		0.11	

additional branch that predicts the pose (i.e., pitch and yaw) of an input image. This branch is utilized if the pose loss is adopted, otherwise it is discarded (same as π-GAN).

6.2 Comparison with π-GAN

We present the comparison results of SURF-GAN with its baseline, π-GAN (Table. 1). Both approaches belong to pure NeRF-GAN, which consists of NeRF networks without following 2D layers. We evaluate FID score [9], pose accuracy, and multi-view consistency of each method. We compute FID score between 50k of generated images and 70k of real images in each dataset. Off-the-shelf 3D model (3DDFA [25]) is utilized to evaluate pose accuracy. The reported pose error (Pose err.) is calculated by averaging the difference between target poses and predicted estimated poses. Multi-view consistency (ID) is evaluated by calculating cosine similarity between canonical view image and others from ArcFace [4]. Although π-GAN shows slightly better results in the pose accuracy of CelebA and runtime, SURF-GAN delivers competitive and superior results. For both models, the increased pose error in CelebA is expected to be due to an alignment issue.

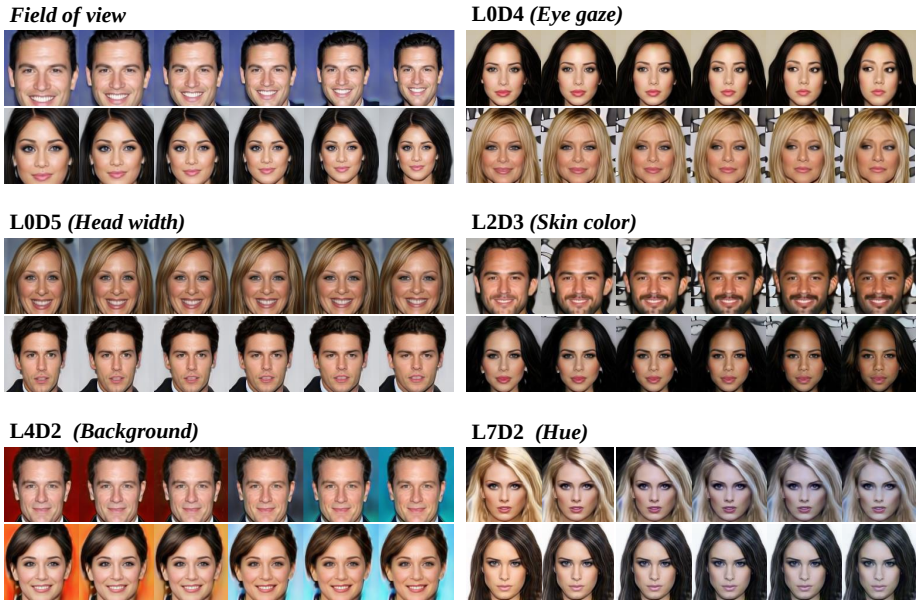


Fig. 2: Additional attributes discovered by SURF-GAN which are not presented in the main paper (CelebA).

6.3 Additional discovered attributes by SURF-GAN

We present the additional attributes in CelebA dataset which are not introduced in the main paper (Fig. 2). Note that "Field of view" is not an discovered attribute, but can be controlled in volume rendering. We also report the semantic attributes of FFHQ in Fig. 3.

6.4 Discussion

Effect of the bottom noise. In addition to the layer-wise latent $\mathbf{z}$, our generator also takes the bottom noise ϵ as an additional input to capture missing variations (Sec. 3.1). Therefore, the intended role for ϵ is to capture the minor variations that have less or not semantic meaning but enhance diversity. Fig. 4 presents generation results when changing only ϵ. As can be seen, the generator synthesizes the images with minor variation while preserving facial identity.

Effect of the progressive growing. As mentioned in Sec. 6.1, we adopt the progressive growing for training. To demonstrate the effectiveness of the strategy, we report FID score of the variants of our method (i.e., w./ and w.o./ progressive growing) for every 1000 iterations. As can be seen the FID curve in Fig. 5, there is a gap between two variants.

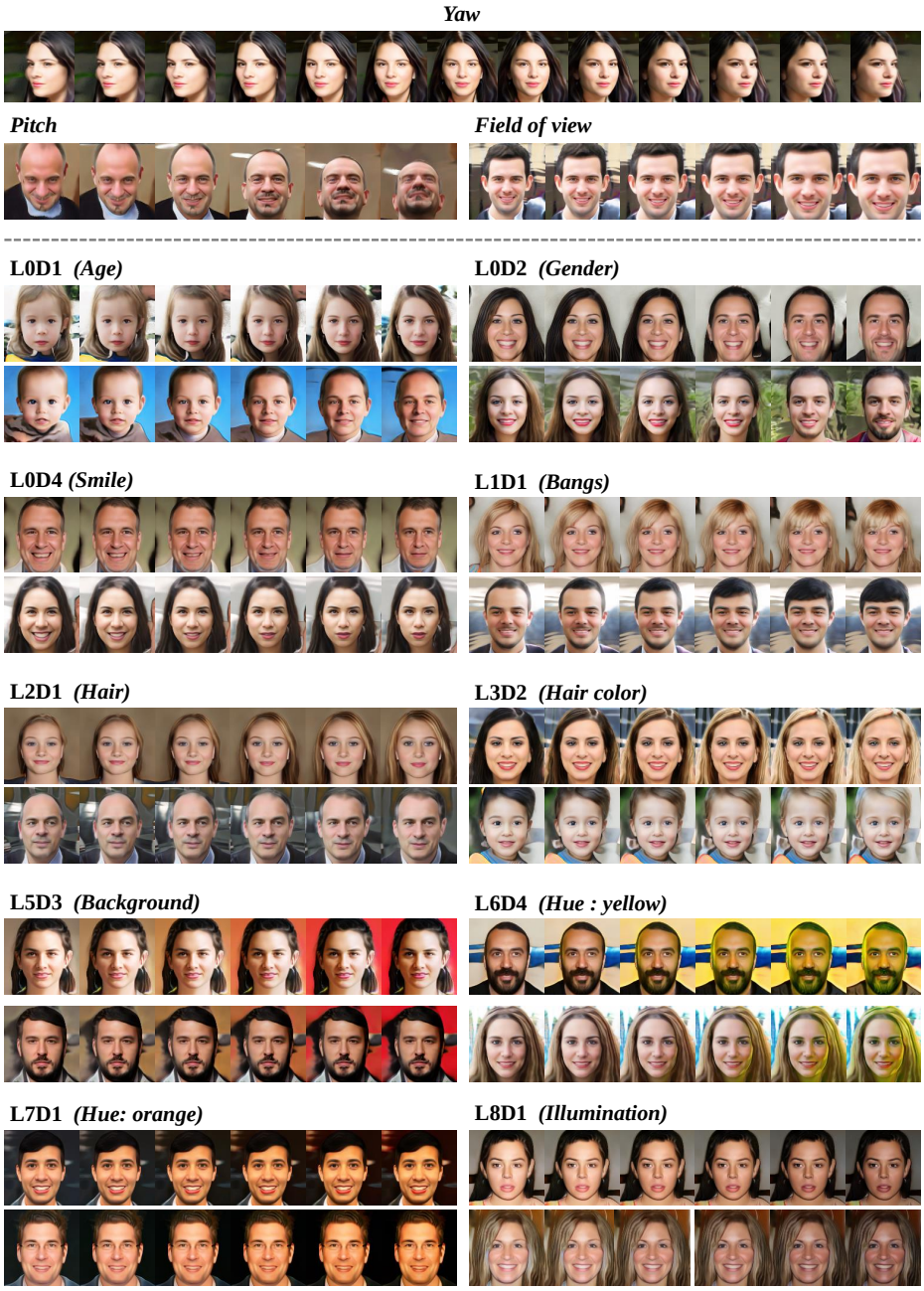


Fig. 3: Semantic attributes discovered by SURF-GAN when using FFHQ dataset.

Fig. 4: The bottom noise ϵ captures subtle variations (64×64 images).

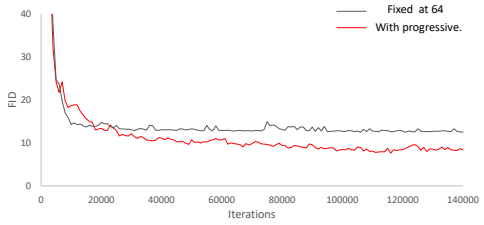


Fig. 5: FID curve of two variants of SURF-GAN, i.e., with and without progressive growing on CelebA (64×64).

Table 2: Ablation study for training SURF-GAN with and without the pose loss on FFHQ (128×128).

	w.o./ $\mathcal{L}_{\text{pose}}$	w./$\mathcal{L}_{\text{pose}}$
FID (↓)	**44.56**	45.92
Pose err.($\times 10^{-2}$) (↓)	2.69	**2.36**

Effect of the pose loss. To improve the pose accuracy, we additionally adopt pose loss $\mathcal{L}_{\text{pose}}$ for training and compare with the original SURF-GAN (w.o./ $\mathcal{L}_{\text{pose}}$). $\mathcal{L}_{\text{pose}}$ is calculated as the difference between input viewing directions of generator and those predicted by the discriminator. It is not an adversarial loss, thus both the generator and discriminator learn to minimize the loss. The results are listed in Table. 2. The introduction of $\mathcal{L}_{\text{pose}}$ reduces pose error (Pose err.), but sacrifices the visual quality. We exploit this model (w./ $\mathcal{L}_{\text{pose}}$) for training 3D-controllable StyleGAN (Sec. 3.2) to offer more elaborate pose samples.

6.5 Limitation.

Although SURF-GAN has several clear advantages such as controllability, there are still inherent limitation as like other 3D-aware GANs. In our model, the color and density of all the points in the rays are calculated independently, thus the amount of computation required to synthesize images increases exponentially as the resolution increases. Such issue has been the catalyst for introducing a method of injecting the prior of SURF-GAN into an efficient and expressive StyleGAN2 generator [13]. It will be one of our future work to achieve high-resolution with clever and efficient ways, e.g., adopting 2D modules [2, 7, 18, 6]in SURF-GAN generator. The other minor limitation is that the same layer

does not always capture the same properties for each training. For example, even if a specific dimension (e.g., $L3D2$) of the trained SURF-GAN captures hair color, the same dimension might capture different attributes in the newly trained model. It seems natural because our method is based on unsupervised training, but it makes the process of assigning properties of each layer necessary after training.

7 Appendix: 3D-controllable StyleGAN

This section presents additional experiments and discussion about 3D controllable StyleGAN.

7.1 Implementation

Latent mapper. The latent mapper consists of five FC layers. It takes an input vector in $\mathcal{W}+$ space and converts it to a canonical vector with the same size. However, the latent mapper does not edit all elements of 18×512 vector, but edits first four style vectors which have known to related to pose [12, 23] (Sec. 4.1), i.e., the input size of the latent mapper is 4×512. Input feature is firstly flatten to 2048-dimensional vector and then converted to intermediate feature $\in \mathbb{R}^{512}$. After going through three intermediate layers, the feature is converted to the canonical vector $\in \mathbb{R}^{4\times512}$ in the last layer.

Training details. We leverage SURF-GAN as a multi-view image generator to train 3D-controllable StyleGAN. As described in Sec. 3.2, the poses of source and target images are randomly sampled from pre-defined distribution. In order to train diverse pose angles, we sample pitch and yaw from uniform distributions instead of Gaussian distribution, i.e., the value of pitch and yaw are uniformly sampled from $[-30°, 30°]$ and $[-45°, 45°]$, respectively. The resolution of the rendered images is 256^2 that is same with input size of pSp encoder [19]. We use a pretrained StyleGAN2 (1024^2) generator for experiments except for MetFace [10] stylization (here we use a 256^2 generator for transfer learning).

7.2 Discussion

Sub-directions. To demonstrate the effectiveness of exploiting orthogonal directions (sub-directions) described in Sec. 3.2, we introduce an interpolation example in Fig. 6. Among learned sub-directions, we select two directions $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$, where both vectors control yaw. As can be seen in the left side of Fig. 6, they influence almost similarly in small pose variations. However, they shows different interpolation outputs when checking the results by scaling both vectors. It means $\overrightarrow{d_1}$ is more involved than $\overrightarrow{d_2}$ for generation of images with large pose variations.

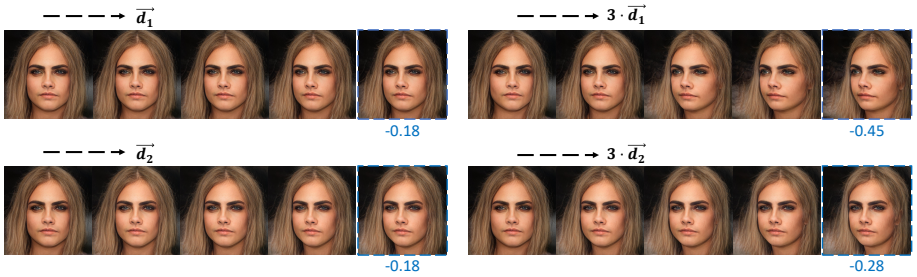


Fig. 6: Example of non-linearity in pose-related vectors. The blue numbers below the figures indicate the detected poses.

Table 3: Results of the identity similarity between two decoded images and difference of the estimated poses between SURF-GAN image and the corresponding image decoded by StyleGAN.

	SURF-GAN	Decoded.
ID	0.66	0.73
Pose diff.	0.003	

SURF-GAN-generated images generalization. We train the latent mapper and the learnable directions using SURF-GAN. The objective function is calculated with the images decoded by StyleGAN. However the question that may arise here is "Can SURF-GAN-generated images be generalized to the training process?". To answer the question, we conduct simple experiments. First, we measure the cosine similarity of two decoded images at different pose angles using ArcFace [4] , and also evaluate how much the pose changes in the decoded image using off-the-shelf pose detector. [25]. The former and the latter are for checking whether identity and pose are maintained, respectively. Although there is a domain gap between SURF-GAN and StyleGAN, the two images with the same identity in SURF-GAN domain maintain the same identity in StyleGAN domain as can be seen Table. 3. Moreover, the pose of the SURF-GAN-generated image is hardly changed by the GAN inversion or decoding.

Extreme cases. As mentioned in Sec. 7.1, we set the poses for training within a certain range because there are few images with extreme poses in the FFHQ dataset. Nevertheless, we validated the extreme case by giving a large value beyond the pose range as input and observed that there are some cases where plausible images are obtained as shown in Fig. 7.

7.3 Additional comparison results

In this subsection, we supplement extra experimental results and discussion to demonstrate the effectiveness of our method.

Fig. 7: Experimental results on extreme poses beyond the range for training.

3D-controllable image synthesis. We first present the additional qualitative comparison with 3D controllable generative models compared in Sec. 4.3. Here we add one more baseline, DiscoFaceGAN [5] which is based on 3DMM. Although DiscoFaceGAN does not allow explicit control over the camera pose, it can be implicitly manipulated by an input latent. Therefore, we display the interpolated images by appropriately adjusting the angles of the images at both ends.

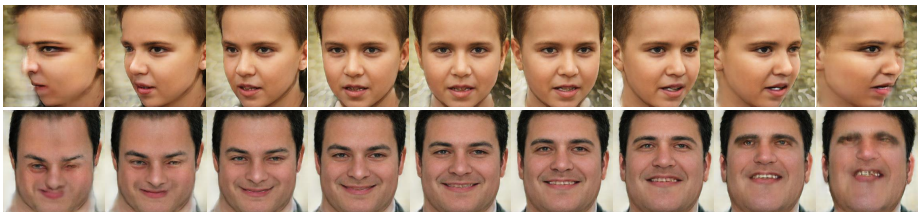
Novel view synthesis. We describe the details not covered in Sec. 4.4 and also present additional qualitative comparison with the competing methods [3, 15, 24] for novel view synthesis (Fig. 9). For all methods, we use the official implementations provided the authors.

π-GAN leverages a latent optimization method [13] to overfit the latent code to the testing image. π-GAN is a 3D-aware generator and learns 3D geometry from unlabelled 2D images without 3D supervision. However, when it is applied to novel view synthesis, π-GAN needs camera extrinsics from the testing image to initiate the following iterative optimization (700 iterations). For the camera pose, we exploit off-the-shelf pose detection method [8]. As shown in Fig. 9, the visual quality deteriorates as it deviates from the original pose. It is difficult to generate the radiance field of the target image only with latent code optimization and a small error in the pose estimation greatly affects the result. In addition, it takes a lot of time (164 sec.) to get results for a single image due to the iterative optimization. This is why we excluded π-gan from the quantitative experiment in Sec. 4.4.

For ConfigNet [15], the real data encoder firstly predicts the latent embedding of a testing image, and then fine-tunes the generator on the image (50 iterations). To handle a real image, it requires pre-processing to align facial images using landmarks from OpenFace [1]. Although it shows an overwhelming runtime in random image generation compared to other methods (Sec. 4.3), the runtime drops significantly due to the introduction of the face detection model in novel view synthesis (Sec. 4.4). Note that the reported runtime in Sec. 4.4 (2.13 sec.) does not include the fine-tuning procedure. The whole process takes about 11.25 seconds (9.12 sec. for fine-tuning). Furthermore, ConfigNet struggles to synthesize images with large pose changes.

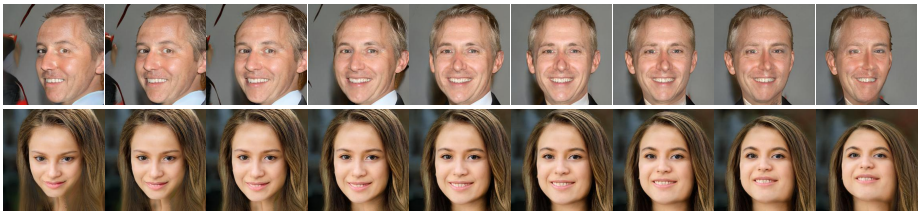
Rotate-and-Render (R&R) is a face rotation method using off-the-shelf 3D fitting network [25] in the overall model, thus it takes some time for 3D fitting (Sec. 4.4). R&R successfully generates a novel view compared to the previously described methods. However, it loses some details of the original image such as hair or background.

Comparison with latent-based models. The pose editing of our method is based on latent manipulation. We introduce comparison results with the existing latent-based method [20] that discovers pose-related direction in the latent space of StyleGAN (Fig. 10). Although InterFaceGAN [20] successfully disentangles the pose attribute, it requires supervision (landmark) for binary classification of yaw in order to find a semantic hyperplane. As Tov et al. [21] have investigated, the results of pose editing with the $\mathcal{W}+$ vectors inverted by pSp [19] shows poor editability. It is alleviated by exploiting e4e encoder [21], but the identity of the input is not well maintained. Above all, the important limitation of latent-based models as well as InterFaceGAN is that they only allow implicit control over pose. Although it is not unfeasible to generate a target pose using these methods, the process might require a few adjustments to obtain an accurate result. Nitzan et al. [17] have introduced the latent-based linear regression method by showing yaw angle has a linear relationship with the distance from InterFaceGAN's yaw hyperplane. Nevertheless, the linearity is not always guaranteed (Fig. 6), and obtaining the hyperplane requires supervision as mentioned above. There may be an clever alternative to acquire the hyperplane without supervision by leveraging the concept of flipping image [22], but it can be applied only to yaw, not other properties such as pitch or field of view.



ConfigNet

DiscoFaceGAN



CIPS-3D

LiftedGAN

Ours

Fig. 8: Additional qualitative comparison with 3D-controllable generative models.

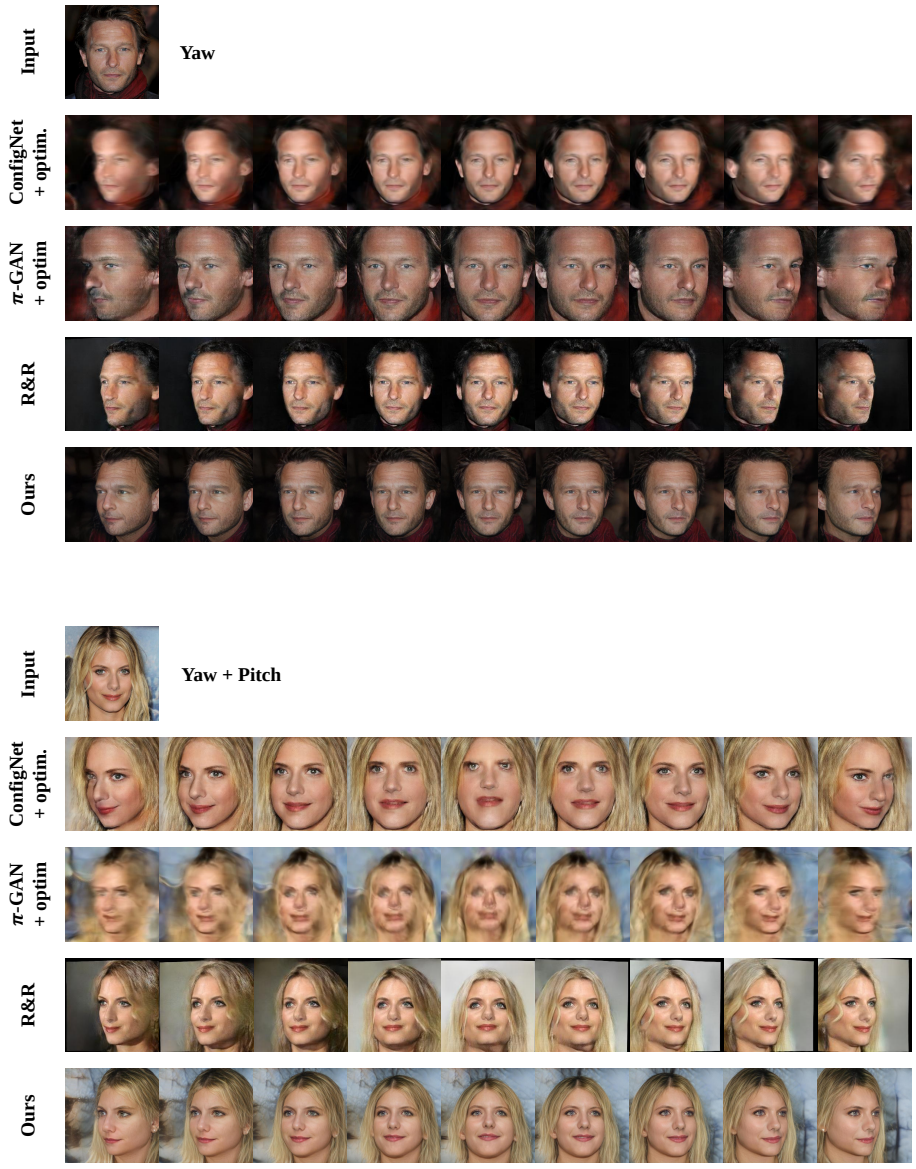


Fig. 9: Additional qualitative comparison with methods which are capable of novel-view synthesis from a single portrait image.

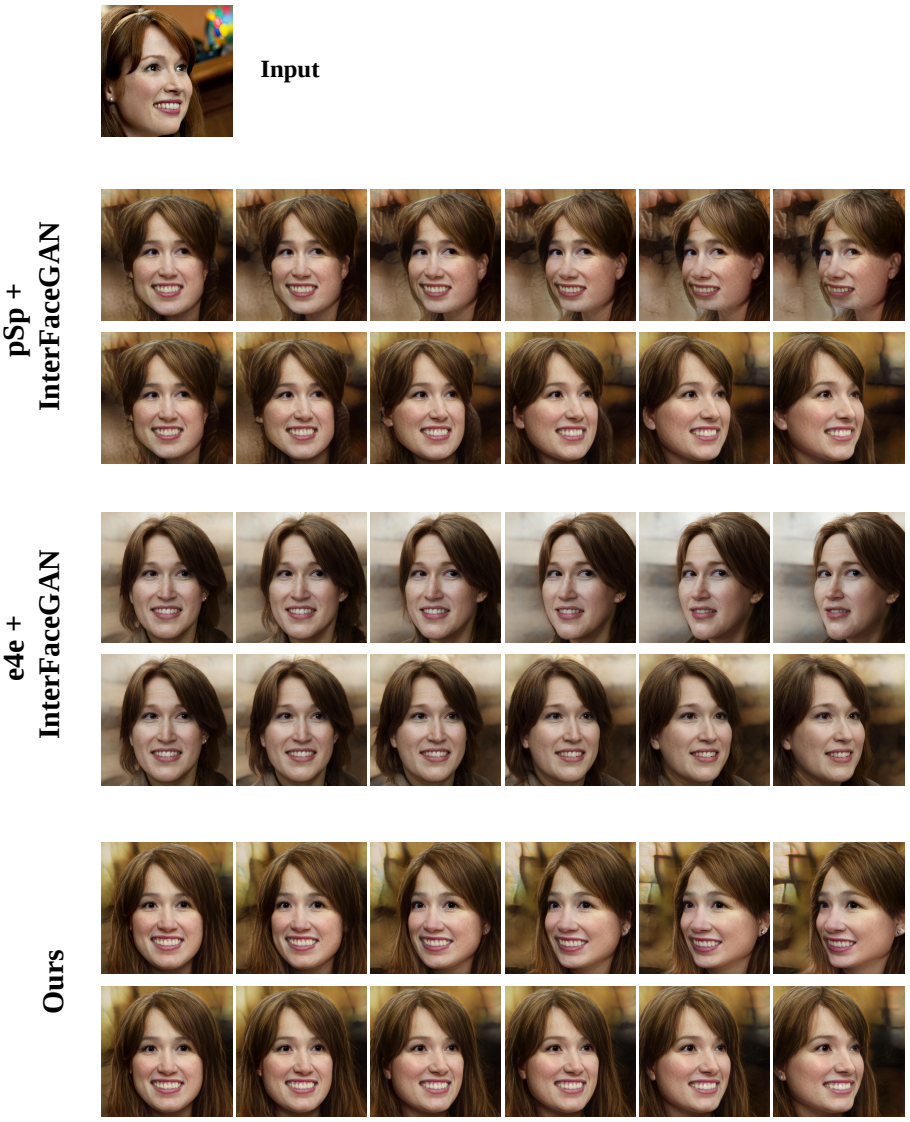


Fig. 10: Comparison with an existing latent manipulation method (InterFaceGAN).

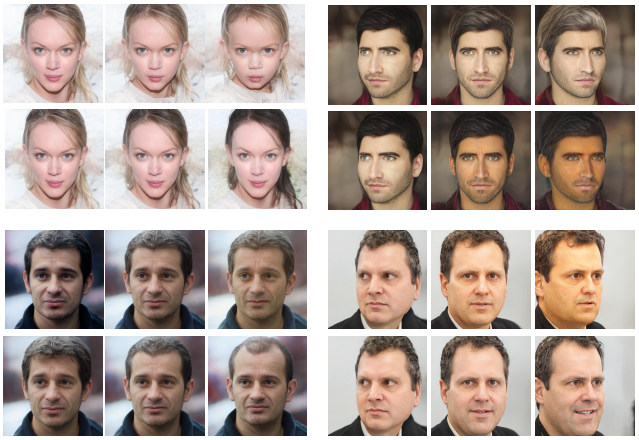


Fig. 11: Editing results of semantic attributes by calculating the direction vector with images generated by SURF-GAN.

Fig. 12: Failure cases of our method (red box). For FOV control, we additionally used FOV as a conditional input for training as well as pitch and yaw.

7.4 Semantic attribute editing

In Sec. 4.5, we presented the results of semantic attribute editing using SURF-GAN, where the direction was calculated by subtracting two inverted SURF-GAN images using pSp encoder [19]. However, there is a trade-off between distortion and editability as Tov et al. [21] demonstrated. As a result, some local attributes in Sec. 4.5 are not changed successfully. Although there might be an effect that we use simple vector arithmetic, the main reason is that the $\mathcal{W}+$ space shows weak editability. To address the issue, we investigate the editing results when using e4e [21] encoder to calculate the direction vector and obtain plausible editing results as shown in Fig. 11. Note that SURF-GAN samples in Fig. 2 and Fig. 3 are utilized for calculating the directions.

7.5 Limitation

Although StyleGAN can generate diverse portrait images with high quality, it struggles to generate out-of-distribution images that do not appear in dataset. Therefore, our method also cannot generate those images because our method does not deviate the latent space of StyleGAN. In addition, our method is also affected by the performance of GAN inversion, thus the performance of our model is not guaranteed for images where the inversion method does not work well. We select several failure cases of our method in Fig. 12. The other limitation is that as like other pose-disentanlged GANs, our method is not capable of generating 3D representations (e.g., mesh or radiance field). Hence, when it comes to video generation, it shows the problem of "texture sticking" [11] (especially in hair) which is one of the most obvious artifacts in GAN generated videos.

References

1. Baltrusaitis, T., Zadeh, A., Lim, Y.C., Morency, L.P.: Openface 2.0: Facial behavior analysis toolkit. In: International conference on automatic face & gesture recognition (FG) (2018)
2. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
3. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
4. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
5. Deng, Y., Yang, J., Chen, D., Wen, F., Tong, X.: Disentangled and controllable face image generation via 3d imitative-contrastive learning. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
6. Deng, Y., Yang, J., Xiang, J., Tong, X.: GRAM: Generative radiance manifolds for 3d-aware image generation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
7. Gu, J., Liu, L., Wang, P., Theobalt, C.: StyleNeRF: A style-based 3d-aware generator for high-resolution image synthesis. arXiv (2021)
8. Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z., Li, S.Z.: Towards fast, accurate and stable 3d dense face alignment. In: European Conference on Computer Vision (ECCV) (2020)
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
10. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (ICLR) (2018)
11. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
12. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
13. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014)
15. Kowalski, M., Garbin, S.J., Estellers, V., Baltrušaitis, T., Johnson, M., Shotton, J.: Config: Controllable neural face image generation. In: European Conference on Computer Vision (ECCV) (2020)
16. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: International Conference on Learning Representations (ICLR) (2018)
17. Nitzan, Y., Gal, R., Brenner, O., Cohen-Or, D.: Large: Latent-based regression through gan semantics. arXiv (2021)

18. Or-El, R., Luo, X., Shan, M., Shechtman, E., Park, J.J., Kemelmacher-Shlizerman, I.: StyleSDF: High-resolution 3d-consistent image and geometry generation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
19. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
20. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
21. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. ACM Transactions on Graphics (TOG) (2021)
22. Wu, S., Rupprecht, C., Vedaldi, A.: Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
23. Zhang, Y., Chen, W., Ling, H., Gao, J., Zhang, Y., Torralba, A., Fidler, S.: Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. arXiv (2020)
24. Zhou, H., Liu, J., Liu, Z., Liu, Y., Wang, X.: Rotate-and-render: Unsupervised photorealistic face rotation from single-view images. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
25. Zhu, X., Liu, X., Lei, Z., Li, S.Z.: Face alignment in full pose range: A 3d total solution. Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2017)