LatentSwap3D: Semantic Edits on 3D Image GANs

Enis Simsar[†,1,2] Alessio Tonioni[3] Evin Pınar Örnek[2] Federico Tombari[2,3]
[1]ETH Zürich [2]Technical University of Munich [3]Google Switzerland

Abstract

3D GANs have the ability to generate latent codes for entire 3D volumes rather than only 2D images. These models offer desirable features like high-quality geometry and multi-view consistency, but, unlike their 2D counterparts, complex semantic image editing tasks for 3D GANs have only been partially explored. To address this problem, we propose LatentSwap3D, a semantic edit approach based on latent space discovery that can be used with any off-the-shelf 3D or 2D GAN model and on any dataset. LatentSwap3D relies on identifying the latent code dimensions corresponding to specific attributes by feature ranking using a random forest classifier. It then performs the edit by swapping the selected dimensions of the image being edited with the ones from an automatically selected reference image. Compared to other latent space control-based edit methods, which were mainly designed for 2D GANs, our method on 3D GANs provides remarkably consistent semantic edits in a disentangled manner and outperforms others both qualitatively and quantitatively. We show results on seven 3D GANs (π-GAN, GIRAFFE, StyleSDF, MVCGAN, EG3D, StyleNeRF, and VolumeGAN) and on five datasets (FFHQ, AFHQ, Cats, MetFaces, and CompCars).

1. Introduction

3D Generative Adversarial Networks (3D GANs) have broad applications in fields like computer graphics and augmented and virtual reality (AR/VR) thanks to their ability to synthesize photorealistic images with explicit camera pose control. 3D GANs could provide greater control over the subject to be edited by ensuring multi-view consistency when combined with semantic attribute editing. Such capabilities empower various applications ranging from realistic virtual try-on, and virtual product placement in movies or video games, to architectural design. For instance, they can enable *changing hair color*, *wearing eyeglasses*, and *smiling* in the case of face generation or *changing fur color and/or breed type* in the context of animal generation.

[†]Conducted this research as part of studies at TUM.

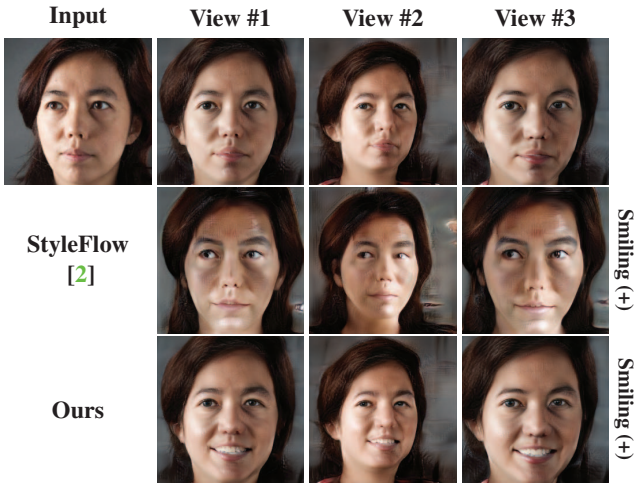


Figure 1: Our method (LatentSwap3D) inverts a given image in the latent space of a pre-trained MVCGAN [74] on FFHQ [25] while enabling novel view synthesis. Rows two and three show a comparison on attribute editing (*e.g.*, smiling) between StyleFlow [2] and ours on a real face.

Existing image editing methods have primarily focused on 2D GANs, and they provide robust control over attributes by manipulating latent spaces [21, 39, 49, 50, 70]. However, current editing methods for 3D GANs are: limited to editing pose, expression, and illumination [14, 32, 34, 57–60], require training of the generator from scratch [9, 13, 14, 22, 28, 34, 52, 55, 56, 58] or require additional semantic segmentation maps as conditioning [9, 13, 22, 55, 56]. Therefore, exploring and controlling semantic attributes on latent spaces of any *pre-trained* 3D GANs for attribute editing *without the need to re-train or fine-tune the generator* remains an open research question. Although 2D editing methods may be effective for certain 3D GANs that inherit the latent space of StyleGAN, noticeably EG3D [6], they often lead to undesirable artifacts for other 3D GANs as shown in Fig. 1. We argue that semantic attribute editing should perform as effectively on any 3D GAN model even if it does not inherit StyleGAN latent space (*e.g.*, GRAF, GIRAFFE, π-GAN, MVCGAN, StyleSDF, and VolumeGAN) [7, 36, 37, 47, 66, 74].

This work proposes a method to achieve multi-view consistent attribute editing on *any* pre-trained 3D GAN, *i.e.*, whether or not inheriting StyleGAN-based latent spaces. Our approach first explores 3D GANs' latent spaces. Then, it identifies latent dimensions that strongly correlate with the desired attribute. Finally, it performs edits by swapping the identified codes with the corresponding codes from the automatically selected reference subject already possessing the desired attribute. Unlike linear operations or predicting the edited latent codes, our proposed swapping method ensures that the edited latent codes remain within the range of valid values expected by the generative model.

Like their 2D counterparts, 3D GANs expose various latent spaces that control image generation. Therefore, as a preliminary step to enable attribute edits, we find the most suitable latent space by measuring disentanglement, completeness, and informativeness (DCI) metrics, as proposed in [15] and firstly used in [63] to assess the quality of latent spaces of generative models. To identify which dimensions in the latent space control the presence or absence of a specific attribute, we employ a method that involves training a random forest [4] with latent codes to perform regression for the presence or absence of the desired attribute. The learned random forest provides a ranking of each feature based on its influence on the output label, allowing us to determine which dimension(s) have greater control over the specific edit. Having identified the relevant dimensions, the method performs the desired transformation by swapping the top-K most essential dimensions with the corresponding dimensions from a reference image that exhibits the desired attribute. This explains why our method is dubbed *LatentSwap3D*. After showing how the number K of swapped dimensions controls the intensity of the transformation, we propose a method to automatically tune K on a per-sample basis to apply the edit without excessively altering the input image, *e.g.*, preserving the identity of the face. The project page can be found at *https://enisimsar.github.io/latentswap3d/*. Our contributions can be summarized as follows:

- We explore 3D GAN latent spaces to determine their ability to encode semantic attributes in terms of disentanglement, completeness, and informativeness (DCI).
- We propose LatentSwap3D enabling attribute editing tasks for *any pre-trained* 2D or 3D generative model *without the need to re-train or fine-tune the generators*. LatentSwap3D has state-of-the-art results in terms of *semantic correctness* by preserving identity.
- We first show results for attribute editing of generated images from random seeds of the 3D generators, then we broaden the capabilities of LatentSwap3D to edit the attributes of real images by GAN inversion.

We test our method by applying the most popular and state-of-the-art generators: π-GAN [7], MVCGAN [74], EG3D [6], StyleSDF [37], GIRAFFE [36], StyleNeRF [17], VolumeGAN [66], and StyleGAN2 [25], trained on five public datasets: FFHQ [25], CelebA [33], AFHQ [10], CompCars [68], and MetFaces [23]. The main paper focuses on the editing results for π-GAN, MVCGAN and EG3D in the FFHQ, CelebA and AFHQ datasets.

2. Related Work

Image editing in GANs. StyleGAN generators [24–26] are widely used to generate high-quality images by converting a random noise vector into a latent code that can encode semantically meaningful attributes [53, 63]. Image editing can then be implemented as manipulations of those latent codes, either supervised [2, 16, 19, 49, 51] or unsupervised [39, 50, 62, 70]. Supervised methods are based on annotated labels or pre-trained attribute classifiers to predict the presence of semantic attributes. InterFaceGAN [49] learns hyperplanes in latent space, whereas StyleFlow [2] employs conditional normalizing flows. Unsupervised approaches, instead, do not require pre-trained classifiers or labels. Semantic Factorization (SeFa) [50] finds semantic directions by retrieving eigenvectors from a projection matrix by singular value decomposition, while LatentCLR [70] uses a contrastive learning-based method to learn directions. Such editing methods are developed primarily for StyleGAN, which has special linearly editable latent spaces [63]. However, many 3D GANs [7, 36, 37, 66, 74] use a non-linear style integration unit [42], making direct 2D editing methods ineffective and causing unwanted effects such as identity change, degenerate facial attributes, and entangled edits. In this work, we propose a generalizable semantic editing method that can be used with any 3D or 2D GAN model.

3D GANs. Recent advancements in combining NeRF with GAN have led to the development of 3D GANs [7, 17, 38, 47, 65, 74] that allow explicit control over the pose of the object being generated. There are two trends for the 3D GAN architectures: (i) **one-staged:** use pure volumetric rendering in the generator and (ii) **two-staged:** use a combination of low-resolution volumetric rendering and 2D GANs to increase the output resolution. GRAF [47] and π-GAN [7] are one-stage generators that provide 3D-aware image and geometry generation using an implicit neural rendering but cannot afford high resolution while training. Two-stage generators [6, 17, 36, 37, 74], include StyleNeRF [17] and MVCGAN [74], which use NeRF-based 3D renderers, and StyleSDF [37], which employs Signed Distance Fields (SDF)-based 3D renderers as the first stage. Additionally, EG3D [6] introduces a hybrid explicit and implicit 3D representation through a tri-plane. Our work proposes an edit method that can be used with any of these models off-the-shelf without additional GAN training.

3D appearance & shape edits. Existing research on attribute editing methods for 3D shapes and appearances focuses mainly on learning an edit during the training phase. 3D face generation methods [28–30,32,34,52,58,59,71] often enable explicit control over attributes. However, some of those methods [14, 32, 34, 57–60] are limited to editing only pose, expression, and illumination, while others [28,52,71] use a set of predefined labels or losses during the training process, limiting controllability during generation. One of them, CONFIG [28], is trained on real and synthetic data with predefined attributes from scratch to enable semantic editing. Alternatively, [9, 13, 22, 55, 56] propose 3D generators that enable portrait image editing by utilizing semantic maps. However, they also require retraining of the generators from scratch. [31] showed high-quality and disentangled edits, such as *gender*, and *age*, using StyleFlow [2] on pre-trained EG3D [6]. However, we will show how StyleFlow underperforms for other attributes and on other 3D GANs. Most of the methods above have a restricted focus, as they can only manipulate the attributes of portrait images and cannot be applied to other datasets [10, 68, 73]. Furthermore, these methods are not architecture agnostic and apply attribute editing as one of the tasks optimized during the training phase [28, 52]. Our method instead enables attribute editing on any generator without requiring GAN training and on any dataset, such as human faces, animals, or cars, as we show in experiments.

Image inversion for generative models. Editing on real images is possible by obtaining the latent code for an input image by *GAN inversion*. There are different inversion approaches, from learning-based by using encoder networks [41, 45, 61] to optimization-based [1, 76] or hybrid [3, 75]. Several 3D-GAN inversion methods have recently been proposed, including optimization-based [7,67,69] and learning-based methods [5,27,44]. We also incorporate image inversion with LatentSwap3D for real image edits.

3. LatentSwap3D

3.1. Overview

We aim to build a generator-agnostic method for *any pre-trained* 3D GAN *without re-training or fine-tuning*. LatentSwap3D consists of two main components. The first one identifies essential features in the latent space of a 3D GAN that controls the desired attribute through a random forest algorithm. Then, the target attribute is applied in an identity-preserving manner through a feature-swapping approach, see Fig. 2 for an overview of the two components.

3.2. Background

Neural radiance fields (NeRFs) are represented as a set of multilayer perceptrons (MLPs), taking as input a 3D coordinate $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, and camera azimuth and elevation angles (ϕ, θ). The output is a spatially varying density and a viewpoint-dependent color. Finally, an image is rendered by sampling rays from the camera location towards the image plane and evaluating the related radiance values [35].

3D GANs are built on top of the exact volumetric rendering and aim to learn to generate NeRF-like volumes from a sampled latent noise vector by training only on unlabeled 2D images. While for 2D GANs, such as StyleGAN [26], the generation is controlled by Adaptive Instance normalization (AdaIN) [20], for a popular family of 3D-GANs, *e.g.*, π-GAN or MVCGAN, it is controlled by feature-wise linear modulation (FiLM) [42] which learns functions f and h which output $\gamma_{i,c}$ and $\beta_{i,c}$ as a function of input $\boldsymbol{x_i}$, $\gamma_{i,c} = f_c(\boldsymbol{x}_i)$ and $\beta_{i,c} = h_c(\boldsymbol{x}_i)$ where $\gamma_{i,c}$ and $\beta_{i,c}$ modulate a neural network's activations $\boldsymbol{F}_{i,c}$ of i^{th} input's c^{th} feature map, via a feature-wise affine transformation:

$$FiLM(\boldsymbol{F}_{i,c}|\gamma_{i,c}, \beta_{i,c}) = \gamma_{i,c}\boldsymbol{F}_{i,c} + \beta_{i,c}. \quad (1)$$

f and h can be arbitrary functions such as neural networks. In this family of generators, $\boldsymbol{x}_i$ is the position in space to render, and $\gamma_{i,c}$ and $\beta_{i,c}$ are obtained starting from an input latent code z and fed into a mapping network to guide image generation through SIREN [54] based FiLM layers:

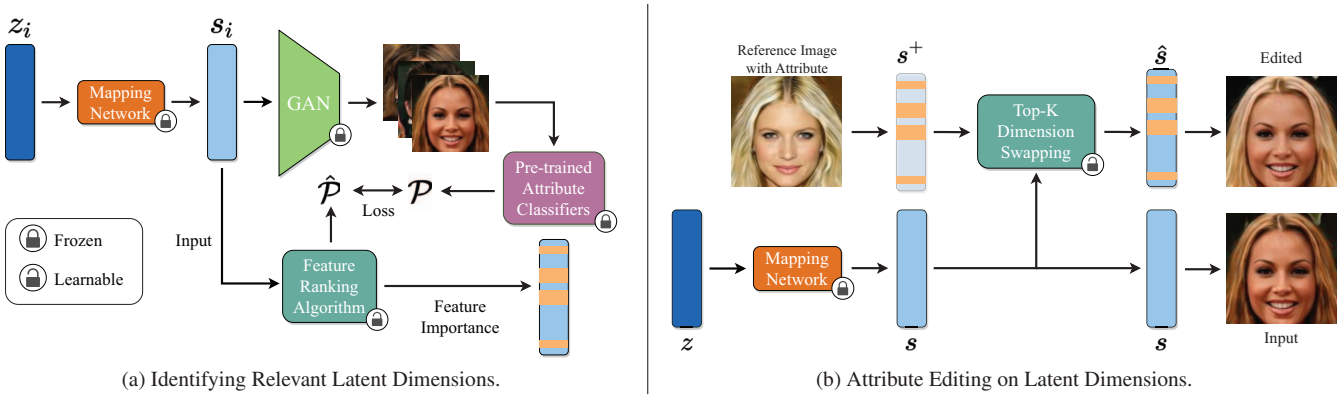
$$\phi_{i,c}(x_i) = \sin(FiLM(\boldsymbol{F}_{i,c}|\gamma_{i,c}, \beta_{i,c})). \quad (2)$$

A notable exception to this sinusoidal modulation paradigm is represented by EG3D [6], which inherits the network structure and modulation style of the StyleGAN family [26] of generative models to generate three planes of features whose inner product defines the volume used for rendering.

For both families, the underlying idea of having a mapping network re-parametrizing the conditioning vector from a random multivariate normal distribution to a *modulation/style space* is shared. However, for models like π-GAN and MVCGAN, the use of sine activation functions makes the latent space periodic and, therefore, more challenging to control compared to models based on AdaIn layers (*e.g.*, StyleGAN or EG3D). For instance, $\beta_{i,c}$ is the phase shift of a sine function and, according to Eq. 2, will give the same output for every $\beta_{i,c} + 2k * \pi$ with $k \in \mathbb{Z}$. While $\gamma_{i,c}$ controls the frequency of the sine function and affects the periodicity of the output. In practice, linear increases or decreases of $(\beta_{i,c}, \gamma_{i,c})$ might result in the opposite effect on the output of the sinusoidal activation. This causes some of the method proposals for the latent space of GANs based on AdaIN layers to fail, as shown in Sec. 4. Furthermore, all 3D GANs include several latent spaces, therefore, we need to identify the most suited one for attribute editing.

3.3. Identifying Relevant Latent Dimensions

The core idea of LatentSwap3D lies in using a feature ranking algorithm to determine the importance of features



(a) Identifying Relevant Latent Dimensions.

(b) Attribute Editing on Latent Dimensions.

Figure 2: (a) We propose to train a random forest regressor taking latent codes s_i to predict the presence/absence of a desired attribute. We use the trained forest to rank the importance of dimensions of s_i concerning the desired attribute. (b) Given the latent code s of an image, first we find the closest latent code in the support set exhibiting the desired attribute (*e.g.*, s^+ to increase *blondeness*), then we swap the top K dimensions related to the attribute to generate an edited latent code $\hat{s}$ that can be decoded in an edited image.

for a given attribute. In particular, for all experiments, we rely on a random forest [4] due to their explainability.

An overview of the process is summarized in Fig. 2a. To find relevant dimensions in the latent space of a 3D GAN, we start by generating a set of images from randomly sampled latent codes z_i and corresponding mapped codes s_i. Then, we assign an attribute score $\mathcal{P}$ for each image in the generated set using pre-trained image attribute classifiers. The scores correspond to the presence/absence of a particular attribute in the generated images. Using these scores, we train a random forest classifier to predict the presence of an attribute from the latent codes of the generator. Since random forests are very effective models for ranking feature importance, we can explicitly identify the dimensions of the latent code that correspond to desired attributes. In practice, we use the occurrence with which a forest decision node selects the input dimensions to rank the relevance of each dimension regarding the presence of a specific attribute [11].

3.4. Attribute Editing on Latent Dimensions

The existing 2D GAN editing methods perform semantic editing on latent spaces by applying algebraic operations. However, this is not applicable to 3D GANs that utilize periodic activation functions during the style integration process, such as π-GAN and MVCGAN, as discussed in Sec. 3.2, which is parameterized by a frequency and phase shift. Inspired by the style mixing method proposed in StyleGAN, we realize image editing by swapping dimensions between reference and target latent codes. While style mixing swaps entire blocks of latent codes to realize interpolation between two hand-picked latent codes, we automatically identify the target code and use the ranking identified in Sec. 3.3 to precisely swap only a small subset of dimensions. Thanks to this targeted swap, we achieve edits that do not alter the identity of the original image.

We demonstrate the attribute editing process in Fig. 2b. After determining the ranking of latent dimensions for a given attribute with the random forest, LatentSwap3D replaces the top-K features of the latent code of an image being manipulated (s) with those of an image (s^+) taken from the support set used to train the random forest and exhibiting the desired attribute. The output latent codes ($\hat{s}$) generate the edited image with the desired attribute. In particular, we pick a reference image whose attribute score is the lowest/highest for the desired attribute to remove/add the corresponding transformation to the manipulated image.

The parameter K should be carefully chosen for each transformation to preserve the identity of the generated image after attribute editing. We use the identity loss $\mathcal{L}_{ID}$ presented in Encoder4Editing [61] to automatically tune the parameter K on a per-sample basis. This loss calculates the cosine similarity between the feature embedding of the original image and that of the edited image. For example, in the face domain, we compute $\mathcal{L}_{ID}$ based on a pre-trained ArcFace [12] face recognition network, while for other domains, a ResNet-50 [18] network trained for MOCOv2 [8] is used. In particular, we select the maximum K that satisfies the constraint $\mathcal{L}_{ID} < \tau$. We provide ablation studies of the parameter K in Sec. 4.2.

To maximize identity preservation, we also choose a suitable reference image by: first selecting the top N images with the highest attribute score from the support set; then choosing the most similar to the one currently being edited according to the cosine similarity between the respective latent codes. This process ensures that features will be swapped among similar samples sharing most attributes except the one we would like to modify.

3.5. 3D Attribute Edits on Real Images

Applying LatentSwap3D to a real image requires first GAN inversion [64] to embed it in the latent space of the pre-trained GAN generator. Furthermore, the inversion of 3D GANs also requires finding the camera pose from which the real image has been acquired [44].

For 3D GAN inversion, we follow an iterative optimization approach summarized in Fig. 3, where the latent vector and pose are optimized alternatively. First, the latent vector is initialized to the mean vector in the latent space, while the camera pose is initialized to a neutral frontal position. Next, we start the inversion process by optimizing the camera location, c, while freezing the latent code, then we swap roles and tune the latent vector s, keeping the camera fixed. This process is repeated for a number of optimization steps. Then the camera is fixed, while the latent code is further optimized for a fixed number of steps.

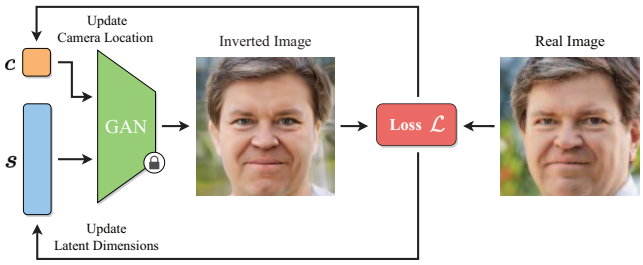


Figure 3: Inversion pipeline for a real image. s, c corresponds to the latent code and camera location, respectively.

We use a linear combination of reconstruction losses computed between the generated image and the reference one to guide the optimization: $\mathcal{L}_2$, $\mathcal{L}_{LPIPS}$ [72], and identity Loss $\mathcal{L}_{ID}$ [61]:

$$\mathcal{L} = \lambda_1 \mathcal{L}_2 + \lambda_2 \mathcal{L}_{LPIPS} + \lambda_3 \mathcal{L}_{ID} \tag{3}$$

where the values of λs are specified in Sec. 4.

4. Experiments

LatentSwap3D is tested with three state-of-the-art 3D GANs: π-GAN [7] as a representative of one-stage 3D generators with periodical latent spaces, MVCGAN [74] as a representative of two-stage 3D generators with high fidelity output, and EG3D [6] used to show how our proposal also generalizes to generators that inherit the latent spaces of StyleGAN. The images used in the experiments are from four different datasets: Flickr-Faces-HQ (FFHQ) [25], Large-scale CelebFaces Attributes (CelebA) [33] Cats [73] and Animal Faces-HQ (AFHQ) [10].

π-GAN [7] is a NeRF-based one-staged 3D GAN. A random noise vector $z \in \mathcal{Z}$ is first transformed into a 4608-dimensional vector $s \in \mathcal{S}$, corresponding to the frequency and phase shifts of FiLM layers. For π-GAN, we use this $\mathcal{S}$ space to apply our edits.

MVCGAN [74] proposes a two-stage 3D GAN. In the first stage, a neural volume renderer generates a low-resolution image and the geometry of a shape. In the second stage, a 2D styles-based generator enables high-resolution image generation. Its mapping network is converting from random noise $z \in \mathcal{Z}$ into intermediate latent codes $s \in \mathcal{S}$, conditioning the neural rendered. $\mathcal{S}$ Space has 4864 dimensions and is the space we select to apply our edits.

EG3D [6] is also a two-stage 3D GAN. Unlike MVCGAN, EG3D firstly feeds latent codes to a style-based 2D generator that predicts three orthogonal feature planes corresponding to the x, y, z axes of a 3D volume. Then, it uses a neural volumetric renderer to decode interpolated features from the three planes into a low-resolution image that later gets fed to a 2D super-resolution network. The mapping network of EG3D converts random noise $z \in \mathcal{Z}$ into intermediate latent codes $s \in \mathcal{S}$, which has 7168 dimensions and is the space we select to apply our edits.

Implementation. LatentSwap3D is investigated on $10K$ synthesized images for each dataset to train random forests. Ablation on the size set can be found in *Supplementary Material*. For the face attributes, such as *gender*, *age*, and *hair color*, we use the pre-trained attribute models of the StyleGAN [26] linear separability metrics. For both Cats and AFHQ datasets, we train a model for each attribute using annotated data from PetFinder.my Adoption Prediction Dataset [43]. For the selection of the number of features K, we set τ to 0.25 for the face domain and 0.1 for other domains. During the attribute editing step, we use a support set of 32 images among whom to pick the reference image. The ablation study on choosing τ and support set size can also be found in *Supplementary Material*. The weights in Eq. 3 are tuned to $\lambda_1 = 1.0$, $\lambda_2 = 0.6$ and $\lambda_3 = 0.3$. To rank feature importance, we use the mean decrease in impurity and the Scikit-learn [40] implementation.

4.1. Exploration of 3D GAN Latent Space

There has been limited investigation into the exploration of the latent spaces of 3D GANs. As a result, the initial phase of our research involves assessing the disentanglement, completeness, and informativeness of a latent space by utilizing the DCI metrics proposed in [15] and adapted in StyleSpace [63]. We conduct experiments to explore each generator's most suitable latent space.

The training data of the DCI regressors are generated using 40 binary classifiers trained with the CelebA attributes [33] such as *blond hair*, *gender*, and *eyeglasses* . $10K$ random noise vectors, $z \in \mathcal{Z}$, are sampled from a multivariate normal distribution and fed into the corresponding generator to get latent codes and the generated images used to train the DCI regressors. Table 1 shows how for π-GAN,

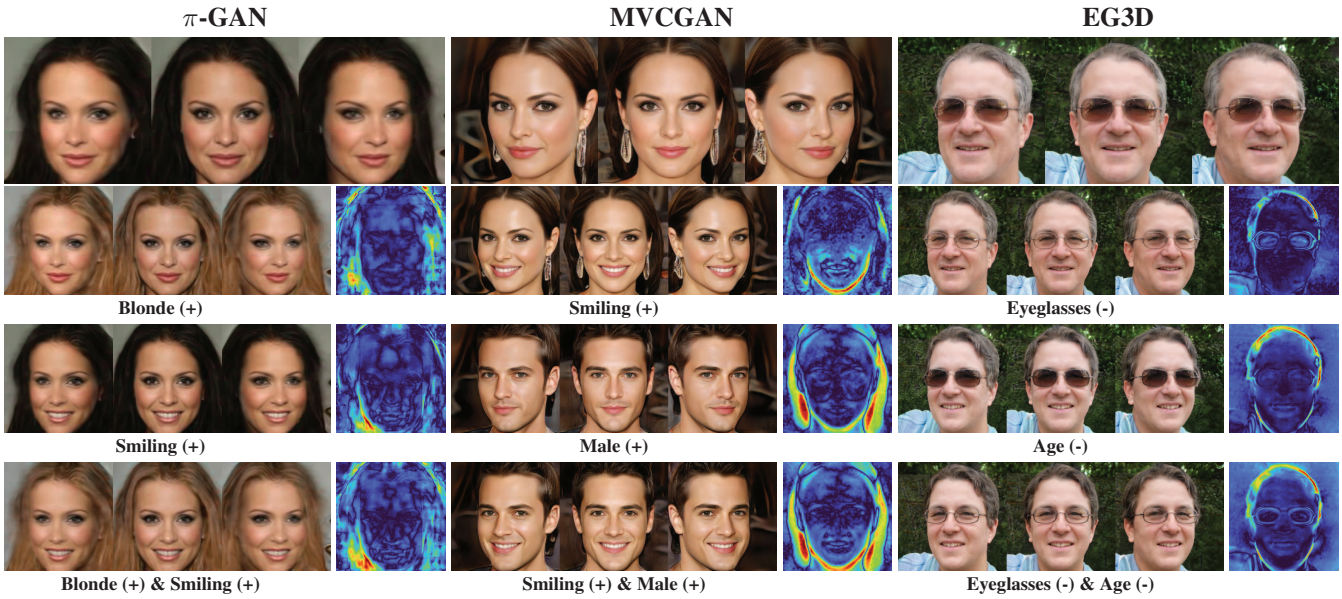


Figure 4: Qualitative results for our method, LatentSwap3D, on CelebA [33] with π-GAN [7], CelebA [33] with MVCGAN [74] and FFHQ [25] with EG3D [6]. A random seed image from different viewpoints is shown on the first row, followed by the edits for specific attributes and their combination (cumulative edits) in the last row, where (+/-) indicates an increase/decrease in the edited attribute. The rightmost column shows a heatmap of the changes in the underlying 3D geometry between the edited and original image.

Generator	Space	Disent.↑	Compl.↑	Inform.↑
π-GAN	$\mathcal{Z}$	0.44	0.31	0.73
	$\mathcal{S}$	**0.80**	**0.91**	**0.98**
MVCGAN	$\mathcal{Z}$	0.43	0.30	0.75
	$\mathcal{S}$	**0.85**	**0.91**	**0.97**
EG3D	$\mathcal{Z}$	0.57	0.33	0.65
	$\mathcal{W}$	**0.86**	**0.51**	**0.91**

Table 1: DCI metrics for the different latent spaces of π-GAN-CelebA, MVCGAN-CelebAHQ and EG3D-FFHQ. $\mathcal{Z}$ contains vectors sampled from a multivariate normal distribution. $\mathcal{S}$ and $\mathcal{W}$ represent the intermediate latent space.

the $\mathcal{S}$ space has significantly better values in terms of disentanglement, completeness, and informativeness. MVCGAN shares similar latent spaces to π-GAN, and the $\mathcal{S}$ space is better than the $\mathcal{Z}$ space. Finally, the DCI metrics for latent spaces of EG3D show that $\mathcal{W}$ space is better than the initial latent space $\mathcal{Z}$. This indicates that intermediate spaces of these models better disentangle the attributes of the generated images.

4.2. Qualitative Evaluation

Edits on generated images. Figure 4 illustrates qualitative edits on the CelebA dataset for π-GAN and MVCGAN, and FFHQ for EG3D, where we apply manipulations on attributes such as *blondness*, *smiling*, *changing gender*, *eyeglasses type*, and *age*. For these experiments, we sample a seed image from a frontal viewpoint, extract the latent code, and apply semantic edit in the latent space. Finally, we render the edited face from multiple views. We observe that our method indeed enables attribute edits in a disentangled fashion while maintaining 3D consistency from multiple views.

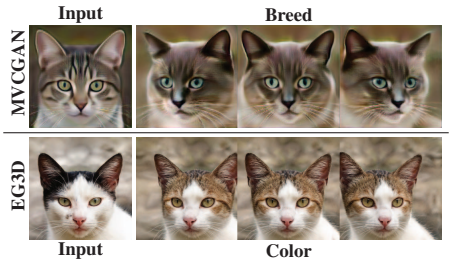


Figure 5: Results for AFHQ dataset [10] with MVCGAN and EG3D generators.

We furthermore visualize 3D difference maps to evaluate the 3D consistency by extracting the depth maps from the underlying 3D geometry between the edited face and the original one and calculating the absolute depth differences. The rightmost column of Fig. 4 shows difference maps in the form of heat maps, and the red color indicates distinct changes. Especially in the case of MVCGAN, visual edits correspond nicely to actual edits in the underlying 3D geometry (*e.g.*, the smiling edit modifies the chin and lips). We further observe that the semantic edit quality is naturally bounded by the 3D generator's quality (*e.g.*, difference maps from π-GAN blonde and smiling are noisier).

In addition to disentangled attribute editing experiments on human faces, Fig. 5 shows the results of MVCGAN, and EG3D on the AFHQ dataset to prove the applicability of our method. Our method can successfully modify the breed and fur color.

Edits on real images. As explained in Sec. 3.5, our method operates on real images captured from any viewpoint, then successfully performs editing tasks on them. Figure 6 shows semantic edits, *i.e.*, smiling and wearing eyeglasses, on the sample inverted in the latent space of MVCGAN.

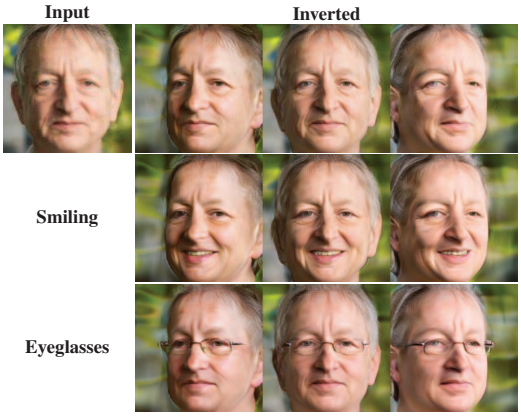


Figure 6: Inversion, editing, and novel view synthesis for real images using our method and MVCGAN as a generator.

Comparison of CONFIG and LatentSwap3D. We compare our proposed method, LatentSwap3D, to CONFIG [28], which is a neural face image generator developed to enable semantic edits. CONFIG has been explicitly trained to manipulate certain attributes and it requires a high amount of synthetic data, while LatentSwap3D finds the latent codes that enable the semantically meaningful edits on images without re-training the generator part.

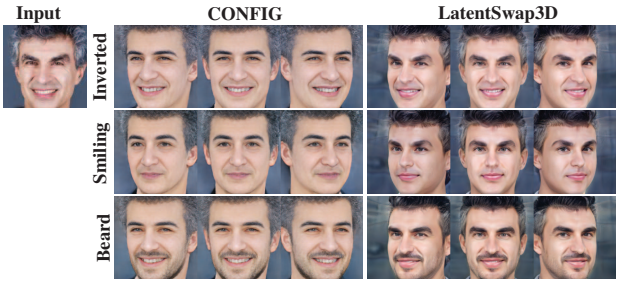


Figure 7: Comparison of CONFIG [28] and our method on smiling and beard attributes for a real face image.

Figure 7 shows real images attribute editing of the two methods for smiling (+) and beard. First, we observe that the inversion quality of real images in LatentSwap3D is better than the CONFIG method. Moreover, as the realism of semantic edits is tightly coupled to inversion quality, our LatentSwap3D generates more realistic edited images that preserve the identity of the subject. This experiment clearly shows the advantage of having a generator-agnostic method like LatentSwap3D that can easily harvest the latest advances on 3D consistent image generation over methods like CONFIG, which are bounded to a specific architecture and training regime.

2D editing methods on 3D GANs. We compare our method with the state-of-the-art 2D-based latent space manipulators, namely, InterFaceGAN [49], SeFa [70], LatentCLR [50], and StyleFlow [2]. In Fig. 8, we show a smile edit on π-GAN, MVCGAN, and EG3D. For SeFa and LatentCLR, identified directions that roughly correspond to the desired edits have been manually selected. Our method provides impressive results for π-GAN, MVCGAN, and EG3D generators, whereas the other methods sometimes result in nonsensical images or entangled edits. Since InterFaceGAN applies linear operations, if the coefficients of the manipulation are too large, they might conflict with the periodicity of latent space and generate unnatural images (such as the face edit on π-GAN). StyleFlow changes the identity and fails to apply the desired attributes. On the other hand, SeFa and LatentCLR provide unsupervised edits, but there are no semantically meaningful edits, and sometimes they cannot preserve the identity.

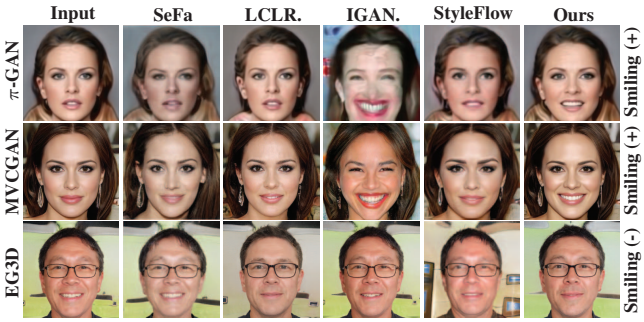


Figure 8: Comparison between InterFaceGAN [49], SeFa [50], LatentCLR [70], StyleFlow [2] and LatentSwap3D on π-GAN, MVCGAN, and EG3D models while performing a *smiling* edit.

Impact of parameter top-K. Figure 9 shows a qualitative example that emphasizes the impact of the number of dimension K swapped for two attribute edits. Higher K values increase the strength of the edit but simultaneously result in images less similar to the input. The percentage reported above/below each sample shows the value of identity loss $\mathcal{L}_{ID}$, in Fig. 9. Increasing K results in images less similar to the input image but more similar to the reference image. We automatically set K for each sample being edited such that $\mathcal{L}_{ID}$, corresponds to τ, does not go above 25% for face datasets and 10% for other domains (animals). The rightmost faces are the most similar in the support set.

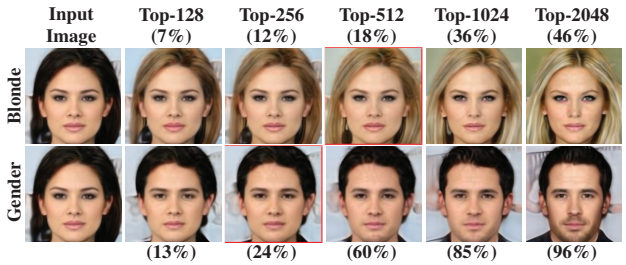


Figure 9: The % corresponds to the identity loss $\mathcal{L}_{ID}$ between edited and original as described in Sec. 3.4

Proposed method on StyleGAN2. LatentSwap3D is not limited to 3D GANs but also works *without modifications* on image-based GANs like StyleGAN2, see Fig. 10. First, by applying the procedure in Sec. 3.3, we identify the latent codes from the style space of StyleGAN2 that are most important for the desired attribute. Then, we swap those latent codes to generate the desired edits, as explained in Sec. 3.4.

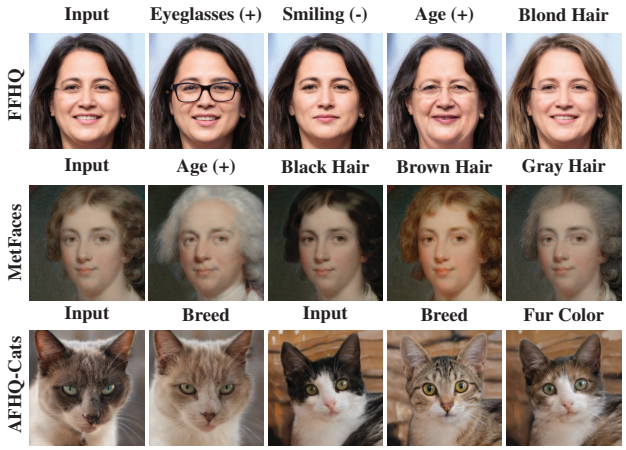


Figure 10: Several attribute edits for human and animal faces on StyleGAN2 [26] for AFHQ [10], MetFaces [23], and FFHQ [25] by using LatentSwap3D.

4.3. Quantitative Evaluation

Semantic Correctness. To evaluate the effectiveness of our attribute edits quantitatively, we use a pre-trained smile attribute classifier [48] to measure the percentage of smiling images in a set of 2500 images of not smiling faces. We edit the images with different methods and measure the increase in percentage. For the result, see Tab. 2. With π-GAN, our method increases the percentage of smiling images by 84% whereas InterFaceGAN and StyleFlow increase only by 77% and 79%, respectively. The same improvement applies to images generated by MVCGAN and EG3D. Percentages of improvement for each of them are 92% and 84%, respectively.

	π-GAN	MVCGAN	EG3D
Unedited Images	4%	3%	9%
InterFaceGAN [49]	81%	84%	85%
StyleFlow [2]	83%	78%	88%
Ours (LatentSwap3D)	**88%**	**95%**	**93%**

Table 2: Semantic correctness metric among different image editing methods for π-GAN [7], MVCGAN [74], and EG3D [6] on smiling attribute edits of face images.

Identity preservation. We measure the identity preservation between input and edited images using an identity verification tool [48] based on FaceNet512 [46]. Table 3 shows the identity preservation metric on 10K images, and compared to the other methods, our method is the best to preserve the identity of the input image.

	π-GAN	MVCGAN	EG3D
LatentCLR [50]	54%	61%	69%
SeFa [70]	62%	64%	58%
InterFaceGAN [49]	30%	51%	71%
StyleFlow [2]	68%	65%	72%
Ours (LatentSwap3D)	**74%**	**71%**	**73%**

Table 3: Identity preservation metric among different image editing methods for π-GAN [7], MVCGAN [74], and EG3D [6] on several attribute edits of face images.

5. Conclusions

To the best of our knowledge, we propose the first generator- and dataset-agnostic semantic editing method for 3D GANs. We show this by applying our method to various generators (*e.g.*, π-GAN, GIRAFFE, StyleSDF, MVCGAN, EG3D and VolumeGAN) and datasets (*e.g.*, FFHQ, AFHQ, Cats, MetFaces, and CompCars). Additionally, our method enables complex edits and multi-view consistent rendering from a single image of a real face or an object, opening the path to multiple practical applications. The broader impact of this work includes possible use cases in compression for video conferencing or 3D manipulation for AR overlays. On the other hand, like all GAN-based image editing methods, LatentSwap3D will suffer from datasets bias and is limited by the images that can be modeled by the GAN being manipulated. However, considering the rapid progress in generative modeling and the generality of our proposed framework, we envision that our method will be equally applicable in future generations of generative models, resulting in even more impressive editing capabilities.

Acknowledgements We are grateful to Google University Relationship GCP Credit Program for the support of this work by providing computational resources.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, 2019.

[2] 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.

[3] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *ICCV*, 2019.

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

[5] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional π-gan for single image to neural radiance fields translation. In *CVPR*, 2022.

[6] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022.

[7] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021.

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[9] Yuedong Chen, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Sem2nerf: Converting single-view semantic masks to neural radiance fields. In *ECCV*, 2022.

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

[11] Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.

[12] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[13] Kangle Deng, Gengshan Yang, Deva Ramanan, and Jun-Yan Zhu. 3d-aware conditional image synthesis. In *CVPR*, 2023.

[14] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *CVPR*, 2020.

[15] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *ICLR*, 2018.

[16] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, 2019.

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

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

[19] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. In *CVPR*, 2022.

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

[21] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. *ICLR*, 2020.

[22] Kaiwen Jiang, Shu-Yu Chen, Feng-Lin Liu, Hongbo Fu, and Lin Gao. Nerffaceediting: Disentangled face editing in neural radiance fields. In *SIGGRAPH ASIA*, 2022.

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

[24] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *NeurIPS*, 2021.

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

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

[27] Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 3d gan inversion with pose optimization. In *WACV*, 2023.

[28] Marek Kowalski, Stephan J. Garbin, Virginia Estellers, Tadas Baltrušaitis, Matthew Johnson, and Jamie Shotton. Config: Controllable neural face image generation. In *ECCV*, 2020.

[29] Jeong-gi Kwak, Yuanming Li, Dongsik Yoon, Donghyeon Kim, David Han, and Hanseok Ko. Injecting 3d perception of controllable nerf-gan into stylegan for editable portrait image synthesis. In *ECCV*, 2022.

[30] Yeonkyeong Lee, Taeho Choi, Hyunsung Go, Hyunjoon Lee, Sunghyun Cho, and Junho Kim. Exp-gan: 3d-aware facial image generation with expression control. In *ACCV*, 2022.

[31] Connor Z. Lin, David B. Lindell, Eric R. Chan, and Gordon Wetzstein. 3d gan inversion for controllable portrait image animation. In *ECCV 2022 Workshop on Learning to Generate 3D Shapes and Scenes*, 2022.

[32] Yuchen Liu, Zhixin Shu, Yijun Li, Zhe Lin, Richard Zhang, and SY Kung. 3d-fm gan: Towards 3d-controllable face manipulation. In *ECCV*, 2022.

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

[34] Safa C Medin, Bernhard Egger, Anoop Cherian, Ye Wang, Joshua B Tenenbaum, Xiaoming Liu, and Tim K Marks. Most-gan: 3d morphable stylegan for disentangled face image manipulation. In *AAAI*, 2022.

[35] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

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

[37] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. In *CVPR*, 2022.

[38] Xingang Pan, Xudong Xu, Chen Change Loy, Christian Theobalt, and Bo Dai. A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis. *NeurIPS*, 2021.

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

[40] 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.

[41] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible Conditional GANs for image editing. In *NeurIPS Workshop*, 2016.

[42] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

[43] Petfinder.my adoption prediction.

[44] Pierluigi Zama Ramirez, Diego Martin Arroyo, Alessio Tonioni, and Federico Tombari. Unsupervised novel view synthesis from a single image. *arXiv preprint arXiv:2102.03285*, 2021.

[45] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*, 2021.

[46] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[47] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. In *NeurIPS*, 2020.

[48] Sefik Ilkin Serengil and Alper Ozpinar. Hyperextended lightface: A facial attribute analysis framework. In *ICEET*, 2021.

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

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

[51] Yichun Shi, Xiao Yang, Yangyue Wan, and Xiaohui Shen. Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing. In *CVPR*, 2022.

[52] Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, and Gerard Medioni. Gan-control: Explicitly controllable gans. In *ICCV*, 2021.

[53] Enis Simsar, Umut Kocasari, Ezgi Gülperi Er, and Pinar Yanardag. Fantastic style channels and where to find them: A submodular framework for discovering diverse directions in gans. In *WACV*, 2023.

[54] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *NeurIPS*, 2019.

[55] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *SIGGRAPH ASIA*, 2022.

[56] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *CVPR*, 2022.

[57] Keqiang Sun, Shangzhe Wu, Zhaoyang Huang, Ning Zhang, Quan Wang, and HongSheng Li. Controllable 3d face synthesis with conditional generative occupancy fields. *NeurIPS*, 2022.

[58] Junshu Tang, Bo Zhang, Binxin Yang, Ting Zhang, Dong Chen, Lizhuang Ma, and Fang Wen. Explicitly controllable 3d-aware portrait generation. *arXiv preprint arXiv:2209.05434*, 2022.

[59] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, 2020.

[60] Ayush Tewari, Mohamed Elgharib, Mallikarjun BR, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. In *SIGGRAPH ASIA*, 2020.

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

[62] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *ICML*, 2020.

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

[64] W. Xia, Y. Zhang, Y. Yang, J. Xue, B. Zhou, and M. Yang. Gan inversion: A survey. *IEEE TPAMI*, 2023.

[65] Xudong Xu, Xingang Pan, Dahua Lin, and Bo Dai. Generative occupancy fields for 3d surface-aware image synthesis. *NeurIPS*, 2021.

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

[67] Yiran Xu, Zhixin Shu, Cameron Smith, Jia-Bin Huang, and Seoung Wug Oh. In-n-out: Face video inversion and editing with volumetric decomposition. *arXiv preprint arXiv:2302.04871*, 2023.

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

[69] Yu Yin, Kamran Ghasedi, HsiangTao Wu, Jiaolong Yang, Xin Tong, and Yun Fu. Nerfinvertor: High fidelity nerf-gan inversion for single-shot real image animation. In *CVPR*, 2023.

[70] 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.

[71] Jichao Zhang, Aliaksandr Siarohin, Yahui Liu, Hao Tang, Nicu Sebe, and Wei Wang. Training and tuning generative neural radiance fields for attribute-conditional 3d-aware face generation. *arXiv preprint arXiv:2208.12550*, 2022.

[72] 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.

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

[74] Xuanmeng Zhang, Zhedong Zheng, Daiheng Gao, Bang Zhang, Pan Pan, and Yi Yang. Multi-view consistent generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2022.

[75] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020.

[76] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016.