Quantitative Manipulation of Custom Attributes on 3D-Aware Image Synthesis

Hoseok Do[1,2] EunKyung Yoo[1] Taehyeong Kim[3†] Chul Lee[1] Jin young Choi[2]
[1]AI Lab, CTO Division, LG Electronics, South Korea
[2]ASRI, Dept. of Electrical and Computer Engineering, Seoul National University, South Korea
[3]Dept. of Biosystems Engineering, Seoul National University, South Korea
{hoseok.do, eunkyung.ryu, clee.lee}@lge.com, {hoseok03, taehyeong.kim, jychoi}@snu.ac.kr

Abstract

*While 3D-based GAN techniques have been successfully applied to render photo-realistic 3D images with a variety of attributes while preserving view consistency, there has been little research on how to fine-control 3D images without limiting to a specific category of objects of their properties. To fill such research gap, we propose a novel image manipulation model of 3D-based GAN representations for a fine-grained control of specific custom attributes. By extending the latest 3D-based GAN models (*e.g.*, EG3D), our user-friendly quantitative manipulation model enables a fine yet normalized control of 3D manipulation of multi-attribute quantities while achieving view consistency. We validate the effectiveness of our proposed technique both qualitatively and quantitatively through various experiments.*

1. Introduction

Recent advances in neural rendering [23,34,38] are making it easy to reproduce virtual 3D objects from real-world objects. Neural rendering approach is not fully scalable in practice since it heavily relies on input images and thereby can not fully represent every possible form, style, and state variation of all real and unreal objects. 3D generative adversarial networks (GANs) models, on the other hand, are more generalizable and extensible since these can not only reproduce 3D objects at scale but also allow easier configurations based on the user's intention [5,32,37], making them more suitable for various 3D image synthesis tasks.

Image manipulation on the latent space of 2D GAN has been extensively studied in recent years [2, 28, 29, 33, 39]. StyleGAN2 [15] has been the dominant technique used due to its flexibility to represent different styles and disentangled latent spaces. More recently, 3D-based GAN [24, 26, 32] for multi-view image synthesis using neural rendering [23, 25] has gained popularity. For instance, those models [5, 10, 27] equipped with StyleGAN2 modules can generate photo-realistic 3D images with a variety of attributes while preserving view consistency. Nevertheless, the existing works do not well explore a fine-grained manipulation of custom attributes (*e.g.*, capturing tiredness in a face, consisting of multiple and complex facial expressions, as shown in Figure 1) of 3D objects that are synthesized using 3D-based GAN models, and therefore it deserves more thorough research. While there have been some attempts [29,33,39] to use the latent spaces generated by GAN models to manipulate generated and real images, these approaches mainly focus on 2D objects and they are not user-friendly because users need to individually determine the appropriate manipulation scale for every use according to every specific intention.

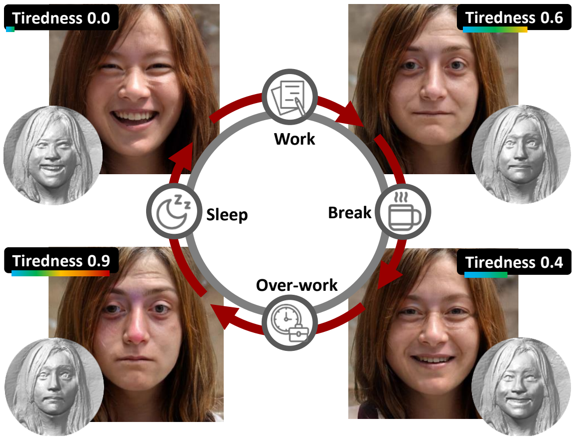

Figure 1. An example of quantitative image manipulation for the face *tiredness* attribute. Attributes expressed as complex facial features, such as tiredness, are not easy to define explicitly. Our method assigns user-defined attributes based on a small number of image samples, allowing quantitative manipulation of 3D objects according to the user's desired state changes.

Achieving view consistency during 3D image manipulation in the latent spaces is crucial to achieve the quanti-

[†] Work done during at LG Electronics

tative manipulation of custom attributes. Previously, each attribute in a multi-view image for an object was inconsistently estimated across viewpoints [8, 16, 17]. We alleviate such multi-view inconsistency problem by treating each attribute in each multi-view image as the same. That is, our 3D manipulation model is based on a 3D-based GAN model like EG3D [5], also equipped with two operators: 1) *attribute quantifier* that estimates the quantity of attribute to be edited, and 2) *navigator* that explores across the latent space to generate a manipulated image. Since the attribute quantifier guides the navigator, the manipulation quality of the navigator depends on the performance of quantifier. As quantifier, an off-the-shelf pre-trained regression model [1, 22] for a specific attribute is often used. It is not always easy to construct the pre-trained quantifier, especially for uncommon custom attributes. Hence, to better deal with custom attributes, our navigator manipulates the image by only assigning the target quantity without exploring the direction and scale of changes of the latent features. The attribute quantifier is first trained on a small number of custom image samples and then evaluates a user-defined attribute as a normalized quantity in the range $[0, 1]$. Using the quantifier, the navigator is then trained to generate and manipulate images corresponding to target custom attributes. We evaluate our approach in various attributes of 3D and 2D objects, including human faces, confirming that our method is qualitatively and quantitatively effective.

2. Related Works

2.1. Latent Space Image Manipulation

StyleGAN2 [15] can generate realistic images with various styles by learning from the styles of dataset images, and effective manipulation is possible because various attributes are disentangled. Some researchers [11, 29, 33, 39] have tried to find the direction for image manipulation in the latent space. GANSpace [11] finds meaningful directions in an unsupervised manner through principal component analysis. StyleCLIP [29] uses contrastive language–image pretraining (CLIP) [30] for text-driven image manipulation. In these methods of finding direction, the moving distance along the direction should be determined heuristically after finding the direction vector for manipulation. To mitigate this issue, StyleFlow [2] proposes a conditional exploration method of GAN's latent space using conditional normalizing flows. Because StyleFlow is based on an open-source algorithm of semantic attributes, it is not easy to extend new attributes. In contrast, our method can accurately manipulate images within the normalized range of attribute variation with a relatively small amount of labeled data.

2.2. 3D Object Manipulation

There have been several studies using GAN to generate 3D-aware images [5, 10, 27, 35]. The 3D GANs introduce new architectures by combining NeRF [23] with GAN. The vast majority of studies about 3D GAN have utilized StyleGAN2, and these models have a structure that is easy to manipulate, like 2D StyleGAN2. However, attribute manipulation based on these models has not yet been actively studied. Some of the manipulation studies that used 3D GAN were limited to the face and used a 3D morphable model (3DMM) [3, 36]-based method [19, 21, 44, 45]. Because we wanted to ensure that our methodology can be used in a domain-agnostic manner, we did not use a domain-specific pre-trained model (*e.g.*, 3DMM).

There have benn some NeRF editing studies [13, 37, 41, 41]. CoNeRF [13] provides fine-grained control over 3D neural representations from a single video. Inspired by the CoNerf approach, we extend a fine-grained control scheme to the 3D GAN model. Compared to CoNeRF, which combines parts of a video to generate a novel image, our method is based on generative models and can therefore manipulate diverse novel attributes.

2.3. Efficient Geometry-Aware 3D GANs (EG3D)

EG3D [5] introduced a 3D GAN framework for 3D-aware image synthesis. By decoupling feature generator and neural renderer, EG3D leverages StyleGAN2 for efficiency and expressiveness. Using a StyleGAN2 backbone, EG3D transforms the random latent feature z into intermediate latent feature w, then generates a feature map. The feature map forms a tri-plane representation, and then the output image is generated using a neural renderer, followed by a super-resolution module, which uses a StyleGAN2 backbone and latent feature w. Because EG3D is designed with a StyleGAN2 backbone, it allows for various manipulations in the latent space. Researchers have studied various approaches for attribute manipulation in the latent space of StyleGAN2, but have not considered 3D structure. To mitigate this issue, we propose a novel approach to 3D image manipulation considering multi-view consistency.

3. Method

We design a custom attribute quantifier to estimate the normalized quantity of an image to provide user-friendly attribute handling and a navigator to manipulate a 3D object's custom attribute for 3D-aware image synthesis. The overall scheme of our method is depicted in Figure 2. On the latent space of EG3D, a source latent feature w_s, which is obtained from a source image I_s, is manipulated to target latent feature w_t. Then, from w_t, EG3D generates the viewpoint-wise target image I_t^n whose quantity vector is estimated by $\hat{Q}_t^n$. For multi-view consistency, the quantifier is trained so that every $\hat{Q}_t^n$ follows the target quantity vector Q_t and the variance of $\hat{Q}_t^n$ is minimized. By using multiple trained quantifiers, the multi-attribute navigator takes on the role of moving the source latent feature w_s to the target w_t

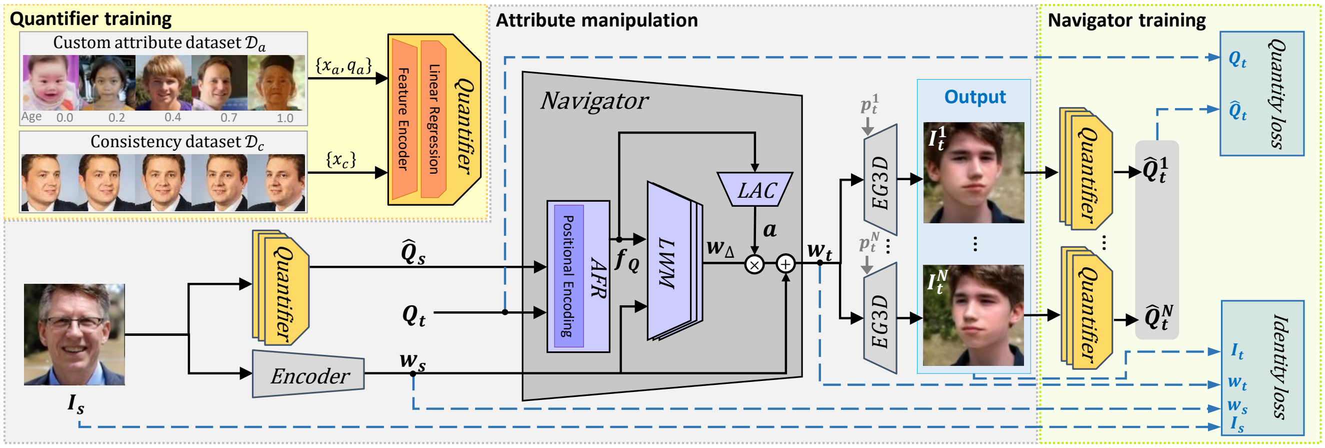

Figure 2. Overall scheme of the proposed method. The quantifiers that estimate the attribute quantities of the source and target images are first trained using the custom dataset. The navigator then manipulates the latent feature for efficient geometry-aware 3D (EG3D) [5] to generate the target image from a source latent feature given along with a target attribute quantity. The navigator is trained so that the generated 3D-aware images are close to the target attribute quantity, yet do not lose their identity.

in the latent space, referring to Q_t and the estimated source quantity vector $\hat{Q}_s$. For training the navigator, the estimated attribute quantity vector $\hat{Q}_t$ is defined by the average of $\hat{Q}_t^n$ over n, which is used to form a loss (*i.e.*, the squared error between Q_t and $\hat{Q}_t$ for accurate manipulation), while preserving the identity of I_s.

3.1. Custom Attribute Quantifier

The quantifier, which consists of a feature encoder and a linear regression model, aims to estimate the quantity q of the custom attribute from the image x. We use the pre-trained image encoder of CLIP [30] as the feature encoder, and the linear regression model comprises a fully connected layer. The quantifier is trained using two datasets simultaneously: the *attribute dataset* $\mathcal{D}_a$, and the *consistency dataset* $\mathcal{D}_c$. The former is used to train user-defined attributes, and the latter is used to train multi-view consistency.

Attribute Dataset Custom attributes are specified with a small amount of labeled dataset $\mathcal{D}_a$ provided by the user. To train the quantifier for an attribute such as age, we construct an attribute dataset $\mathcal{D}_a = \{x_a, q_a\}$ with fine-grained quantity labels in a range of $[0, 1]$ to express various attributes on a normalized scale. To assign quantity labels to the images in $\mathcal{D}_a$, we first equally divide the quantity range of $[0, 1]$ into M groups and divide the dataset $\mathcal{D}_a$ into M groups in the order of attribute change. Each group comprises Ω_a images. Then, the quantity label q_a for each image in the m-th group G_m in $\mathcal{D}_a$ is assigned to the value of $\frac{m-1}{M-1}$. To mitigate the lack of samples in $\mathcal{D}_a$, synthetic samples are augmented using Cutmix [42] between images of adjacent groups. For the image augmented by combining an image in G_m with that in G_{m+1} using Cutmix of the combination ratio γ, the quantity label q_a is assigned to $\frac{m-\gamma}{M-1}$. Since γ is sampled from the uniform distribution $[0, 1]$, the dense quantity labels are sufficient to train the quantifier for regression, yielding any quantity in $[0, 1]$. Furthermore, we also augment data samples by random color jittering. Color jittering contributes to reducing the bias that is a potential concern in visual recognition.

Consistency Dataset We construct the consistency dataset $\mathcal{D}_c = \{x_c\}$ using EG3D to train the quantifier that estimates a quantity consistently regardless of the viewpoint. The consistency dataset $\mathcal{D}_c$ is constructed using EG3D without the additional annotation cost. We utilize EG3D to generate multi-view image samples for a large number of objects. From one random latent feature w, we generate V images $\{x_c^1, x_c^2, \cdots x_c^V\}$ using different V camera viewpoints. Finally, the consistency dataset $\mathcal{D}_c$ is constructed using Ω_c random latent features, so it consists of multi-view images of Ω_c different objects.

Loss and Training We design a loss function for training the quantifier that can estimate user-defined attributes while maintaining multi-view consistency. The first term of the loss is the mean squared error (MSE) between the estimated $\hat{q}_a$ and label q_a for an input sample x_a in $\mathcal{D}_a$. The second term is the variance for the estimated $\hat{q}_c$ vector for V multi-view images for a single object in the consistency dataset, where $\hat{q}_c^v$ is the estimated quantity of x_c^v in $\mathcal{D}_c$. Quantifier training loss $\mathcal{L}_q$ is given by

$$\mathcal{L}_q = \sum_{x_a \in \mathcal{D}_a} (q_a - \hat{q}_a)^2 + \lambda_c \sum_{v=1}^{V} \sum_{x_c^v \in \mathcal{D}_c} \left(\hat{q}_c^v - \bar{\hat{q}}_c^v\right)^2, \quad (1)$$

where $\bar{\hat{q}}_c^v$ is the mean of $\hat{q}_c^v$ over all v, and λ_c is a hyper-parameter for balancing among the two loss terms. For the efficiency of training, we use a mini-batch of M images sampled, one from each group in $\mathcal{D}_a$, and M objects sampled from $\mathcal{D}_c$.

For K attributes estimation, K quantifiers are trained individually on a custom attribute. For inference, each attribute quantity scalar $q^{(k)}$ from the k-th quantifier is concatenated to create K-dimensional quantity vector $Q = [q^{(1)}, q^{(2)}, \cdots, q^{(K)}]$. A quantity vector can be designed by a combination of various quantifiers, depending on an application.

3.2. Multi-Attribute Navigator

Design Using the pre-trained quantifier, the navigator controls the attribute change in the latent space, following the given target attribute quantity. To this end, the navigator manipulates the feature in the 512-dimensional latent space $\mathcal{W}+$, where the source feature w_s is changed into the target feature w_t with the guidance of $\hat{Q}_s$ and Q_t. We design the navigator shown in Figure 2 to handle multiple attributes effectively. First, the attribute feature representer (AFR) enables more precise adjustment of attribute features by expanding the quantity information with positional encoding. AFR also represents multi-attribute feature in the attribute quantity feature f_Q. Second, layer-wise attention controller (LAC) estimates which layer among the L layer-wise mappers (LWMs) to attend each attribute manipulation.

AFR converts $\hat{Q}_s$ and Q_t to a 512-dimensional attribute quantity feature f_Q. First, $\hat{Q}_s$ and Q_t are concatenated and mapped to higher dimensional space using the positional encoding proposed by NeRF [23]. Then, a fully connected layer maps the positional encoded feature to a 512-dimensional attribute quantity feature f_Q. The input vector with 1024 dimensions is generated by concatenating w_s and f_Q, and then, the input vector is fed to each LWM. From the input vector, l-th LWM generates w^l_Δ, which represents the movement on the latent space for l-th layer of EG3D. An LWM follows the architecture of StyleGAN2's mapping network with four layers.

LAC outputs the layer-wise attention vector $a = [a_1, a_2, \cdots, a_L]$ from attribute quantity feature f_Q. And the target latent feature w_t is generated as follows:

$$w^l_t = w_s + a_l w^l_\Delta, \tag{2}$$

$$w_t = [w^1_t, w^2_t, \cdots, w^L_t]. \tag{3}$$

Because each layer of StyleGAN2 is responsible for different levels of an image's style, different attributes have different layers that are appropriate for manipulation. The LAC module, which learns layer-wise attention, is more effective for custom attribute manipulation compared to prior studies [2, 29], which manually specify the appropriate layers for each attribute. LAC is a two-layer MLP with sigmoid activation in the last layer to output in the range of $[0, 1]$.

Loss and Training The navigator is trained such that $\hat{Q}_t$ follows Q_t, while the identity between I_s and I_t is preserved. To this end, we randomly generate Q_t during the training. To enable the navigator to manipulate multiple attributes simultaneously, we randomly select J attributes to train out of a total of K attributes at every training iteration. That is, training attribute set $\mathcal{T}$ is generated as follows:

$$\mathcal{T} = \{\tau_1, \tau_2, \cdots, \tau_J\} \subset \{1, 2, \cdots, K\}. \tag{4}$$

Then, Q_t is generated by replacing J elements of $\hat{Q}_s$ with J random target values (q_t). To ensure that the navigator performs well within the entire range of $[0, 1]$, we use q_t randomly sampled from the uniform distribution on $[0, 1]$ during the navigator's training. Hence, the navigator can be trained to move the source latent feature to a point in the latent space representing an intermediate target quantity.

For view consistency in manipulation, we generate multi-view target images from w_t. Using N random camera viewpoint p^n_t, N multi-view images I^n_t are generated. And target quantity vector $\hat{Q}^n_t$ is calculated from I^n_t, then $\hat{Q}_t$ is calculated as the average of $\hat{Q}^n_t$ vectors.

With the random target attribute quantity Q_t and random source latent w_s, we train the navigator by minimizing the losses, which can be grouped into three main subsets:

$$\mathcal{L}_n = \mathcal{L}_Q(Q_t, \hat{Q}_t) + \mathcal{L}_{id}(w_t, w_s, I_t, I_s) + \mathcal{L}_a(a). \tag{5}$$

The quantity loss $\mathcal{L}_Q$ guides the navigator to manipulate the image according to the target attribute quantity. J-dimensional quantity vector Q' for training attributes set $\mathcal{T}$ is generated as follows:

$$Q' = [q^{(\tau_1)}, q^{(\tau_2)}, \cdots, q^{(\tau_J)}]. \tag{6}$$

And the quantity loss consists of two MSE terms between quantity vectors as

$$\mathcal{L}_Q = \|Q'_t - \hat{Q}'_t\|^2_2 + \lambda_q \|Q_t - \hat{Q}_t\|^2_2, \tag{7}$$

where λ_q is the hyper-parameter for balancing the loss. In quantity loss, the first term is for J attributes in $\mathcal{T}$ and the second term is for all K attributes. Because random target quantities are given for J attributes out of a total of K attributes, $K - J$ attributes of Q_t are from $\hat{Q}_s$. Hence, the second term can lead to both preserving the source attribute quantity and manipulating it to the target attribute quantity.

The identity loss $\mathcal{L}_{id}$ is designed by

$$\mathcal{L}_{id} = \|w_t - w_s\|^2_2 + \frac{\lambda_L}{N} \sum_{n=1}^{N} \text{LPIPS}(I_s, I^n_t), \tag{8}$$

where λ_L is the hyper-parameter for balancing the loss, and LPIPS denotes the learned perceptual image patch similarity [43] calculated by the distance between the activation of two images.

The third term, $\mathcal{L}_n$, is a regularization loss for layer-wise attention as

$$\mathcal{L}_a = |\alpha - \bar{a}|, \tag{9}$$

where $\bar{a}$ denotes the average of a_l over l, and α is the hyper-parameter for the target average value of attention.

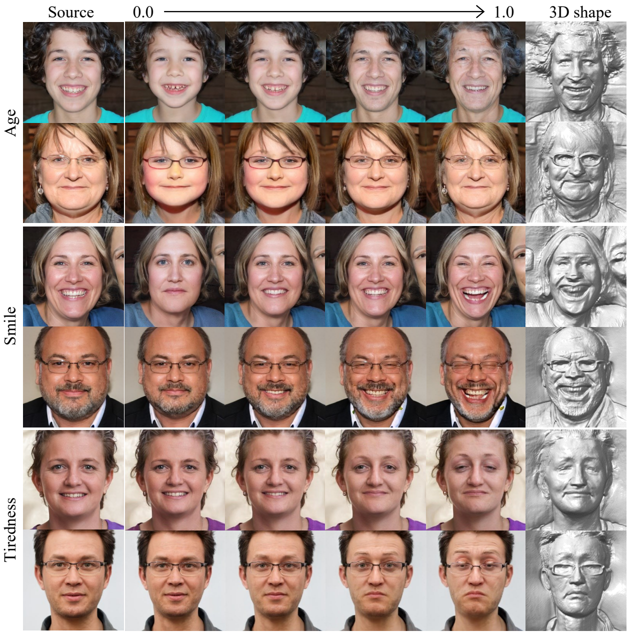

Figure 3. Quantitative manipulation results of faces for age, smile, and tiredness attributes. The image in the first column is the source, and the following four images are the manipulated images with gradually increased target quantity. The image in the last column is the 3D shape manipulated with the target quantity 1.0.

4. Experiments

4.1. Implementation Details

We evaluated our results on 3D GAN, EG3D [5]. We tested face and cat categories, each pre-trained using the FFHQ [14] and AFHQv2 [6] datasets, respectively. We experimented with the face category's age, smile, gender, eyeglasses, tiredness, pain, thinness, drunkenness, and eyebrow attributes. In the cat category, we experimented with age, eye, and fur attributes.

Custom Attribute Quantifier For the attribute dataset $\mathcal{D}_a$ of attributes, we used either a public dataset or constructed it manually from an unlabeled dataset. We used the FFHQ-Aging [28] dataset for age and gender attributes, and we used the KDEF-dyn [4] dataset and CelebAMask-HQ [20] dataset for smile and eyeglasses, respectively. For custom attributes that had no attribute dataset, we manually constructed $\mathcal{D}_a$ for each attribute. We constructed each $\mathcal{D}_a$ for attributes of cats by manually sampling from the AFHQv2 dataset. We constructed each $\mathcal{D}_a$ for tiredness, pain, thinness, drunkenness, and eyebrow attributes by webcrawling images and annotating labels. When we manually constructed $\mathcal{D}_a$, each dataset size was up to 100 samples, taking an average of 20 minutes. Details for constructing $\mathcal{D}_a$ are provided in the supplement. For the consistency dataset $\mathcal{D}_c$, we used $V = 10$, $\Omega_c = 10,000$. We trained the quantifier for 50 epochs with a batch size of M. We set λ_c to 0 until 10 epochs, then linearly increased it to 0.01 until 25 epochs, and kept it at 0.01 until the end of training.

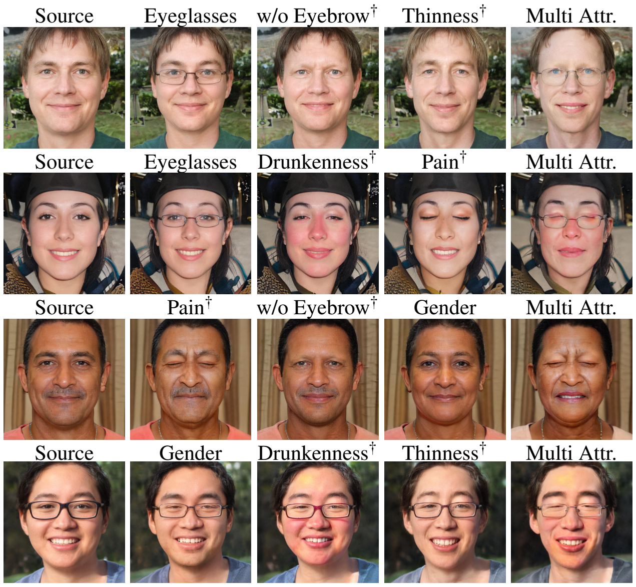

Figure 4. Various custom attribute manipulation results. The image in the first column is the source, and the following three images are the results of each manipulation for a single custom attribute. The image in the last column is the multi-attribute result of simultaneously manipulating the three prior attributes.
(† = custom attributes implemented via manually constructed D_a)

Navigator We trained the navigator with 30,000 iterations with a batch size of 16 on 4 Tesla V100 GPUs. We used EG3D to randomly generate w_s with truncation factor $\phi = 0.7$ during navigator training. We set hyper-parameters $\lambda_q = 0.5$, $\lambda_L = 0.1$, $\alpha = 0.5$, and $N = 2$.

4.2. Custom Attribute Manipulation

As shown in Figure 3, the faces were gradually manipulated for the attributes of age and smile according to the target attribute quantity q_t normalized in the range $[0, 1]$. Regardless of the source attribute quantity $\hat{q}_s$ of the source image I_s, the attribute quantity of the manipulated image changed according to the target attribute quantity. Comparing the source images in the examples of smile attributes, a woman (third row) smiles and a man (fourth row) is expressionless, which means that the source attribute quantity is different. In our results, the manipulated images of the two people's faces have similar facial expressions depending on the target attribute quantity. When q_t approaches 0, they are both expressionless; when q_t approaches 1, they both smile broadly.

Figure 4 shows that our model can manipulate faces for various custom attributes and can also manipulate multiple attributes simultaneously. We show more results of the quantitative manipulations for these attributes in the supplement. Figure 5 represents gradually manipulated images of cats with custom attributes such as fur, eye, and age. Be-

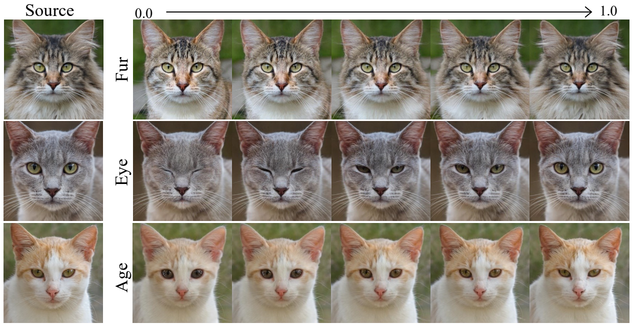

Figure 5. Quantitative manipulation results of cats for custom attributes (fur, eye, and age). The image in the first column is the source, and following five are the manipulated images.

cause we did not use any domain-specific pre-trained models, our method provides a domain-agnostic technique.

To demonstrate the expandability of our method, we apply our method to StyleGAN2, which has pre-trained models in various categories. Face and cat categories were tested, pre-trained using the Stanford Cars [18] dataset and the LSUN Church [40] dataset, respectively. As shown in Figure 6, cars are gradually manipulated for the attributes of year and type, and churches are gradually manipulated for the cloudiness attribute. The manipulated results are independent of the source attribute quantity, as were the cases for faces in Figure 3. In the case of the car year attribute, it is difficult to intuitively describe the changes in appearance according to the year attribute quantity. However, our method can visualize the changes. In detail, the newer the car is, the more emphasized the horizontal lines on the grill, the sharper the headlights, and the more curved the overall design. This result shows that our approach can be used in a similar manner as GANalyze [9], which visualizes abstract concepts such as memorability through GAN. Implementation details and more experimental results of StyleGAN2 are provided in the supplement.

4.3. Comparisons

In this section, we compare our method with other quantitative manipulation methods. For quantitative comparison, we experimented with image manipulation of the faces on age, smile, gender, and eyeglasses attributes. We compared the performance of our approach with that of StyleFlow [2], which proposes a quantitative manipulation. We also implemented StyleFlow+Q, which uses our custom quantifier, while StyleFlow uses a pre-trained model as a quantifier[*]. Details for implementing StyleFlow on EG3D are provided in the supplement.

The resultant image I^{ϕ}_{θ} was manipulated from the source image I^{ϕ} using the θ-th target value ($\phi = 1, \cdots, \Phi$, $\theta = 1, \cdots, \Theta$). In addition, we used Face++ [1] to measure r^{ϕ}_{θ}, which is the resultant attribute quantity of I^{ϕ}_{θ}.

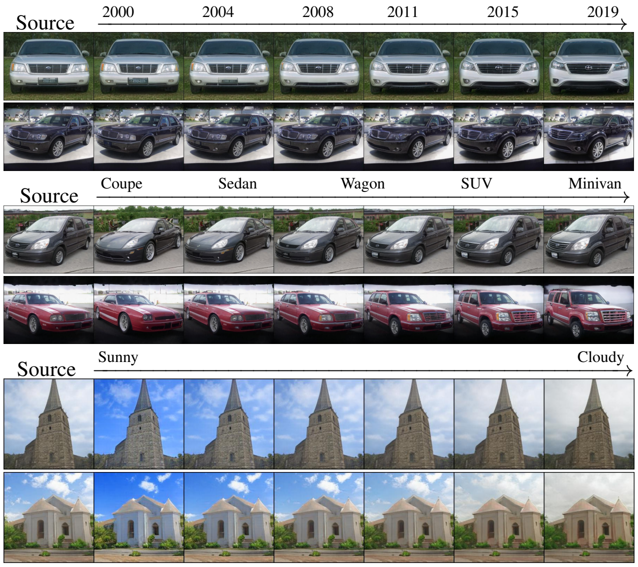

Figure 6. Quantitative manipulation results of cars for year and type attributes, and churches for cloudiness attribute in StyleGAN2. The image in the first column is the source, and the following six are the manipulated images with the target quantity increased from 0.0 to 1.0.

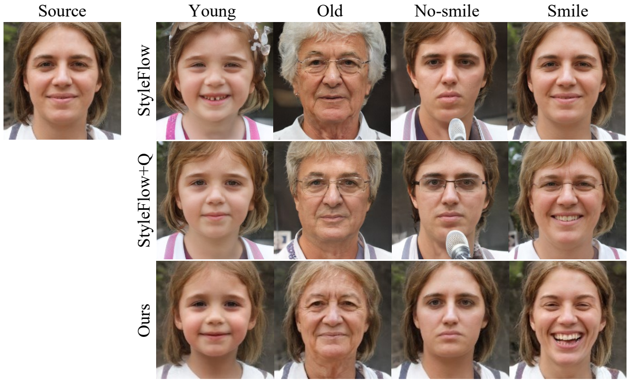

Figure 7. Qualitative comparison for age and smile attributes manipulation in EG3D. Competing methods fail to preserve identity when manipulated to *Old* and *No-smile*.

Quantitative Comparisons Image manipulation seeks to accomplish two primary objectives that are often conflict: 1) To modify an image to achieve specific intended attributes, and 2) to maintain original image's identity. To evaluate our approach to the objectives, we employed five metrics. *Manipulation exactness* and *accuracy* are both metrics employed to evaluate the success of the first objective, while *identity preservation* serves as the metric to access the achievement of the second objective. *Manipulation efficiency* measures success in both objectives, and *identity preservation* is a metric for evaluating consistency

[*] StyleFlow originally used Microsoft Face API [22] as a quantifier, but Microsoft no longer supports it. Therefore, we used Face++ [1] as a quantifier for StyleFlow implementation on EG3D.

Method	Age				Smile				Gender		Eyeglasses	
	$\sigma_m \downarrow$	$E_d \downarrow$	$\rho \uparrow$	$\sigma_v \downarrow$	$\sigma_m \downarrow$	$E_d \downarrow$	$\rho \uparrow$	$\sigma_v \downarrow$	*Acc.* $\uparrow$	$E_d \downarrow$	*Acc.* $\uparrow$	$E_d \downarrow$
StyleFlow	8.40	0.373	21.3	2.304	37.38	0.382	29.9	4.859	83%	0.586	**96%**	0.510
StyleFlow+Q	10.46	0.331	22.1	2.304	27.49	0.320	34.4	2.635	78%	0.670	79%	0.486
Ours	**8.03**	**0.245**	**32.6**	**2.235**	**21.50**	**0.210**	**51.0**	**2.255**	**95%**	**0.537**	**96%**	**0.468**

Table 1. Quantitative comparison with competing methods in EG3D. We evaluated the manipulation performance on age, smile, gender, and eyeglass attributes using five evaluation metrics: manipulation exactness (σ_m), manipulation accuracy (*Acc.*), identity preservation (E_d), manipulation efficiency (ρ), and view consistency (σ_v). Our method gives the best results.

across multiple manipulated views.

Specifically, we defined the metrics as follows: **1) Manipulation exactness** refers to the consistency of results for the same input target value. From the standard deviation $\sigma_{m,\theta}$ of the θ-th resultant quantities $\{r_\theta^1, \cdots, r_\theta^\Phi\}$, the evaluation metric σ_m is the mean of $\sigma_{m,\theta}$. Regarding **2) manipulation accuracy**, we measured the ratio of correctly manipulated images for binary-class attributes (*e.g.*, gender, eyeglasses). For **3) identity preservation**, we determined the average Euclidean distance E_d between the embeddings of two manipulated images, which were generated using two adjacent target values, as

$$E_d = \frac{1}{(\Theta - 1)\Phi} \sum_{\theta=1}^{\Theta-1} \sum_{\phi=1}^{\Phi} \| F(I_{\theta+1}^\phi) - F(I_\theta^\phi) \|, \tag{10}$$

where F is the embedding of the pre-trained face recognition model [7]. The metric of **4) manipulation efficiency** refers to trade-off ratio ρ between attribute manipulation and identity preservation as $\frac{(r_\Theta - r_1)}{(\Theta-1)E_d}$. Lastly, **5) view consistency** was calculated across multiple manipulated views. In each manipulation case, we generated three multi-view images $\{I_\theta^{\phi,1}, I_\theta^{\phi,2}, I_\theta^{\phi,3}\}$ and measured the resultant attribute quantities $\{r_\theta^{\phi,1}, r_\theta^{\phi,2}, r_\theta^{\phi,3}\}$. From the standard deviation $\sigma_{v,\theta}^\phi$ of the resultant attribute quantities, the evaluation metric σ_v is the mean of $\sigma_{v,\theta}^\phi$.

We conducted experiments with $\Phi = 100$ and $\Theta = 8$. As shown in Table 1, our approach outperforms the competing methods in all evaluation metrics. StyleFlow and StyleFlow+Q show similar performance, implying that our custom quantifier has reasonable performance compared to a public pre-trained model, Face++.

Qualitative Comparisons Age and smile attributes were tested for qualitative comparison, as shown in Figure 7. StyleFlow+Q sometimes significantly changes the source identity (*e.g.*, gender and eyeglasses status are changed in both *Old* and *No-smile* manipulation cases). StyleFlow also sometimes changes the source identity, and the manipulation performance is worse than others in the *Smile* case. Our method can manipulate images while preserving the source identity with better manipulation performance (*e.g.*, broadly smiling in the *Smile* manipulation case).

Method	Age			Smile		
	$\sigma_m \downarrow$	$E_d \downarrow$	$\rho \uparrow$	$\sigma_m \downarrow$	$E_d \downarrow$	$\rho \uparrow$
Talk-to-Edit	14.3	0.221	32.9	22.6	0.212	40.9
StyleFlow	16.9	**0.195**	28.4	29.1	**0.099**	88.9
Ours	**10.9**	0.225	**47.2**	**20.2**	0.103	**96.6**

Table 2. Quantitative comparison with competing methods in StyleGAN2. We evaluated the manipulation performance on age and smile attributes using three evaluation metrics: manipulation exactness (σ_m), identity preservation (E_d), and manipulation efficiency (ρ).

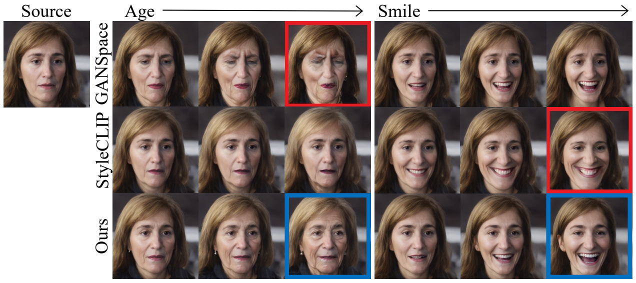

Figure 8. Comparison between models regarding the degree of image manipulation according to the manipulation strength (left: age attribute; right: smile attribute). Red boxes denote unnatural manipulated images due to excessive manipulation strength.

Comparisons in 2D We compare our method with four baselines in StyleGAN2. For quantitative comparison, we compare our method with StyleFlow [2] and Talk-to-Edit [12], which propose quantitative manipulations. For qualitative comparison, we compare our method with StyleCLIP [29] and GANSpace [11], which provide direction for attribute manipulation. The experiment was conducted by manipulating the age and smile attributes of the FFHQ test dataset using the authors' official implementations.

As shown in Table 2, our approach outperforms StyleFlow and Talk-to-Edit in manipulation exactness σ_m. In the E_d score, StyleFlow is slightly better than ours, and Talk-to-Edit is similar to ours, but our approach outperforms both methods in terms of the efficiency score ρ. This result implies that our method better preserves identity when an attribute changes by the same quantity.

Model		$\sigma_m \downarrow$	*Acc.* $\uparrow$	$E_d \downarrow$
Design	w/o LAC module	15.84	96.0%	0.415
	w/o AFR module	15.37	89.0%	0.387
Training	w/o entire attrs. in L_Q	14.78	87.5%	0.405
	w/o intermediate Q_t	15.65	**97.0%**	0.402
	Full model	**14.76**	95.5%	**0.365**

Table 3. Ablation study on quantitative manipulation. We experimented on four attributes (age, smile, gender, and eyeglasses) and reported the mean values of exactness (σ_m), accuracy, and identity preservation (E_d).

In Figure 8, we compare the degree of image change according to the gradual increase in the manipulation strength. For each manipulation, we set the manipulation strength using the values suggested in their provided code. Each method enables natural manipulation of the attribute by using the appropriate strength; however, the appropriate strength varies for each attribute and source. In GANSpace, a greater strength is suitable for the smile attribute, and a lesser strength is suitable for the age attribute, which is the opposite in StyleCLIP. Our approach can manipulate attributes in a normalized range, so ours can generate naturally manipulated images without exploring the manipulation strength.

4.4. Ablation Study

Quantitative Manipulation We conducted several ablation experiments on our method. Four ablation factors were selected from our navigator: two from the design (LAC, AFR) and two from the training scheme (entire attributes in $\mathcal{L}_Q$, intermediate target quantity). In the ablation study on the LAC module, we generated w_t by adding $w_\triangle$ to w_s. In the ablation study on the AFR module, we generated f_Q by concatenating $\hat{Q}_s$ and Q_t, then repeating it. When entire attributes are not used in $\mathcal{L}_Q$, we trained the navigator with $\lambda_q = 0$. When an intermediate target quantity is not used, we trained the navigator by randomly sampling q_t value from zero or one. We evaluated the results using the mean values of evaluation metrics in Section 4.3. As shown in Table 3, the full model exhibits the best performance in evaluation exactness and identity preservation. An ablation study on the LAC module shows slightly better accuracy but worse performance in exactness and identity preservation. The AFR module improved the performance of manipulation exactness σ_m from 15.37 to 14.76. Using entire attributes in navigator training improved the performance of accuracy from 87.5% to 95.5%. Using intermediate target quantity in navigator training improved the manipulation exactness σ_m from 15.65 to 14.76. By training the navigator with a random intermediate target quantity, the navigator could move the source latent feature to the target more precisely, as intuitively intended.

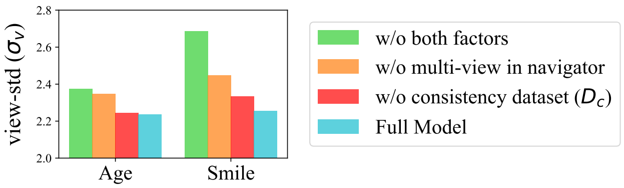

Figure 9. Ablation study on view consistency. We experimented on two attributes (age, smile) and reported the viewpoint-wise standard deviation σ_v (a lower value is better).

View Consistency We conducted another ablation study on view consistency performance. Two ablation factors were selected, one from the quantifier and one from the navigator. In the ablation study on the quantifier, we trained the quantifier without consistency dataset $\mathcal{D}_c$ by setting $\lambda_c = 0$. In the ablation study on the navigator, we trained the navigator without multi-view images by setting $N = 1$. Furthermore, we also experimented with a case without both factors of the quantifier and navigator. As shown in Figure 9, each factor in the quantifier and navigator for multi-view consistency contributes to the view consistency.

5. Conclusion

Using our method, users can easily manipulate custom attributes of the source 3D image with only a target attribute quantity in the range of $[0, 1]$. This goal was achieved by introducing a novel attribute quantifier that can estimate the normalized attribute quantity from a given image. The navigator enables fine-control manipulation of multi-attribute quantities using the attribute quantifier. Our method achieved 3D-aware image manipulation via using a consistent attribute quantity in multi-view. We validated the effectiveness of our method both qualitatively and quantitatively through various experiments.

Limitations Since our study is based on the latent space of a pre-trained GAN model, the performance of manipulation is limited by that of the pre-trained model. EG3D has fewer pre-trained models than StyleGAN2, so examples are limited, but it is expected to expand as various pre-trained models are released. Moreover, we used images generated from random latent features as a source rather than real images. Examples of real image manipulation using the inversion method [31] are provided in the supplement.

Further discussion of the potential negative social impacts is provided in the supplement.

Acknowledgement This work was supported by IITP grant funded by Korea government(MSIT) [No.B0101-15-0266, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis; NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)]

References

[1] Face++ face detection api. https://www.faceplusplus.com/. Accessed: 2023-02-23. 2, 6

[2] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)*, 40(3):1–21, 2021. 1, 2, 4, 6, 7

[3] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3d morphable models. *International Journal of Computer Vision*, 126(2):233–254, 2018. 2

[4] Manuel G Calvo, Andrés Fernández-Martín, Guillermo Recio, and Daniel Lundqvist. Human observers and automated assessment of dynamic emotional facial expressions: Kdef-dyn database validation. *Frontiers in psychology*, 9:2052, 2018. 5

[5] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 1, 2, 3, 5

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 5

[7] Adam Geitgey. Github – face recognition. https://github.com/ageitgey/face_recognition/, 2021. 7

[8] Markos Georgopoulos, Yannis Panagakis, and Maja Pantic. Investigating bias in deep face analysis: The kanface dataset and empirical study. *Image and Vision Computing*, 102:103954, 2020. 2

[9] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5744–5753, 2019. 6

[10] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 1, 2

[11] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 2, 7

[12] Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. Talk-to-edit: Fine-grained facial editing via dialog. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13799–13808, 2021. 7

[13] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18623–18632, 2022. 2

[14] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 5

[15] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 2

[16] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Empirically analyzing the effect of dataset biases on deep face recognition systems. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2093–2102, 2018. 2

[17] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[18] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 6

[19] 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. *arXiv preprint arXiv:2207.10257*, 2022. 2

[20] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020. 5

[21] Yuchen Liu, Zhixin Shu, Yijun Li, Zhe Lin, Richard Zhang, and SY Kung. 3d-fm gan: Towards 3d-controllable face manipulation. *arXiv preprint arXiv:2208.11257*, 2022. 2

[22] Microsoft. Azure face. https://azure.microsoft.com/en-in/services/cognitive-services/face/., 2020. 2, 6

[23] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 4

[24] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019. 1

[25] Thu H Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. *Advances in neural information processing systems*, 31, 2018. 1

[26] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer*

Vision and Pattern Recognition, pages 11453–11464, 2021. 1

[27] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022. 1, 2

[28] Roy Or-El, Soumyadip Sengupta, Ohad Fried, Eli Shechtman, and Ira Kemelmacher-Shlizerman. Lifespan age transformation synthesis. In *European Conference on Computer Vision*, pages 739–755. Springer, 2020. 1, 5

[29] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 1, 2, 4, 7

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2, 3

[31] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42(1):1–13, 2022. 8

[32] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 1

[33] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1, 2

[34] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8269–8279, 2022. 1

[35] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022. 2

[36] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3d face morphable model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1126–1135, 2019. 2

[37] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022. 1, 2

[38] Liwen Wu, Jae Yong Lee, Anand Bhattad, Yu-Xiong Wang, and David Forsyth. Diver: Real-time and accurate neural radiance fields with deterministic integration for volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16200–16209, 2022. 1

[39] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021. 1, 2

[40] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[41] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 2

[42] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 3

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4

[44] Xiaoming Zhao, Fangchang Ma, David Güera, Zhile Ren, Alexander G Schwing, and Alex Colburn. Generative multiplane images: Making a 2d gan 3d-aware. *arXiv preprint arXiv:2207.10642*, 2022. 2

[45] Yiyu Zhuang, Hao Zhu, Xusen Sun, and Xun Cao. Mofanerf: Morphable facial neural radiance field. *arXiv preprint arXiv:2112.02308*, 2021. 2