Exp-GAN: 3D-Aware Facial Image Generation with Expression Control – Supplementary Material –

Yeonkyeong Lee[1], Taeho Choi[1], Hyunsung Go[2], Hyunjoon Lee[1], Sunghyun Cho[3], and Junho Kim[2]

[1]Kakao Brain [2]Kookmin University [3]POSTECH

https://github.com/kakaobrain/expgan

S.1 Additional Details on Training and Evaluation

Training Details We train our model with batch size of 12 (per GPU). The learning rates for G_{vol}, G_{face} and G_{img} are set as 0.0025, 0.002 and 0.002, respectively. For the discriminators D_{vol} and D_{out}, we use learning rates 0.0002 and 0.002, respectively. The loss weights are set as $\lambda_{\mathbf{p}} = 15$, $\lambda_{\text{r1}} = 10$, $\lambda_{\text{MSE}} = 1$, and $\lambda_{\text{opacity}} = 10$. Following [9,2], we use the Adam optimizer to train all the generator and discriminator networks with $(\beta_1, \beta_2) = (0.0, 0.9)$. We train the network for 400 epochs in total. It took about seven days to train our network with eight Tesla V100 GPUs.

Details of Comparison with DiscoFaceGAN [5] Since DiscoFaceGAN [5] uses the Basel Face Model (BFM) [11] differently from the others, we establish the following experimental setting for fair comparison with DiscoFaceGAN in Sec. 4.1 in the paper. Specifically, in the evaluation, we estimate both BFM and DECA blendshape coefficients for each image in the FFHQ dataset. We then use the BFM coefficients for DiscoFaceGAN to generate a synthetic image, and estimate DECA coefficients from it. We then measure the difference between the DECA coefficients estimated from the original FFHQ image and from the generated image to evaluate how faithfully the facial expression in the original image is reconstructed in the generated image.

Details on the Multi-View Consistency (MV) Metric Fig. S1 illustrates the evaluation process for multi-view consistency in Sec. 4.1 in the paper. To measure multi-view consistency, we first render nine images by rotating the view direction from left to right. Then, we obtain five reference images by sampling every other image, and reconstruct the other four in-between view images using IBRNet [17] with the reference images. We finally evaluate the difference between the reconstructed images and the original in-between view images in PSNR to measure the multi-view consistency.

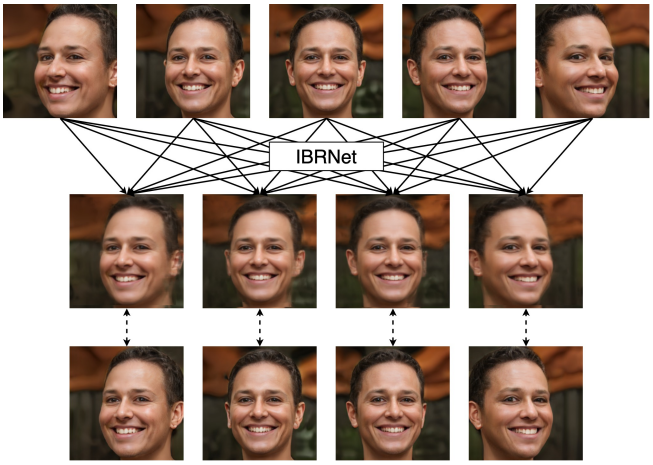

Fig. S1. Evaluation process for multi-view consistency. From five reference images generated by our method (top row), we reconstruct four in-between views using IBRNet [17] (middle row). The reconstructed results are compared with their corresponding targets, again generated with our method (bottom row).

Fig. S2. Results with π-GAN as the neural volume generator. Our Exp-GAN is less sensitive to the choice of the backbone for the neural volume generator.

S.2 Additional Ablation Study

Different Backbone for Neural Volume Generator Fig. S2 shows the results when Exp-GAN uses π-GAN [3], instead of EG3D [2], as a backbone for the neural volume generator. As Fig. S2 shows plausible results with π-GAN, we empirically show that Exp-GAN is not tightly coupled with EG3D [2] but can be incorporated with any 3D-aware generative models.

Feature Integration Scheme We conduct an ablation study to investigate the effect of our depth-based feature integration scheme proposed in Sec. 3.3 in the paper. To this end, we build a baseline model by replacing the depth-based feature integration step with naïve feature concatenation that merges facial and volumetric features similarly to [1,7]. As shown in Fig. S3, feature concatenation causes multi-view inconsistency, such as the inconsistent hair. Such visual artifacts with feature concatenation come from the imperfect separation of facial and non-facial features. In contrast, our approach is able to generate more view-consistent results as shown in Fig. S7. Furthermore, as quantitatively reported

Fig. S3. Limitation of naïve feature concatenation in place of our depth-based feature integration. Hair is inconsistently rendered as the camera pose changes.

Fig. S4. Results with and without camera pose conditioning. (Top row) without camera pose conditioning. (Bottom row) our result. Without camera pose information, eyes are not correctly rendered.

in Sec. 4.2 in the paper, our depth-based feature integration scheme achieves a higher FID score than the naïve feature concatenation scheme, proving the superiority of our approach over the baseline.

Camera Pose Conditioning in the Generator Inspired by [2], we regularize our neural facial generator with camera pose conditioning, as described in Sec. 3.1 in the paper. Fig. S4 shows the effect of camera pose conditioning in our model. The ablation of camera pose conditioning improperly handles the biases that correlate camera poses and eye gazes, causing artifacts around the eyes.

S.3 Additional Results

Parameter Control Figs. S5, S6, S7 and S8 show examples of interpolating facial expression coefficient $\boldsymbol{\beta}$, facial shape coefficient $\boldsymbol{\alpha}$, camera pose, and latent vector $\mathbf{z}$, respectively. Notice that our method can independently and explicitly interpolate each parameter while disentangling the other attributes. Another interpolation example that simultaneously controls expression and camera pose is shown in Fig. S9, which demonstrates that our method can control several parameters smoothly and independently.

Fig. S5. Facial expression coefficient interpolation. The expression coefficients are interpolated from *neutral* to *smile*.

Fig. S6. Facial shape coefficient interpolation.

Fig. S7. Camera pose interpolation.

Fig. S8. Latent vector interpolation. Our method keeps facial shapes and expressions the same between interpolation results.

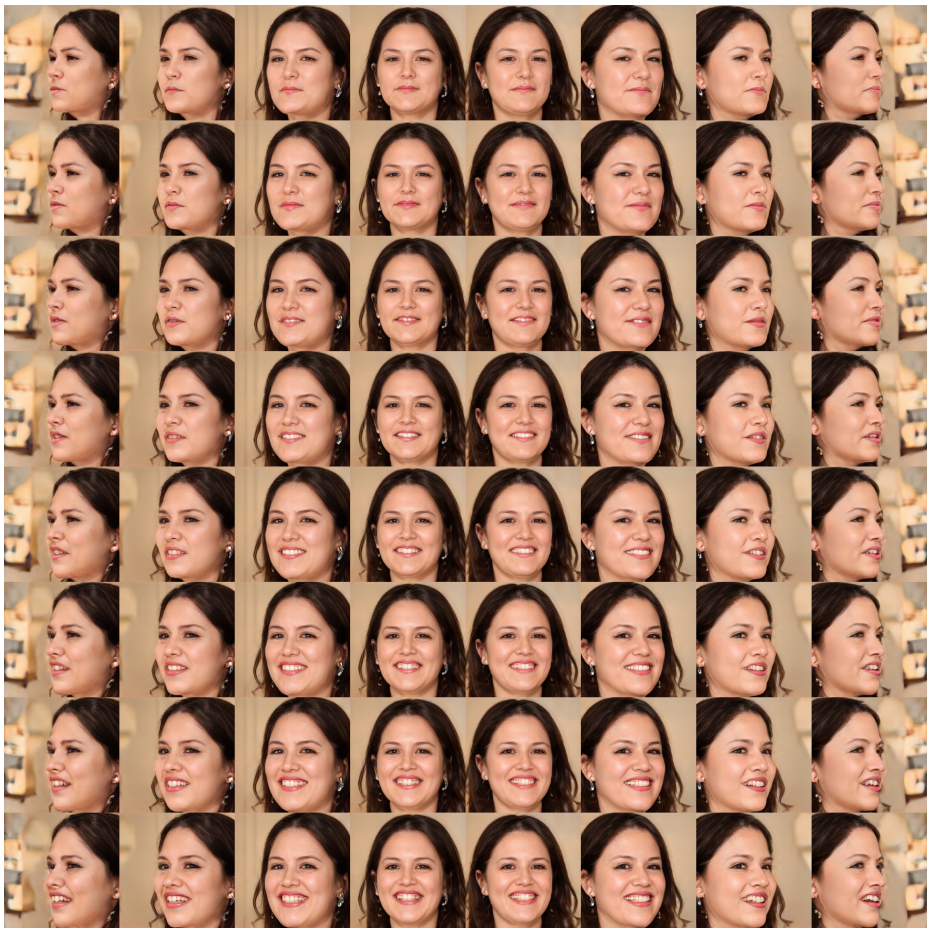

Fig. S9. Expression and camera pose control. (x-axis) camera pose; (y-axis) facial expression.

Fig. S10. COLMAP [13,14] reconstruction examples. In these examples, we generate two sets of 128 images of the same identity with different facial expressions (neutral for the first row and smile for the third row). The second and fourth rows show point clouds reconstructed from the images on the first and third rows, respectively. The points clouds are rendered in different view points to show the quality of the reconstructed 3D structure. As shown in the examples, thanks to the view consistency of our results, the point clouds are successfully reconstructed.

3D Reconstruction from Generated Images To further verify multi-view consistency of our method, we also run COLMAP [13,14] to extract 3D point cloud from the rendered images, as done in [2]. Fig. S10 shows an example; 3D reconstruction can be successfully performed from the generated images, indicating that our method generates images in a view-consistent way, while supporting control over facial expressions.

Facial Reenactment Similarly to [16,18], our method can be used for facial reenactment. Given an input video, we extract camera pose and blendshape coefficients from each frame of the video and generate images with the extracted parameters. Fig. S14 shows examples of facial reenactment. Different from [16,18], our Exp-GAN is able to freely generate facial avatars from random latent vectors.

Single-view reconstruction with GAN inversion Fig. S11 shows examples of single image 3D reconstruction. Similar to [2], we use the pivotal tuning inversion (PTI) [12] to fit input images into our network. We extract all the condition

Fig. S11. Examples of GAN inversion. From left to right: input image, depth map of the reconstruction, reconstructed image, rendered images with different camera pose and facial expression.

Fig. S12. Additional examples of GAN inversion. From left to right: input image, reconstruction result, rendered from two different camera poses. Images in FFHQ dataset are reconstructed.

parameters $\mathbf{p}$ required for our network from the input image by using DECA [6]. Fig. S12 show additional examples of GAN inversion. We reconstruct 3D avatar from a single input image, then change camera pose and expression.

Comparisons with StyleRig [15] StyleRig [15] presents facial attribute editing over pretrained StyleGAN's intermediate latent space $\mathcal{W}$ with rig-like controls. Inheriting the limitations of StyleGAN, StyleRig shows limited view consistency. Also, as discussed in Sec. 8.2 in [15] and Fig. 6 of [15]'s supplementary material, it struggles synthesizing asymmetrical facial expressions, as StyleRig hardly handles the bias in the distribution of expressions in the training dataset. In contrast, Exp-GAN provides plausible results in the cases of asymmetrical facial expressions which are rare in the FFHQ dataset [8], as shown in Fig. S13. Notice that Exp-GAN synthesizes natural asymmetrical facial expressions around the mouth and eyebrows using 3DMM-controlled blendshape coefficients.

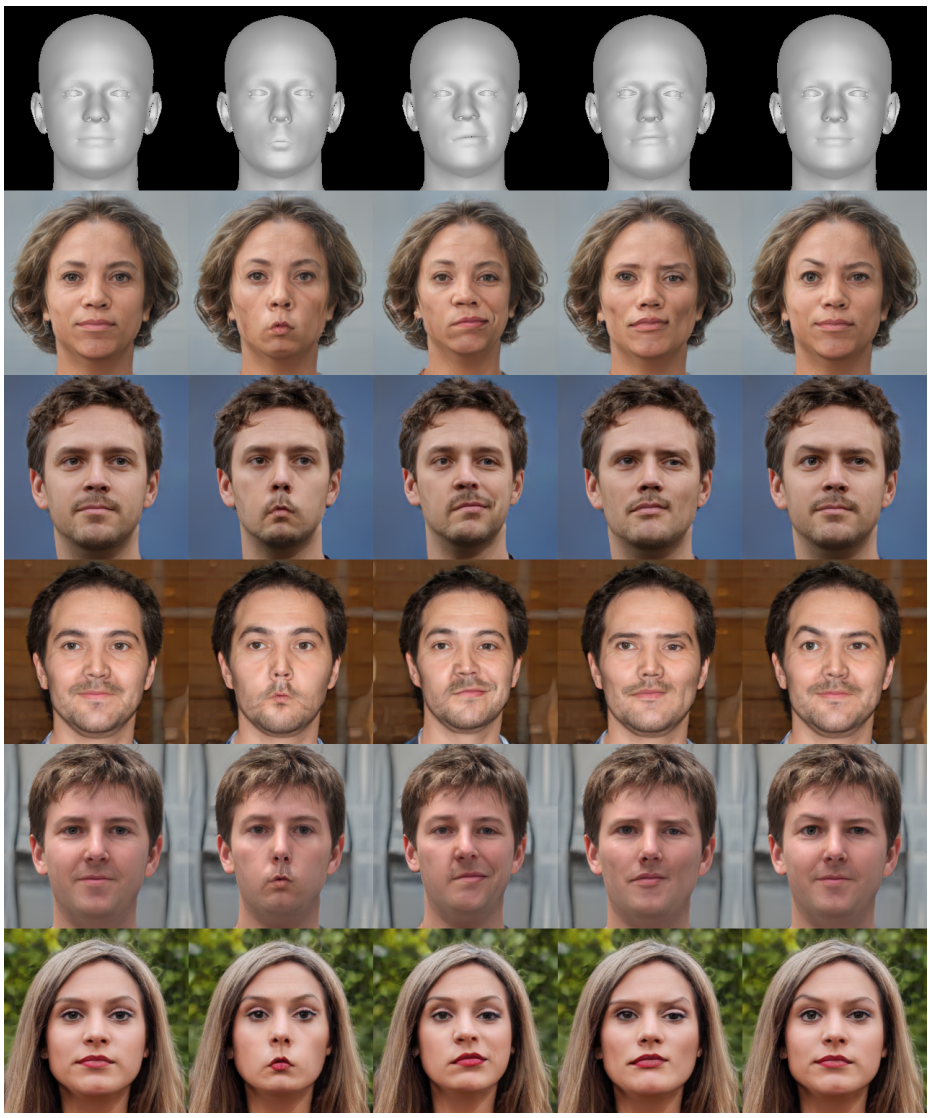

Fig. S13. Our Exp-GAN can synthesize facial expressions that are rare in the training dataset [8]. (1st column) neutral expression as a baseline, (2nd column) pouty mouth, (3rd–5th columns) asymmetrical facial expression around the mouth and eyebrows.

Table S1. Ablation study of the loss terms using the view direction trick [10]. The scores in the parentheses are those of Table 1 in the paper.

	FID ↓	BS ↓	MV ↑	ID ↑
Ours (full)	(7.44)	(0.05)	23.98 (23.84)	0.642 (0.622)
w/o blendshape coeff. reg.	(8.90)	(0.10)	26.05 (25.93)	0.764 (0.723)
w/o $\mathcal{L}_{\text{MSE}}$	(12.08)	(0.05)	23.64 (23.30)	0.641 (0.628)
w/o $\mathcal{L}_{\text{opacity}}$	(10.53)	(0.05)	24.08 (23.96)	0.624 (0.617)

Comparisons with SofGAN [4] Although SofGAN [4] is *not* designed for expression controls, we can evaluate how SofGAN well reflect expressions in generated image, as follows. We extract the samples of blendshape coefficients from the FFHQ dataset, as described in Sec. 4.1 in the paper. Then, we use the *mask images of FFHQ* [4] for the Rendered Segmaps of SofGAN to synthesize output images. Next, we re-estimate blendshape coefficients from the output images. We finally measure the BS metric, which is the mean squared distance between input and re-estimated blendshape coefficients. In experiments, we obtained the BS score of 0.08 for SofGAN, which is worse than ours (see Table 1 in the paper).

SofGAN [4] shows limited view consistency as well as a texture sticking problem. Although SofGAN generates a volumetric semantic occupancy field (SOF), the Rendered Segmap of SOF, obtained by volume ray casting, is a semantic class map with *discrete* class labels. As a result, Rendered Segmaps between coherent viewpoints cannot contain the coherently projected 3D features in the image space that are crucial to obtain view consistency. For the multi-view consistency of SofGAN we obtained the MV score of 22.332, which is worse than ours (see Table 1 in the paper).

View Direction Trick We can use the view direction trick proposed in [10] to obtain better metric scores, as shown in Table S1. Specifically, during the inference time, we use the camera pose fixed to the frontal view for the neural face and volume generators for better identity consistency. Then, we render an image for the input camera pose at the image synthesis module. Table S1 shows slightly better MV and ID scores in all ablation settings, compared to Table 2 in the paper. FID and BS scores in Table S1 are identical to those in Table 2 in the paper, since they are not measured with changing view directions. To clearly report the 3D-awareness of our Exp-GAN, all generated images and metric scores in the paper and the supplementary material are provided *without* using the view directional trick.

S.4 Uncurated Results and Videos

Fig. S15 shows uncurated results. We also provide a set of videos to further demonstrate our Exp-GAN in the project page.

Fig. S14. Facial reenactment. (Top row) reference video frames; (second row) rendered face meshes visualizing blendshape coefficients and camera poses; (third to last rows) generated facial avatars.

Fig. S15. Uncurated examples generated with latent vectors $\mathbf{z}$ randomly sampled with random seeds $[0, 6]$. While the camera pose changes, facial expressions of each example are controlled as (1st column) neutral, (2nd column) smile, (3rd column) open mouth, (4th column) half-open mouth, and (5th column) frown, respectively.

References

1. Bühler, M.C., Meka, A., Li, G., Beeler, T., Hilliges, O.: VariTex: Variational Neural Face Textures. In: Proc. ICCV. pp. 13890–13899 (2021)
2. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient Geometry-aware 3D Generative Adversarial Networks. In: Proc. CVPR. pp. 16123–16133 (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: Proc. CVPR. pp. 5799–5809 (2021)
4. Chen, A., Liu, R., Xie, L., Chen, Z., Su, H., Yu, J.: SofGAN: A Portrait Image Generator with Dynamic Styling. ACM Trans. Graphics **42**(1) (2022)
5. Deng, Y., Yang, J., Chen, D., Wen, F., Tong, X.: Disentangled and Controllable Face Image Generation via 3D Imitative-Contrastive Learning. In: Proc. CVPR. pp. 5154–5163 (2020)
6. Feng, Y., Feng, H., Black, M.J., Bolkart, T.: Learning an Animatable Detailed 3D Face Model from In-The-Wild Images. ACM Trans. Graphics (Proc. SIGGRAPH 2021) **40**(8), Article No. 88 (2021)
7. Ghosh, P., Gupta, P.S., Uziel, R., Ranjan, A., Black, M., Bolkart, T.: GIF: Generative Interpretable Faces. In: Proc. 3DV. pp. 868–878 (2020)
8. Karras, T., Laine, S., Aila, T.: A Style-Based Generator Architecture for Generative Adversarial Networks. In: Proc. CVPR. pp. 4401–4410 (2019)
9. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and Improving the Image Quality of StyleGAN. In: Proc. CVPR. pp. 8110–8119 (2020)
10. 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: Proc. CVPR. pp. 13503–13513 (2022)
11. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D Face Model for Pose and Illumination Invariant Face Recognition. In: IEEE International Conference on Advanced Video and Signal Based Surveillance. pp. 296–301 (2009)
12. Roich, D., Mokady, R., Bermano, A.H., Cohen-Or, D.: Pivotal tuning for latent-based editing of real images. arXiv preprint arXiv:2106.05744 (2021)
13. Schönberger, J.L., Frahm, J.M.: Structure-from-Motion Revisited. In: Proc. CVPR. pp. 4104–4113 (2016)
14. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise View Selection for Unstructured Multi-View Stereo. In: Proc. ECCV. pp. 501–518 (2016)
15. Tewari, A., Elgharib, M., Bharaj, G., Bernard, F., Seidel, H.P., Pérez, P., Zöllhofer, M., Theobalt, C.: StyleRig: Rigging StyleGAN for 3D Control over Portrait Images. In: Proc. CVPR. pp. 6142–6151 (2020)
16. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In: Proc. CVPR. pp. 2387–2395 (2016)
17. Wang, Q., Wang, Z., Genova, K., Srinivasan, P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: IBRNet: Learning Multi-View Image-Based Rendering. In: Proc. CVPR. pp. 4690–4699 (2021)
18. Wang, T.C., Mallya, A., Liu, M.Y.: One-Shot Free-View Neural Talking-Head Synthesis for Video Conferencing. In: Proc. CVPR. pp. 10039–10049 (2021)