3D GAN Inversion with Pose Optimization
- Supplementary Material -

A. Implementation Details

A.1. Architectures and Hyperparameters

To implement the latent code encoder $\mathcal{E}$, we follow both the implementation and training strategy of e4e [15], which is a well-proven encoder architecture to map the input image into the distribution of $\mathcal{W}+$. We manipulate the output dimension of the encoder into $\mathbb{R}^{1\times 512}$ to fit in our method. For the camera pose estimator $\mathcal{P}$, we manipulate the simple resnet34 [5] encoder to find theta and phi angles which are further calculated into the extrinsic matrix. In the case of dataset which has an additional roll angle component in the rotation such as *cat faces*, we choose the 6D rotation representation proposed by [17]. Although the target images are cropped and refined by [2], there exists an additional camera translation that euler angles cannot thoroughly define. Thus, we set an additional coordinate variance on the camera position as a learnable parameter. See supplementary for details and qualitative evaluation of additional translation.

A.2. Pre-training Latent Encoder $\mathcal{E}$ and Pose Estimator $\mathcal{P}$

In order to train the encoder $\mathcal{E}$, we adopt LPIPS loss [16] denoted as $\mathcal{L}^{\mathcal{E}}_{\text{lpips}}$ in order to reconstruct the given image. Additionally, we minimize the gap between $\bar{\mathbf{w}}+\triangle\mathbf{w}$ and the embedding space $\mathcal{W}$ of $\mathcal{G}_{\text{3D}}$ by employing non-saturating GAN loss [3] and delta-regularization loss:

$$\mathcal{L}^{\mathcal{E}}_{\text{adv}} = -\mathbb{E}[\log\mathcal{D}(\mathcal{G}^{\text{c}}_{\text{3D}}((\bar{\mathbf{w}}+\triangle\mathbf{w}),\pi_{\text{ps}};\theta))], \tag{1}$$

$$\mathcal{L}^{\mathcal{E}}_{\text{reg}} = ||\triangle\mathbf{w}||^2_2. \tag{2}$$

Moreover, in the case of pose estimator $\mathcal{P}$, the predicted output $\hat{\pi}$ from a given image is directly compared with π_{ps}. Let rotation $\hat{\text{R}}\in\mathbb{R}^{3\times 3}$, translation $\hat{t}\in\mathbb{R}^3$, and scale factor $\hat{s}\in\mathbb{R}^1$ denote as a decomposed set of π. Since the scale factor is given as a constant, we formulate our loss function as:

$$\mathcal{L}^{\mathcal{P}}_{\text{rot}} = ||\text{R}^{-1}_{\text{ps}}\cdot\hat{\text{R}}-\text{I}_{3\times 3}||_2, \tag{3}$$

$$\mathcal{L}^{\mathcal{P}}_{\text{trans}} = ||t_{\text{ps}}-\hat{t}||_2, \tag{4}$$

where R_{ps} and t_{ps} are decomposition of π_{ps}, and $\text{I}_{3\times 3}$ is identity matrix.

In summary, our total loss functions ($\mathcal{L}^{\mathcal{E}}$ for Latent Encoder $\mathcal{E}$ and $\mathcal{L}^{\mathcal{P}}$ for Pose Estimator $\mathcal{E}$) for pre-training the encoder are defined by:

$$\mathcal{L}^{\mathcal{E}} = \mathcal{L}^{\mathcal{E}}_{\text{lpips}} + \lambda_{\text{adv}}\mathcal{L}^{\mathcal{E}}_{\text{adv}} + \lambda_{\text{reg}}\mathcal{L}^{\mathcal{E}}_{\text{reg}}, \tag{5}$$

$$\mathcal{L}^{\mathcal{P}} = \mathcal{L}^{\mathcal{P}}_{\text{rot}} + \lambda_{\text{trans}}\mathcal{L}^{\mathcal{P}}_{\text{trans}}. \tag{6}$$

Figure 1: **Generated pseudo ground-truth images for training latent encoder and pose estimator.** We utilize EG3D as a pseudo ground-truth generator to sample images paired with corresponding latent representations and camera viewpoint. These pseudo pairs $\{\mathbf{w}, I_{\text{pseudo}}\}$ and $\{\pi, I_{\text{pseudo}}\}$ are used as training data for latent encoder and pose estimator respectively.

A.3. Pseudo Dataset

We visualize a randomly selected pseudo dataset in Fig. 1. We construct 30,000 number of pseudo pairs for *human faces*, and 10,000 for *cat faces*. Each of the pseudo pairs rendered from a unique latent vector $\mathbf{w}_{\text{ps}}$ with randomly sampled camera pose parameter π_{ps} within the camera pose boundary given by [1]. As can be seen from Fig. 1, the generative power of 3D GANs gives promising guidance both for the latent encoder and pose estimator. In the case of *cat faces*, we additionally gave a *roll* rotation to the pose parameter which gives the generated pseudo images to rotate on the image plane.

A.4. Preparing Facial Images for GAN Inversion.

EG3D [1] uses a slightly different cropping method for their training process. We follow the official code, found here: `https://github.com/NVlabs/eg3d/tree/FFHQ_preprocess` which uses face detection and pose-extraction pipeline [2] in order to identify and crop the face region and infer the camera viewpoint of the image. The inferred camera viewpoint is leveraged in our baseline 2D GAN inversion implementation and is also exploited as a ground-truth dataset to evaluate our pose estimation performance.

A.5. Implementation Details of Comparison Baselines.

The following algorithms show our method, along with comparisons using 2D GAN inversion methods directly to 3D-aware GANs.

Algorithm 1: Our proposed method.

Input: real image x; canonical pose π_c; gradient-based optimizer F'; depth smoothness regularizer DR; depth based reprojection $proj(\cdot)$; camera intrinsics K

Output: the reconstructed image $\hat{y}$

Initialize() the code and pose $(w, \pi) = (w', \pi')$;
while *not converged* **do**
  $Loss \leftarrow L\left(x, \mathcal{G}^{\mathrm{c}}_{3\mathrm{D}}(\mathbf{w}, \pi; \theta)\right)$;
  $w \leftarrow w - \eta F'(\nabla_w Loss)$;
  $\mathrm{projected} \leftarrow \mathcal{G}^{\mathrm{c}}_{3\mathrm{D}}(\mathbf{w}, \pi; \theta)\langle proj(\mathcal{G}^{\mathrm{d}}_{3\mathrm{D}}(\mathbf{w}, \pi; \theta), \pi, K)\rangle$;
  $Loss \leftarrow L\left(\mathcal{G}^{\mathrm{c}}_{3\mathrm{D}}(\mathbf{w}, \pi; \theta)\right), \mathrm{projected},)$;
  $\pi \leftarrow \pi - \eta F'(\nabla_\pi Loss)$;
end
while *not converged* **do**
  $Loss \leftarrow L\left(x, \mathcal{G}^{\mathrm{c}}_{3\mathrm{D}}(\mathbf{w}, \pi; \theta))\right) + DR(\mathcal{G}^{\mathrm{d}}_{3\mathrm{D}}(\mathbf{w}, \pi; \theta))$;
  $\theta \leftarrow \theta - \eta F'(\nabla_\theta Loss)$;
end
$\hat{y} \leftarrow \mathcal{G}^{\mathrm{c}}_{3\mathrm{D}}(\mathbf{w}, \pi; \theta)$

Algorithm 2: GT camera pose during optimization.

Input: real image x from viewpoint π^*; a pre-trained generator $\mathcal{R}(\cdot, G(\cdot; \theta))$; gradient-based optimizer F'.

Output: the latent code w

Initialize() the code $w = w'$;
while *not converged* **do**
  $Loss \leftarrow L\left(x, \mathcal{G}^{\mathrm{c}}_{3\mathrm{D}}(\mathbf{w}, \pi^*; \theta)\right)$;
  $w \leftarrow w - \eta F'(\nabla_w L)$;
end
if *pivotal tuning* **then**
  while *not converged* **do**
    $Loss \leftarrow L\left(x, \mathcal{G}^{\mathrm{c}}_{3\mathrm{D}}(\mathbf{w}, \pi^*; \theta)\right)$;
    $\theta \leftarrow \theta - \eta F'(\nabla_\theta L)$;
  end
end

Algorithm 3: Gradient descent to optimize camera.

Input: real image x; a pre-trained generator $\mathcal{R}(\cdot, G(\cdot; \theta))$; gradient-based optimizer F'.

Output: the latent code w

Initialize() the code and pose $(w, \pi) = (w', \pi')$;
while *not converged* **do**
  $Loss \leftarrow L\left(x, \mathcal{G}^{\mathrm{c}}_{3\mathrm{D}}(\mathbf{w}, \pi; \theta)\right)$;
  $(w, \pi) \leftarrow (w, \pi) - \eta F'(\nabla_{w,\pi} L)$;
end
if *pivotal tuning* **then**
  while *not converged* **do**
    $Loss \leftarrow L\left(x, \mathcal{G}^{\mathrm{c}}_{3\mathrm{D}}(\mathbf{w}, \pi; \theta)\right)$;
    $\theta \leftarrow \theta - \eta F'(\nabla_\theta L)$;
  end
end

B. Discussion

B.1. Difficulties of 3D GAN inversion

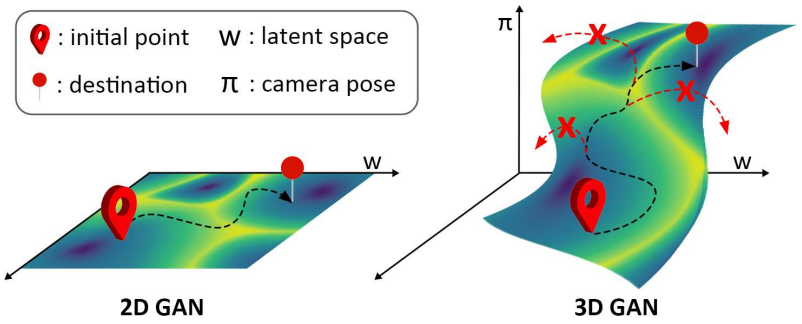


Figure 2: Comparison of the latent spaces of 2D GANs and 3D GANs

As demonstrated in the main paper, 3D GAN inversion is non-trivial, since one struggles to optimize if the other is inaccurate. As illustrated in Fig. 2, optimizing latent features in 3D GANs gives an additional consideration to pose optimization, which needs to be optimized simultaneously. Specifically, this becomes difficult when either latent feature or camera pose is imperfect. In Fig. 3, we show failure cases to illustrate how the imperfect camera pose estimation leads to shape distortion. The second column of Fig. 3 is the final inversion result without pivotal tuning. As the pose estimation fails, the second column shows misalignment in the head pose. Even though the pivotal tuning step resolves the visual perception error, they often fail on novel view synthesis (4-6 columns). This is because the tuning procedure forces the implicit volume to fit into an input image with a misaligned camera parameter, which becomes distortions to the other viewpoints.

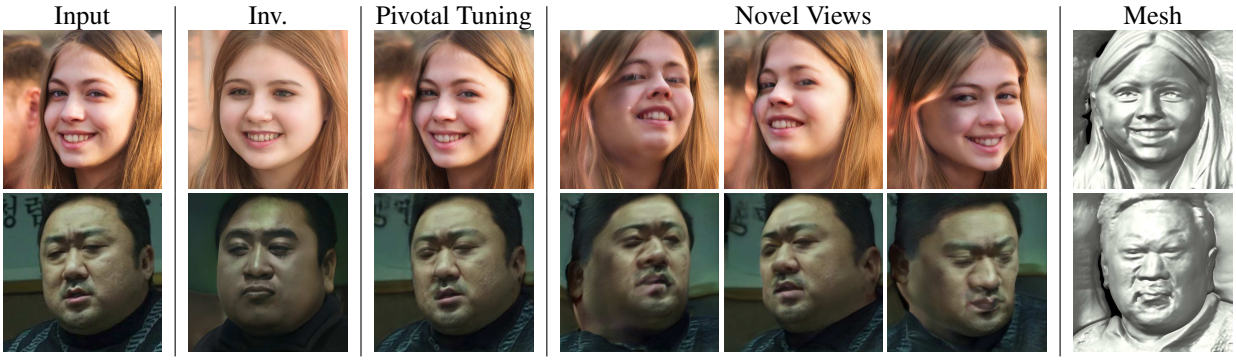


Figure 3: **Importance of accurate camera pose estimation.**

B.2. Camera Misalignment

Furthermore, we conduct another ablation study on camera misalignment, illustrated in Fig. 4. The top row shows an inversion process using our training strategy, while the bottom row does not optimize the additional trainable translation vector. Following the preprocessing stage of [2], they create a by-product of camera translation, which means every sample is not on the object-centric condition. Check for the optimization step (2-5 columns) of the first row to find the gradual change of its head position to the upper-left direction. The second row, however, cannot find its optimal direction because of the strict condition of the translation vector. Formally speaking, the lack of translation optimization makes the camera see always the center of the implicit space, thus limiting the optimizable path of the camera pose.

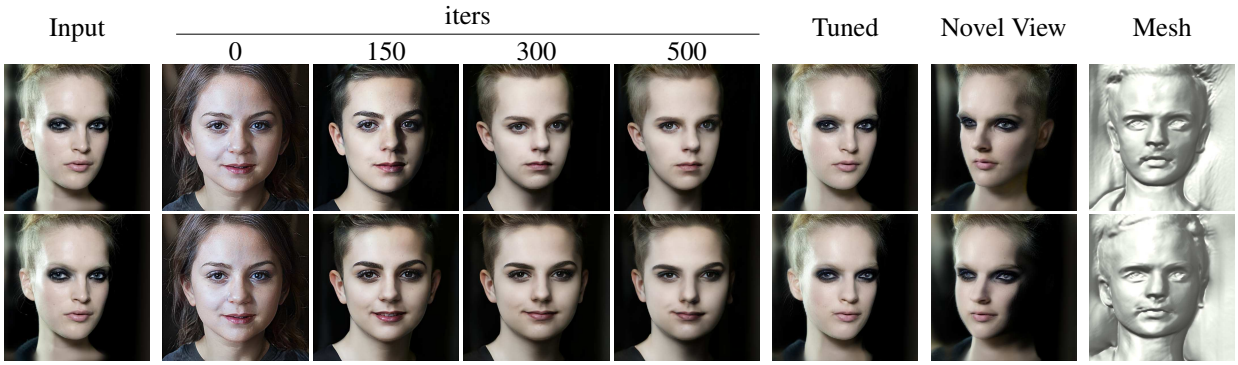


Figure 4: **Importance of additional learnable translation parameter.** We compare our model(top row) with the model that doesn't have additional learnable translation parameters(bottom row). The top row shows both gradual changes in head location on the image plane and camera direction, while the other cannot converge to suitable camera rotation.

B.3. Comparison with 2D GANs

We also provide additional visual comparisons of inverting 3D GANs as opposed to inverting 2D GANs. As demonstrated in Fig. 5, 3D GAN inversion offers more reliable reconstruction in novel views, especially when given a facial image with a great deal of rotation. Furthermore, in Fig. 6 we perform latent-based edits while changing the camera pose, for which the latent code of the EG3D model only needs to be moved in one direction, while the latent code for StyleGAN2 needs to be moved in two directions.

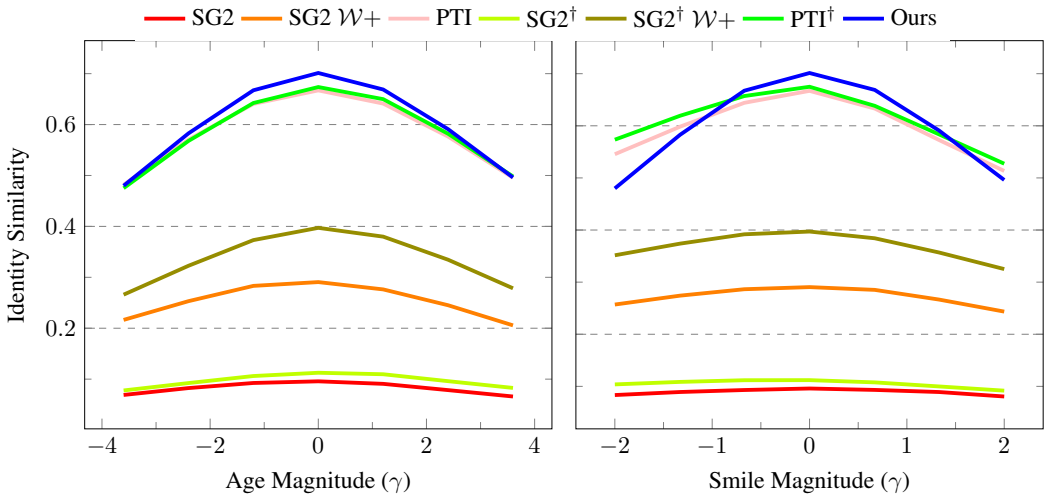
Figure 5: **Comparison of pose manipulation of real images inverted with 2D GANs and 3D GANs.** As can be seen, pose manipulation using PTI on StyleGAN2 (first row) only allows for implicit control by the editing magnitude and larger step sizes result in undesired transformations. On the other hand, using our method on EG3D (second row), allows for explicit control and because the acquired latent representation are viewpoint independent, the edits geometrically consistent.

Figure 6: **Additional results of simultaneous attribute editing and viewpoint shift comparison of 2D and 3D GANs.** We compare editing results of applying attribute editing (smile) and viewpoint interpolation at the same time on the latent code acquired by PTI [13] on StyleGAN2 [9] and the latent code acquired by our method on EG3D [1]

C. Qualitative Evaluation of Editing

Attr.	γ	SG2	SG2 $\mathcal{W}+$	PTI	SG2†	SG2† $\mathcal{W}+$	PTI†	Ours
Age	-3.6	-7.320	-6.598	-6.625	-6.522	-6.228	-6.831	-7.146
	-2.4	-5.455	-4.804	-4.526	-4.834	-4.518	-4.726	-4.922
	-1.2	-3.02	-2.557	-2.3	-2.737	-2.476	-2.518	-2.584
	0	0	0	0	0	0	0	0
	1.2	3.142	2.696	2.656	3.141	3.075	2.570	2.725
	2.4	6.464	5.74	5.506	6.686	6.414	5.574	5.929
	3.6	9.868	8.926	8.558	10.267	10.077	8.668	9.296
Smile	-2	-3.150	-2.392	-2.715	-2.627	-2.111	-2.107	-2.711
	-1.33	-2.210	-1.597	-1.838	-1.829	-1.501	-1.447	-1.882
	-0.67	-1.129	-0.798	-0.920	-0.940	-0.764	-0.714	-0.937
	0	0	0	0	0	0	0	0
	0.66	0.964	0.817	0.850	0.831	0.711	0.608	0.919
	1.33	1.925	1.583	1.734	1.558	1.271	1.364	1.854
	2	2.762	2.226	2.524	2.189	1.861	1.962	2.694

(a) Quantitative evaluation of manipulation capability



(b) Quantitative evaluation of identity preservation

Table 1: **Quantitative evaluation of editability.** In (a), we apply varying magnitudes γ of age edit and smile edit to the latent codes acquired by each method and measure the amount of age change and difference of smile extent respectively. In (b), we also compare the identity similarity between the original image and edited images.

A well-performed semantic edit should preserve the original identity of the object while performing meaningful modifications to the desired attributes and that attributes only. From the latent code and camera pose derived by each inversion method, we first evaluate the editing capabilities by manipulating an attribute with the same magnitude and measuring the amount of variation between the original image and the edited image. While camera pose editing is a standard attribute to compare latent space manipulation, viewpoint editing cannot be used to compare the different methods as the generator in question controls geometric attributes explicitly. Instead, we chose *age* and *smile* for the attributes to compare latent space manipulation, using the trait-specific estimators, DEX VGG [14] for *age* and the face attribute classifier used in [10] for *smile*. The results are shown in Table 1a. On the other hand, an ideal manipulation of the acquired latent code should achieve not only high editing ability but also high identity preservation for the unchanged attributes. For each inversion method, we compute the identity similarity between the original and edited images for different editing magnitudes. As before, we use the CurricularFace method [6] to calculate identity similarity. The results are shown in Table 1b.

D. Additional Ablation Results

$\mathcal{E}\&\mathcal{P}$	$\mathcal{L}_{\text{warp}}$	$\mathcal{L}_{DR}$	LPIPS↓	MS-SSIM↑	ID Sim.↑	FID↓
✗	✗	✗	0.0789	0.8221	0.6671	32.7366
✓	✗	✗	0.0780	0.8259	0.6823	31.1518
✗	✓	✗	0.0783	0.8248	0.6750	31.3179
✓	✓	✗	0.0771	0.8295	0.7005	30.6272
✓	✓	✓	0.0777	0.8280	0.7013	30.1198

Table 2: **Reconstruction metric comparison of various combinations of proposed methods.** We evaluate our proposed optimization scheme by experimenting with certain combinations of proposed methods. We mark (✓) when the given method is employed and (✗) when it is not.

In Table 2 we show the importance of the various components of our approach by turning them on and off in turn. We demonstrate the effectiveness of each method and used in conjunction, can reliably reconstruct the 3D object.

E. Additional Experimental Results

Following the baseline comparison in our main paper, we provide additional inversion results in Fig. 7 and also provide random viewpoints using the acquired latent code. Fig. 8, Fig. 9 and Fig. 10 applies our inversion method on multiple facial datasets, and we demonstrate the effectiveness of 3D GAN inversion by providing the reconstructed mesh and novel views using the acquired latent code. Fig. 11 depicts our reconstruction ability on non-facial domain, namely cats in the AnimalFace10 dataset.

Finally, additional GANspace [4] editing results can be found in Fig. 12 and Fig. 13, where we use the edited latent code to generate 3D mesh and novel views and prove our method is capable of 3D shape editing.

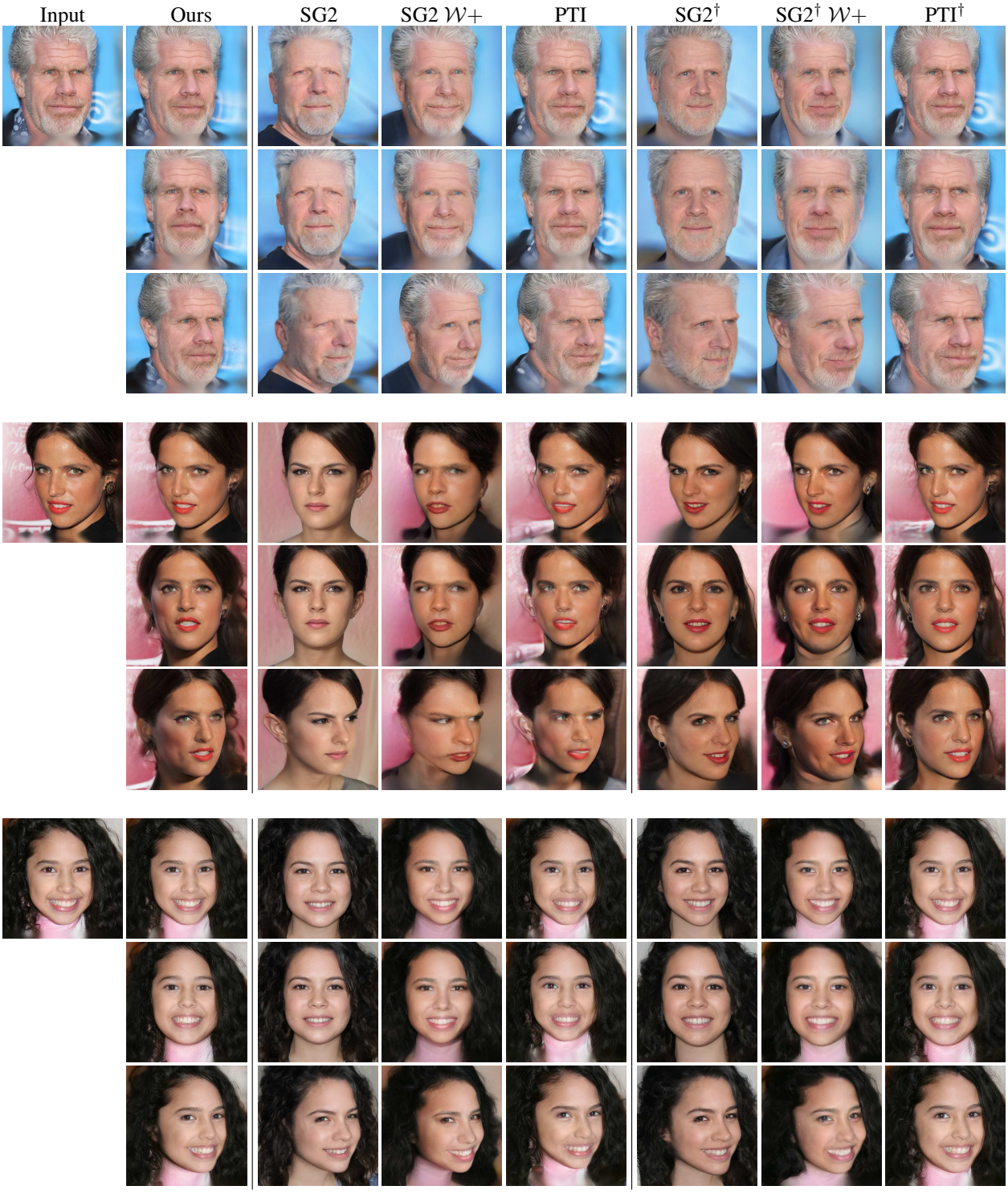


Figure 7: **Additional ablation studies on CelebA-HQ [7, 12] dataset**

Figure 8: **Additional qualitative results on in-domain dataset FFHQ [8]**

Figure 9: **Additional qualitative results on out-of-domain dataset CelebA-HQ [7, 12]**

Figure 10: **Movie Scene Novel view synthesis with 3D GAN Inversion.** We crop facial images from numerous famous movies and invert them into the latent space of EG3D [1]. We demonstrate novel view synthesis of these facial images along with visualization of 3D reconstructed mesh.

Figure 11: **Additional qualitative results on out-of-domain dataset AnimalFace10 dataset [11]**

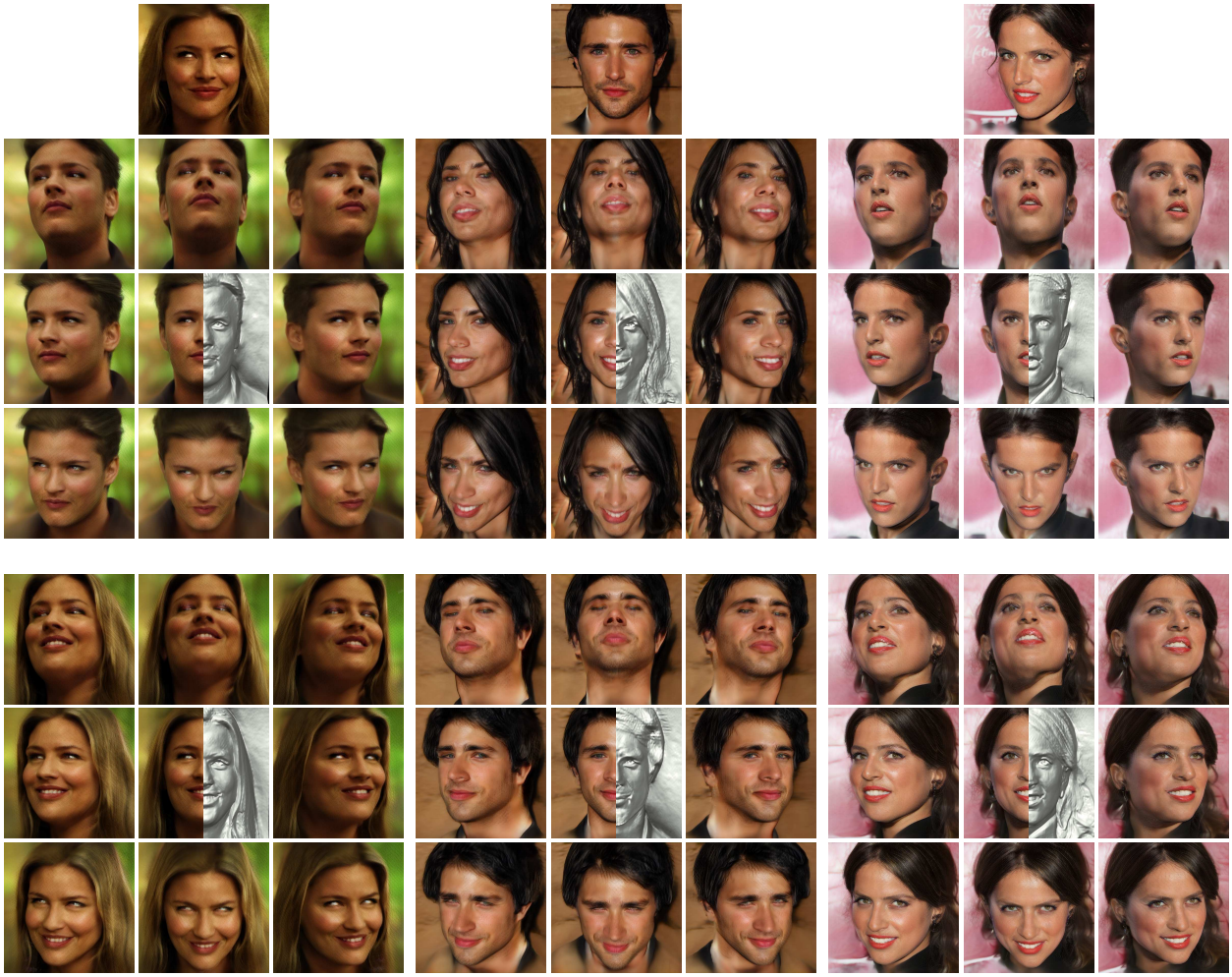


Figure 12: **3D facial Editing.** We demonstrate the style-aware geometry editing: gender(first row) and smile(second row).

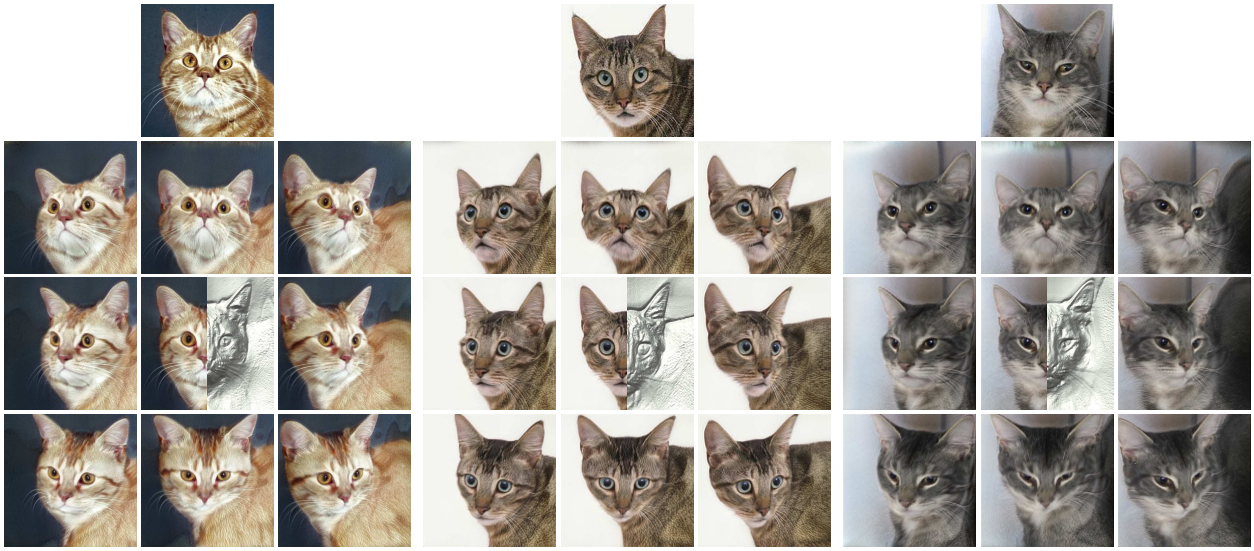


Figure 13: **Editing of cats in AnimalFace10 dataset [11] We evaluate 3D-aware edits: pupil size, and demonstrate our method enables latent-based editing for domains other than human face.**

References

[1] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022.

[2] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *CVPRW*, 2019.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014.

[4] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[6] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: Adaptive curriculum learning loss for deep face recognition. In *CVPR*, 2020.

[7] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[9] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020.

[10] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In *CVPR*, 2021.

[11] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *ICCV*, 2019.

[12] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[13] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021.

[14] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *ICCVW*, 2015.

[15] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM*, 2021.

[16] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[17] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, June 2019.