Supplementary Materials of LatentSwap3D: Semantic Edits on 3D Image GANs

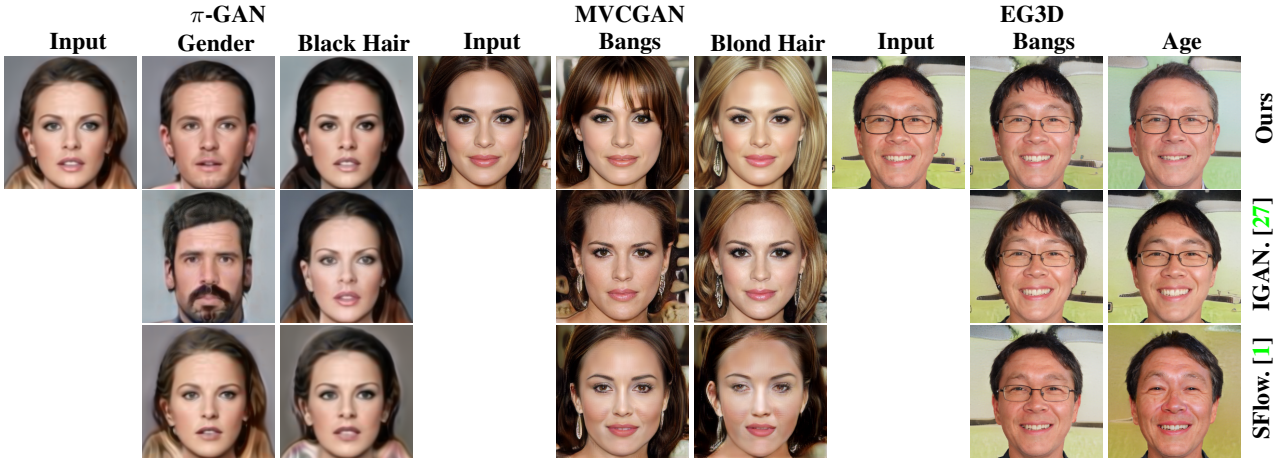

Figure 1: Detailed comparison to other methods for π-GAN, MVCGAN and EG3D. The edited attributes are annotated.

We first provide additional quantitative and qualitative results for our LatentSwap3D. Then, we show ablation studies on hyper-parameters, alternative edit techniques, and feature ranking methods. Finally, we discuss limitations, implementation details, and future work.

1. Additional Results

1.1. Additional Comparison to Other Methods

In addition to Fig. 7 in the main manuscript, we report here an additional comparison among LatentSwap3D, InterFaceGAN [27] and StyleFlow [1]. As seen in Fig. 1, our approach provides meaningful semantic edits without changing the identity on the input image and performs best for all desired attributes among 3D GANs, while the other methods may change the identity or make entangled edits.

1.2. Additional Animal Editing

To further support our findings, we included additional attribute editing examples for animals in addition to Fig. 5 in the main manuscript. Figure 2 shows the successful edits of color and breed attributes on pre-trained π-GAN, MVCGAN, and EG3D generators using our proposed LatentSwap3D.

In Fig. 3, we show additional qualitative results on applying edits to samples generated from a StyleGAN model trained on AFHQ [4] - Dogs dataset. We use the attribute classifiers presented in Sec. 4.3 to identify the dimension to edit. Note that even if the classifiers are trained on cat images, they can successfully be used for attribute editing on dog images.

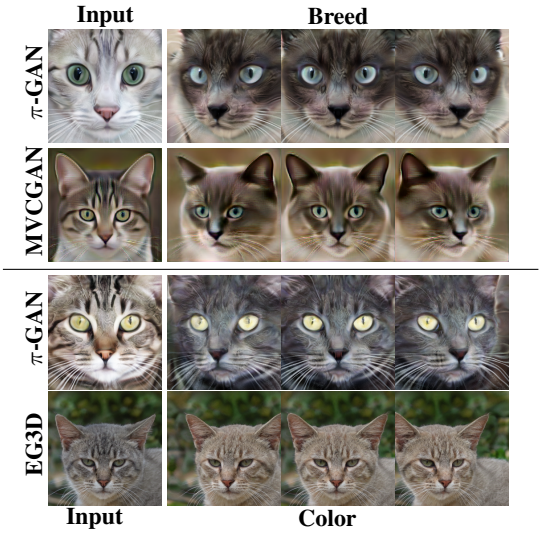

Figure 2: Results for Cats dataset [35] with π-GAN and AFHQ dataset [4] with MVCGAN and EG3D generators.

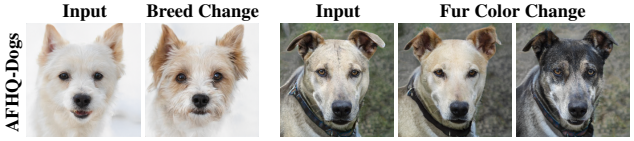

Figure 3: Results for AFHQ dataset [4], for dog images, from LatentSwap3D on StyleGAN2.

1.3. Additional Real Image Editing Results

We show additional real image editing results for MVCGAN. Thanks to the generator model's high-resolution output, our editing results are also in very high resolution. In Fig. 4, we show results for face inversion and several attribute editings, *e.g.*, smiling, changing the hair color, and wearing eyeglasses. In all cases, our edits correctly maintain the 3D consistency of the generated face.

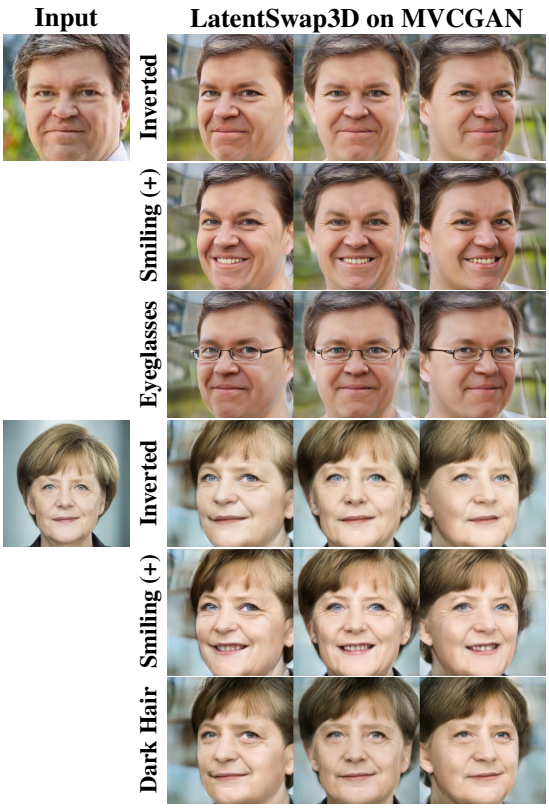

Figure 4: Additional inverted and edited examples from our approach on MVCGAN.

1.4. LatentSwap3D on Other 3D-aware Generators

GIRAFFE consists of NeRF and 2D GANs. The NeRF part outputs the features of the 3D shape and texture, while the 2D GAN part outputs the final image [20]. In Fig. 5, we show smiling and wearing eyeglasses edits from LatentSwap3D on the GIRAFFE - FFHQ model. To test how well LatentSwap3D generalizes to different datasets, we extended the experiment to include CompCars [32] using the pre-trained GIRAFFE generator. Furthermore, due to the lack of classifiers for car attributes, as a proof of concept, we trained a ResNet-50 to classify the color of a car from scratch on Myauto.ge Cars Dataset [19]. As seen from Fig. 6, our approach can successfully edit the color of the cars using these classifiers.

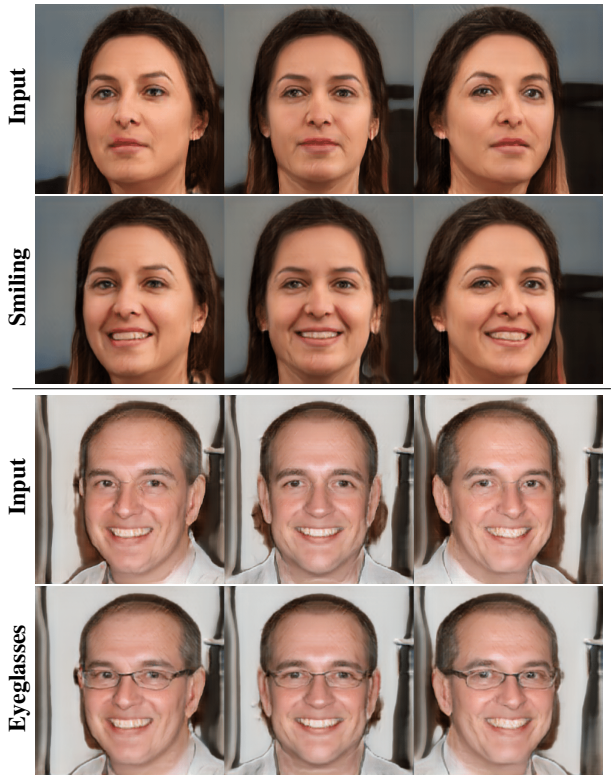

Figure 5: LatentSwap3D on GIRAFFE [20] - FFHQ

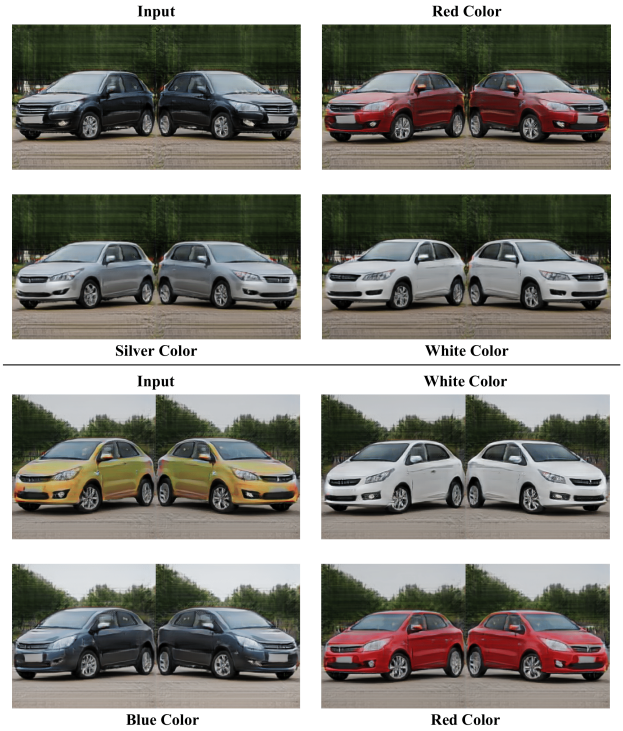

Figure 6: LatentSwap3D on GIRAFFE - CompCars [32].

	π-GAN				MVCGAN				EG3D			
	CelebA		Cats		FFHQ		AFHQ		FFHQ		AFHQ	
Method	FID	KID	FID	KID	FID	KID	FID	KID	FID	KID	FID	KID
Unedited	50.7	0.045	57.4	0.055	54.1	0.048	47.1	0.041	47.5	0.039	39.5	0.037
LCLR.	53.4	0.051	60.1	0.062	**55.6**	0.052	51.2	0.047	59.6	**0.049**	**40.1**	**0.031**
SeFa	68.2	0.062	59.2	0.059	69.4	0.063	**49.2**	0.045	64.3	0.051	44.1	0.038
IGAN.	**48.9**	**0.034**	59.6	0.059	62.3	0.056	53.1	**0.039**	**58.8**	0.053	45.8	0.039
SFlow.	52.1	0.047	59.1	0.058	56.3	**0.051**	50.8	0.041	60.5	0.050	40.4	0.034
Ours	51.2	0.048	**58.8**	**0.057**	60.8	0.053	50.3	0.041	61.1	0.051	42.1	0.035

Table 1: Quantitative comparison of FID and KID among different image editing methods for π-GAN, MVCGAN, and EG3D on attribute edits of face and animal images. The selected attributes are mentioned in Sec. 4.4.

VolumeGAN is a high-quality 3D-aware generative model explicitly trained to learn a structural and a textural representation, and it is based on NeRF [31]. The results of our approach on VolumeGAN - FFHQ are provided in Fig. 7. Our approach applies the desired attributes, *e.g.*, removing eyeglasses, changing the hair color, and reducing the facial hair, to the latent space of VolumeGAN, without changing the identity of the input face.

StyleNeRF is another high-resolution 3D-aware generative model that integrates a NeRF into a 2D style-based generator [7]. StyleNeRF is able to generate high-resolution and 3D consistent images/shapes from unstructured 2D images. Figure 8 shows our attribute editing, *e.g.*, smiling, removing bangs, and changing the hair color on StyleNeRF - FFHQ. LatentSwap3D operates successfully on the latent space of StyleNeRF by preserving the identity.

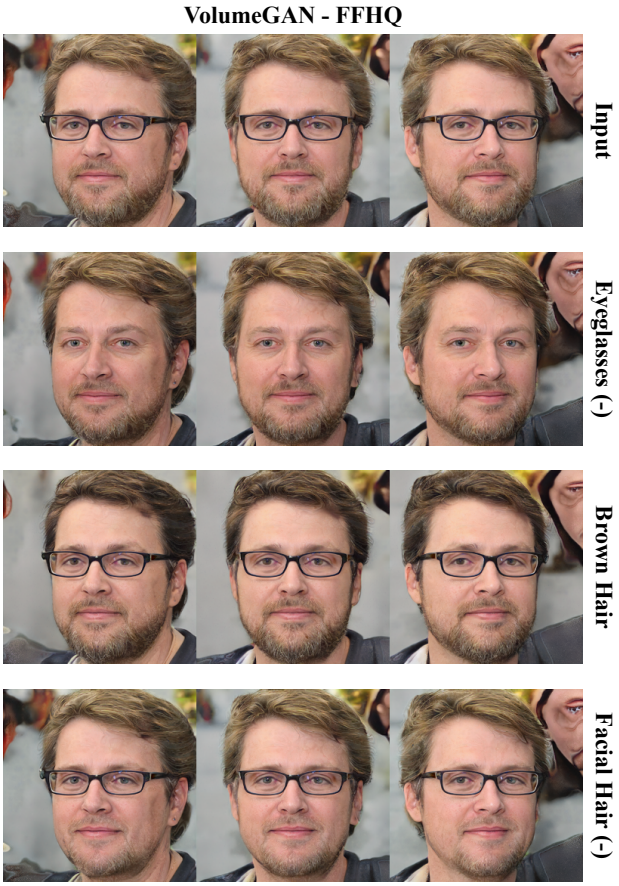

Figure 7: LatentSwap3D on VolumeGAN [31] - FFHQ.

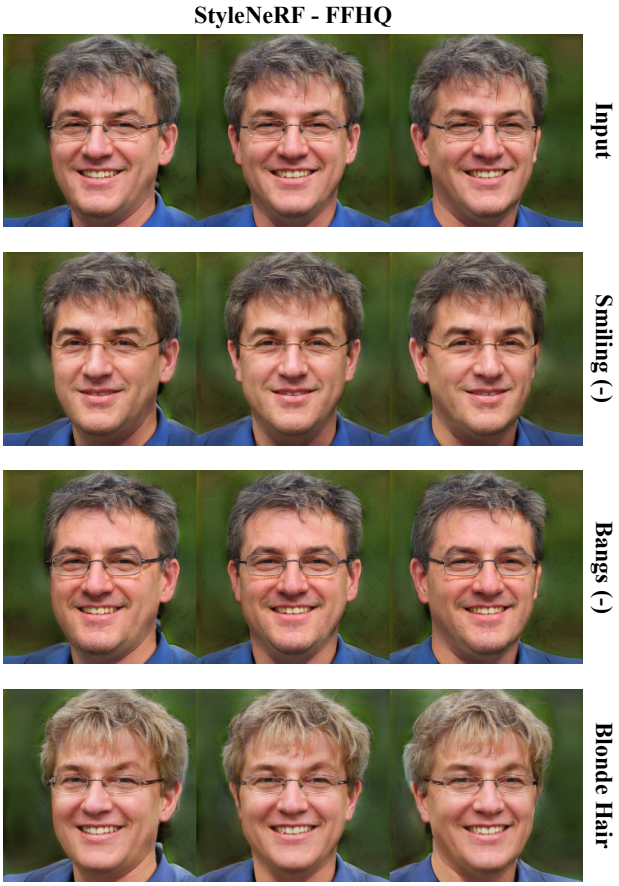

Figure 8: LatentSwap3D on StyleNeRF [7] - FFHQ.

1.5. Additional Quantitative Analysis

We report *Distribution-level Image Quality* metrics in addition to *Identity Preservation* and *Semantic Correctness* metrics in the main manuscript. We calculate Frechet Inception Distance (FID) [10, 26] and Kernel Inception Distance (KID) [2] between 10K edited images and the CelebA test dataset for the face domain and AFHQ test dataset for the animal domain. Although our method does not have the best FID and KID metrics, it is on par with other methods. Since these metrics do not indicate the methods' editing capability [16], results in Tab. 1 should be considered together with *semantic correctness* and *identity preservation* metrics in the main manuscript Sec. 4.3, Tab. 2, and Tab. 3 which instead measure the effectiveness of the edits. LatentSwap3D outperforms competitors in those metrics while providing competitive FID.

2. Ablation Studies

This section reports some ablation studies on a study on hyper-parameters of LatentSwap3D, alternative latent space edit methods: linear operations and direct optimization of the desired attribute, a comparison of the use of random forests vs. other ranking methods for latent dimensions ranking, and camera pose optimization during inversion.

2.1. Hyper-Parameters

Selection of training set size. We conduct an ablation study to assess the impact of training set size on a random forest that identifies the relevant dimensions for the given attribute. To this end, we generated additional samples from pre-trained generators at no extra cost. More samples will provide more diversity in the training set of the random forests; as expected, increasing the size of the training sets improves semantic correctness and identity preservation. However, doubling the training from 10K to 20K samples has diminishing returns (same *Semantic Correctness*), as seen in Tab. 2. For the sake of efficiency, we picked 10K for all our experiments.

Training Set Size	Sem. Cor. ↑	Ident. Pres. ↑
5K samples	92%	73%
10K samples (default)	95%	71%
20K samples	95%	70%

Table 2: Semantic correctness and Identity preservation metrics on various training set sizes on MVCGAN - FFHQ.

Selection of τ. The metrics for *Semantic Correctness* and *Identity Preservation* with respect to different values of τ are presented in Tab. 3. It is important to note that the choice of τ significantly impacts the identity preservation metric. Specifically, as τ increases, the identity preservation metric decreases. In contrast, the *Semantic Correctness* metric exhibits diminishing returns after $\tau = 25\%$. Therefore, we picked this value and kept it constant during our experiments.

τ	Sem. Cor. ↑	Ident. Pres. ↑
15%	78%	88%
25%	95%	71%
35%	96%	65%
45%	97%	58%
55%	97%	44%
65%	97%	31%

Table 3: Semantic correctness and Identity preservation metrics on different values of τ on MVCGAN - FFHQ.

Selection of support set size. We also report the semantic correctness and identity preservation metrics for different support set sizes for finding the reference image. As shown in Tab. 4, decreasing the support set size leads to improved semantic correctness since the reference image has more representative features based on the desired attribute. Conversely, increasing the support set size improves the identity loss metric. Based on the results, the optimal support set size identified is 32, which we used during all our experiments.

Support Set Size	Sem. Cor. ↑	Ident. Pres. ↑
1	95%	70%
16	95%	70%
32	95%	71%
128	94%	72%
1024	93%	72%

Table 4: Semantic correctness and Identity preservation metrics on different support set sizes on MVCGAN - FFHQ.

2.2. Alternative Edit Techniques

Linear operations. StyleGAN-based generators use AdaIN [11] layers to guide the image generation process. AdaIN layers apply a linear transformation to the input features; therefore, they are suitable to be modified with simple linear transformations in the latent space. For example, recently [30] showed that such linear operations are enough to provide disentangled and fine-grained manipulations in the latent space of StyleGAN2 [14]. While some 3D GANs employ a style space where linear operations can be applied, such as EG3D, others do not, and one of the objectives of this work was to be able to develop a method that is completely generator agnostic. For example, π-GAN and StyleSDF use SIREN [29] layers that enforce periodicity due to the presence of sin-based activation

functions in the learned latent space. Intuitively linear edits of latent codes (such as additions or subtractions) will not perform nicely in a periodic latent space, therefore motivating the need to resort to the feature swapping mechanism of LatentSwap3D. To verify this intuition, we conducted an ad-hoc experiment performing linear operations, such as addition and subtraction, on the latent spaces of π-GAN and StyleSDF. We reported the results in Fig. 9. Linear edits in this context are defined as constant changes on the top 256 features ranked from our trained Random Forests. To increase or decrease the corresponding latent codes, we look at the sign of the difference between the latent codes of the image that will be edited and the reference images that have the desired attribute. Linear operations on π-GAN can result in images with some artefacts (*e.g.*, π-GAN - smiling) or lower intensity edits (*e.g.*, π-GAN - black hair). For StyleSDF, linear operations can apply the desired edit and generate realistic images but have an undesired side effect on the identity. The StyleSDF - female edit changes the background and clothing, while the StyleSDF - age edit also changes the hairstyle. For both generators and all four attributes, LatentSwap3D can perform disentangled edits free of artifacts and preserving identity.

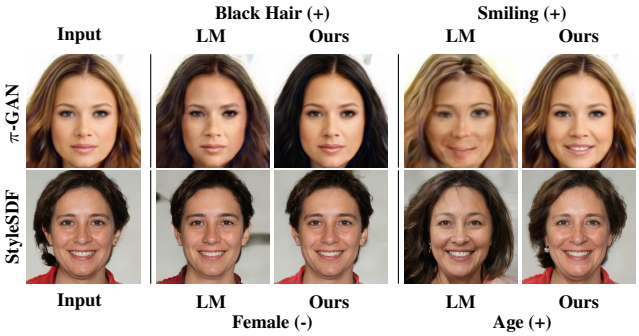

Figure 9: Effect of *linear manipulations* (LM) on the latent space of the 3D GANs, π-GAN and StyleSDF.

Direct optimization. Since we have differentiable image classifiers for the attribute we would like to edit, an alternative to LatentSwap3D would be to directly optimize latent codes to maximize the presence of the desired attribute as measured by the respective classifiers. To compare against this alternative, we first trained binary image classifiers based on ResNet-50 [9] on the CelebA [17] dataset for each attribute. Then, we take the latent codes of the original image as the initial point and try to learn an offset in the latent space that, when summed to the initial latent code, applies the desired transformation. To optimize the offset, we feed the generator the original latent code modified by the offset, generate an edited image, and provide it to the attribute classifier. At this point, we can compute as a loss function the cross-entropy loss between the output of the classifier and the class of the desired attribute and minimize it to optimize the offset using back-propagation directly. For the optimization procedure, we perform 400 iterations with Adam [15]. As shown in Fig. 10, this method struggles to preserve the identity of the edited image (see all lines). Moreover, it learns edits that are not realistic but are classifier-biased, such as the smiling (+) attribute that brightens the teeth. In contrast, the smiling (-) attribute changes the color of the teeth to the skin color; see light-green arrows in Fig. 10.

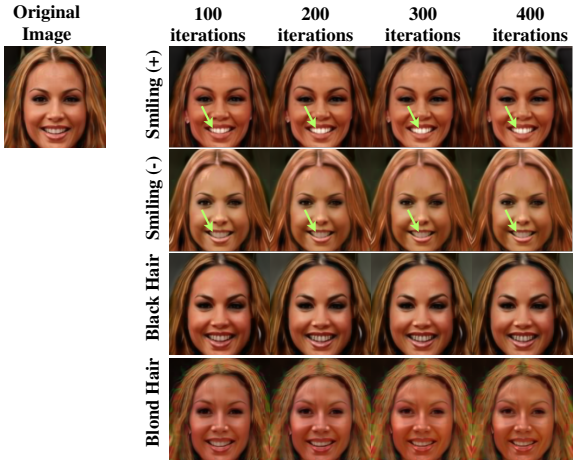

Figure 10: Directly optimize the latent codes using pre-trained image classifiers and back-propagation.

2.3. Other Feature Ranking Methods

In LatentSwap3D, we use Random Forests [3] based feature ranking; in this section, we experimentally motivate this choice by considering other methods for feature ranking. We consider three alternatives: (i) the *SelectKBest* [23] method from the popular SciKit-Learn library that sorts the feature based on a score function, such as χ^2 [22], and selects the k features with the highest scores, (ii) *Support Vector Machine* (SVM) [5] based method that takes the absolute values of the feature coefficients of a trained linear SVM, and (iii) *SHapley Additive exPlanations* (SHAP) [18] based method that explains the output of trained machine learning models by calculating the importance of the features. As can be seen in Fig. 11, SHAP- and RF-based methods show similar performance on the attributes *female (-), smiling (+)*. However, for *blondness (+) and makeup (+)* attribute edits, the random forest-based ranking provides higher quality. On the other hand, SVM-based ranking has comparable results for *smiling (+) and makeup (+)*, but for the other attributes fails to generate the corresponding edits. Finally, the *SelectKBest* method performs similarly to the SVM-based ranking method, but it has a small effect on the blondness attribute.

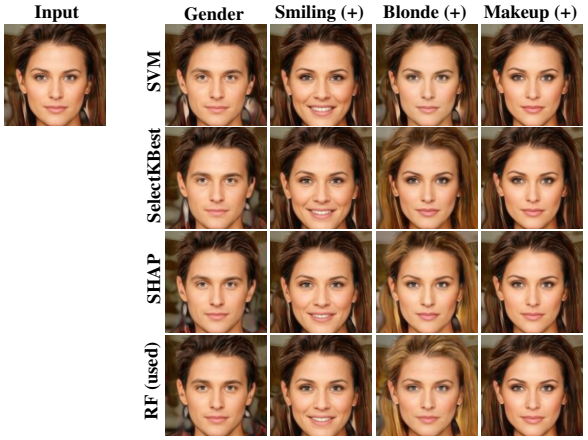

Figure 11: Comparison of various feature ranking methods on the latent space of π-GAN.

2.4. Off-the-shelf Inversion Method

Though the central objective of this paper does not revolve around proposing an inversion technique, we illustrate a use-case scenario using our proposed method for real image editing in the main paper. Figure 12 shows the combination of our work with a SoTA inversion method [33] for EG3D. Our method remains valid and applicable, combined with arbitrary GAN inversion methods (including the most recent ones).

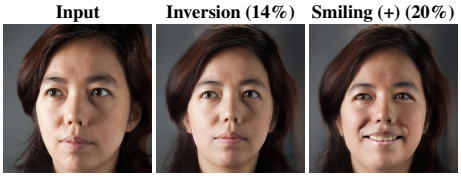

Figure 12: Percentage (%) denotes the identity change from the input image.

2.5. Consecutive and Complex Edits

We provide consecutive edits in Fig. 13. LatentSwap3D operates multiple edits, such as *blue eyes*, *smiling*, *blonde*, *gender*, and *age*.

2.6. Details on Camera Pose Optimization

Using off-the-shelf face pose estimation can be an alternative to the proposed method for the specific case of faces. However, it will hinder the generalizability of the inversion procedure to only those datasets or object categories for which a pose estimator can be trained. Our alternating optimization schema, instead, only relies on the assumption of having a trained generator and, as such, we believe, provides a more general solution. To show the impact of optimizing the pose on the inversion process, we report a comparison in Fig. 14.

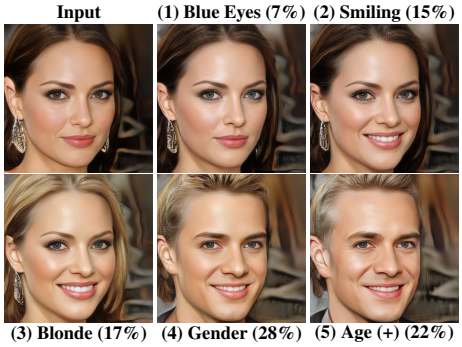

Figure 13: We report five sequential edits, where % denotes the identity change between consecutive edits.

Figure 14: Importance of Camera Optimization in Inversion Procedure. Off-the-shelf pose estimator [6].

3. Limitations

Under-represented Attributes on Training Datasets of GANs. During the development of this work, we identified some attribute manipulations that cannot be applied in the latent space of pre-trained 3D-aware image generators. These usually cover under-represented classes in the original training set, such as faces with a hat or earrings. We hypothesize that these samples fall out of distribution for the generator, so they do not have specific dimensions in the latent space allocated to them. For this reason, reproducing them with our editing technique is difficult. We show some failed edits in Fig. 15.

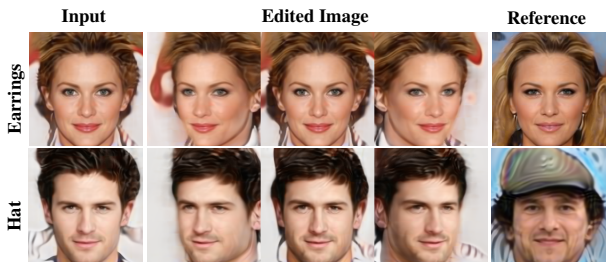

Figure 15: Failure edits on the latent space of π-GAN for classes under-represented in the training set.

Real Images Inverting Capabilities of GANs. We showed promising initial results on editing real images via GAN inversion followed by LatentSwap3D. During the development of this work, we found that the inversion of an image in the latent space of 3D generators is quite challenging and sometimes fails to generate high-quality outputs or maintain the identity of the inverted face. This is particularly true for StyleSDF, where the inverted faces resemble the original but not perfectly. We show one example inversion in Fig. 16. However, this limitation is naturally solved using newer and more powerful generators with better inversion capabilities, *e.g.*, MVCGAN. As shown in the main manuscript, our model can produce consistent attribute editing on real images with a powerful generator.

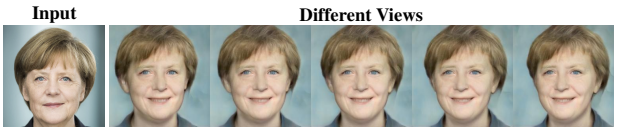

Figure 16: Inversion on the latent space of StyleSDF. While the global appearance matches, the identity is not preserved.

4. Implementation Details

4.1. Datasets

We test our proposed model, LatentSwap3D, with 3D-aware generative models on images from six different datasets: CelebA (256x256) [17], FFHQ (1024x1024) [13], Cats (256x256) [35], AFHQ (512x512) [4], and CompCars [32] (256x256) method to the 2D GAN model StyleGAN2 on FFHQ (1024x1024) [13], AFHQ (512x512) [4], and MetFaces (1024x1024) [12].

CelebA [17] is a large-scale dataset of ∼200K face images of over 10K different celebrities and 40 annotated attributes for each image. Its resolution is (256x256). π-GAN and MVCGAN provide pre-trained weights for CelebA.

FFHQ dataset consists of over 70K high-resolution (1024x1024) and high-fidelity images of human faces [13]. The dataset has diverse samples regarding age, ethnicity, and wearing accessories. It is used in this work to generate images from StyleNeRF, VolumeGAN, GIRAFFE, StyleSDF, MVCGAN, and EG3D.

Cats contains 6K images (128x128) of cat heads [35]. The dataset is used to generate cat images from π-GAN.

AFHQ is a dataset that contains over 15K high-resolution (512x512) and high-quality images of animal faces [4]. The dataset has three domains: dogs, cats, and wildlife animals, and each domain has 5K samples.

MetFaces consists of over 1K human faces (1024x1024) extracted from works of art. The images are automatically aligned and cropped [12]. This dataset is used only when evaluating LatentSwap3D on StyleGAN2.

CompCars contains ∼137K images of 1716 unique car models [32]. Its resolution is 256x256. This dataset is used only when evaluating LatentSwap3D on GIRAFFE.

4.2. Runtime Analysis

LatentSwap3D consists of two main steps, as shown in the main manuscript Fig. 2a and 2b. We measure the runtime on an NVIDIA Tesla T4 GPU Machine with 12-cores.

Method	Identifying Step	Editing Step	Image Inversion
π-GAN	180 min.	600 ms/im.	20 min./im.
MVCGAN	68 min.	600 ms/im.	8 min./im.
EG3D	72 min.	600 ms/im.	8 min./im.

Table 5: Overall runtime analysis of the proposed method. The values for the first column are calculated using a dataset of 10K generated images.

Identifying Relevant Latent Dimensions. For the step in the main manuscript Sec. 3.3, there are three main processes: **(i)** generating the training set from random sampling in the latent space of the generator, **(ii)** predicting the probabilities of the presence of the desired attribute in the generated images using pre-trained image classifiers, and **(iii)** training a random forest to predict the presence of the desired attribute from the latent codes. Considering π-GAN as a generator, the image generation step takes 2 hours for 10K images, while for MVCGAN and EG3D, it takes 8 and 12 minutes, respectively. Labeling the 10K images using the pre-trained image classifiers takes 45 seconds per attribute. Finally, the training process of the random forests takes 1 minute per attribute.

Attribute Editing on Latent Dimensions. The second step is described in the main manuscript, Sec. 3.4, which takes around 600 milliseconds per image for all generators.

3D Edits on Real Images. The runtime analysis for each generator's inversion of real images is shown in Tab. 5. The inversion procedure can be sped up using encoder-based inversion approaches. However, we leave it to future development.

4.3. Details of Animal Attribute Classifiers

We trained ResNet-50 [9] classifiers to predict *Siamese breed* and *brown color* by using the dataset [24]. Since we do not have frontal and zoom-in views of the animals, we apply a haar detector[1] for cat faces for the dataset. Our

[1] https://github.com/kipr/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml

model can successfully edit AFHQ and Cats datasets by leveraging these attribute classifiers.

4.4. Details on the Quantitative Analysis

For *Distribution-level Image Quality* and *Identity preservation* metrics, we use 2000 generated images per attribute from five different attributes, 10K in total per method. For LatentSwap3D, InterFaceGAN [27], and StyleFlow [1], we select the attributes for the three generators as follows: for π-GAN we tested *gender, smile, age, hair color, and heavy makeup*, while for MVCGAN and EG3D, we picked *gender, smile, age, glasses, and adding beard*. SeFa [28] and LatentCLR [34] are unsupervised edits discovery methods. Therefore we cannot isolate specific attribute editing transformations. So instead, we take the top five *semantics* for SeFa and five *directional models* for LatentCLR.

4.5. Details on the Comparison to Other Methods.

Since the other 3D editing methods apply to specific architectures or have their own generator part, we pick 2D attribute manipulators that have been proven to work well on 2D generators as baselines. They can also be applied to latent spaces of 3D GANs. InterFaceGAN [27] and StyleFlow [1] are the closest competitors to our method and were originally proposed for image generators. Similarly to LatentSwap3D, InterFaceGAN leverages pre-trained attribute classifiers to find the corresponding linear edit directions in the latent space of trained generators. However, as mentioned, linear edits are sub-optimal in the periodic space determined by the SIREN [29] activation functions used in π-GAN, MVCGAN, and others. On the other hand, StyleFlow uses the attribute information during the training of normalizing flows as conditions. When editing the desired attribute on a face sample, the user can give the desired attribute as a condition. In our comparison, we also consider methods for unsupervised discovery of editing directions: SeFa [28] and LatentCLR [34]. Both methods do not have assumptions about the characteristic of the generator to which they are applied. Therefore, they can be easily adapted to NeRF-based generators like π-GAN or MVCGAN.

5. Future Work

Real Images Inverting Capabilities of GANs. While a better 3D-aware GAN inversion was outside the scope of this work, we believe that in the future, some of the proposed techniques for style-based 2D generators like [25] could be adapted for the new category of 3D-aware generators and combined with LatentSwap3D to enable even more powerful edits on real images. For instance, if encoder-based inversion [25] is adapted, it will speed up the inversion process.

Improvement on Disentanglement. As an exciting direction to overcome the limitation of the improvement of disentanglement, we plan to explore a way of constraining the latent space of NeRF-based GAN models to exhibit such disentanglement properties. Similar paths have been recently proposed for style-based generators [8].

Finding Semantic Edits by Unsupervised or Self-supervised Manner. This study is one of the pioneers for conducting semantic edits in 3D-aware generative models. Therefore, future studies can adapt the 2D unsupervised and self-supervised image manipulators like [21, 28, 34], to provide unsupervised methods for finding semantic edits.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. In *TOG*, 2021.

[2] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[6] 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.

[7] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. In *ICLR*, 2022.

[8] Ligong Han, Sri Harsha Musunuri, Martin Renqiang Min, Ruijiang Gao, Yu Tian, and Dimitris Metaxas. Ae-stylegan: Improved training of style-based auto-encoders. In *WACV*, 2022.

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

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[11] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[12] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020.

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

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

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[16] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. In *NeurIPS*, 2021.

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

[18] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[19] Myauto.ge cars dataset.

[20] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021.

[21] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021.

[22] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 2011.

[24] Petfinder.my adoption prediction.

[25] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. In *ToG*, 2022.

[26] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 8 2020. Version 0.1.1.

[27] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE TPAMI*, 2020.

[28] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, 2021.

[29] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020.

[30] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 2021.

[31] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. In *CVPR*, 2022.

[32] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, 2015.

[33] Fei Yin, Yong Zhang, Xuan Wang, Tengfei Wang, Xiaoyu Li, Yuan Gong, Yanbo Fan, Xiaodong Cun, Ying Shan, Cengiz Oztireli, et al. 3d gan inversion with facial symmetry prior. In *CVPR*, 2023.

[34] Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *ICCV*, 2021.

[35] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection - how to effectively exploit shape and texture features. In *ECCV*, 2008.