Class-Continuous Conditional Generative Neural Radiance Field

Jiwook Kim [1] **Minhyeok Lee** [1]

Abstract

The 3D-aware image synthesis focuses on conserving spatial consistency besides generating high-resolution images with fine details. Recently, Neural Radiance Field (NeRF) has been introduced for synthesizing novel views with low computational cost and superior performance. While several works investigate a generative NeRF and show remarkable achievement, they cannot handle conditional and continuous feature manipulation in the generation procedure. In this work, we introduce a novel model, called Class-Continuous Conditional Generative NeRF (C^3G-NeRF), which can synthesize conditionally manipulated photorealistic 3D-consistent images by projecting conditional features to the generator and the discriminator. The proposed C^3G-NeRF is evaluated with three image datasets, AFHQ, CelebA, and Cars. As a result, our model shows strong 3D-consistency with fine details and smooth interpolation in conditional feature manipulation. For instance, C^3G-NeRF exhibits a Fréchet Inception Distance (FID) of 7.64 in 3D-aware face image synthesis with a 128^2 resolution. Additionally, we provide FIDs of generated 3D-aware images of each class of the datasets as it is possible to synthesize class-conditional images with C^3G-NeRF.

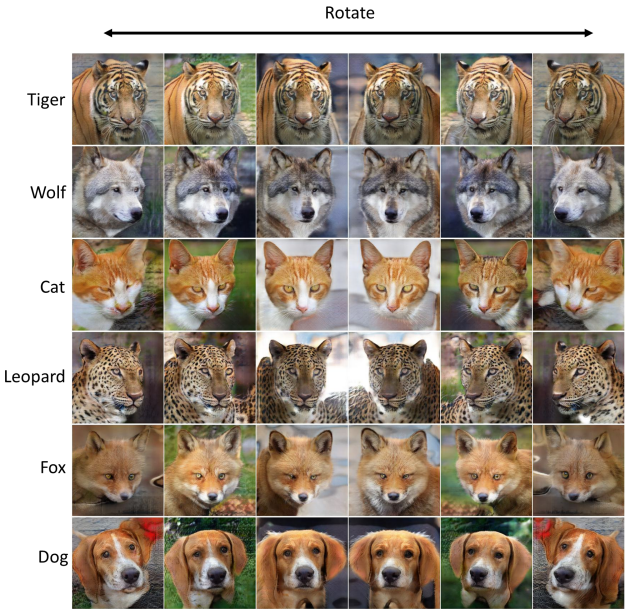


Figure 1. **Synthesized images of each class of AFHQ by our model (with a 256^2 resolution).** A row displays a single object with different rotation input vectors. Note that the images of different classes are generated by a single model with different conditional input vectors. Our model can generate various views of an object that conserves strong 3D-consistency.

1. Introduction

There have been many approaches (Kingma & Welling, 2013; Van Den Oord et al., 2016) for synthesizing novel images. Generative Adversarial Network (GAN) (Goodfellow et al., 2014) has shown outstanding results in generating images by learning the distributions of datasets, resulting in the synthesis of photo-realistic instances. Furthermore, many studies have improved the ability to generate high-resolution images (Chen et al., 2016; Karras et al., 2017; 2020; Choi et al., 2018). Notwithstanding the advances made by existing studies, GANs still have limitations when synthesizing multiple views of a single object due to most data collections being based on two-dimensional information, which can lead to instability in 3D-consistency.

In order to address this problem, 3D-based GANs (Wu et al., 2016; Nguyen-Phuoc et al., 2019; 2020) have been studied, which make use of volume rendering methods. However, those methods require high computational power and memory in order to train effectively. In recent years, Mildenhall et al. (2021) proposed the Neural Radiance Field (NeRF) as an alternative to conventional voxel-based volume rendering methods (Kajiya & Von Herzen, 1984; Drebin et al., 1988; Henzler et al., 2019). NeRF greatly reduces the complexity of computations and memories required compared to other approaches. Accordingly, NeRF has been extended for use in 3D-based GAN studies (Schwarz et al., 2020; Niemeyer

[1]School of Electrical & Electronics Engineering, Chung-Ang University, Seoul 06974, Korea. Correspondence to: Minhyeok Lee <mlee@cau.ac.kr>.

arXiv Preprint

& Geiger, 2021; Chan et al., 2021; Deng et al., 2022) with excellent results and low complexities.

Nevertheless, those NeRF-based GANs cannot control image generations with conditional labels because they do not provide the required conditional information into the generator of the GAN. GIRAFFE (Niemeyer & Geiger, 2021), which is a NeRF-based generative model, was capable of generating 3D-aware images without any extra information, such as camera location and direction of certain images for training, which conventional NeRF required. Even though GIRAFFE demonstrated surpassing results, it is not able to disentangle the various features (Bengio et al., 2013; Locatello et al., 2019; Nguyen-Phuoc et al., 2020) in images that are needed in order to generate an image containing desired features. Jo et al. (2021) tackled this problem by trying to provide condition information, such as image types and texts; however, they cannot represent the intensity of conditions.

In this paper, we propose a novel method to address a new task of 3D-aware conditional image generation. The given task requires that conditional label values (Mirza & Osindero, 2014; Odena et al., 2017; Miyato & Koyama, 2018) be able to control features of generated images, as shown in Figure 1, and that the condition label values be continuous in order to continuously change the intensity of the corresponding conditions. To the best of our knowledge, this paper is the first to tackle this task.

We have named our proposed method Continuous Conditional Generative Neural Radiance Field (C^3G-NeRF). Our model is based on the GIRAFFE, which showed superior performances in 3D-aware image generation. For the conditional generation, our model takes numerical and continuous values as input, which enables continuous feature representations. Most datasets containing conditions are composed of numerical conditions, so it is expected that our model can be applied to numerous datasets.

Conventional NeRF-based generative models often require a lot of time for training. Hence, we employed residual modules (He et al., 2016a;b) in our model architecture to accelerate training speed and synthesize superior images.

Generating multi-views of an instance is a significant problem in generating 3D-aware images. To address this problem, C^3G-NeRF aims at providing multi-view with fine 3D-consistency. We provide results of 3D controllable image synthesis in three datasets which are AFHQ (Choi et al., 2020), CelebA (Liu et al., 2015) and Cars (Ashrafi, 2022). We experiment to show controlling image synthesis by translating and rotating an object as well as adding numbers of the objects in a single image.

The contributions of this paper are as follows:

- We propose a model called C^3G-NeRF to tackle a novel task: conditional and continuous feature manipulation in 3D-aware image generation.
- We reduced training time and increased performance by using residual modules in the NeRF architecture.
- We demonstrated the conditional and continuous feature manipulation in 3D-aware image generation with multiple datasets: AFHQ, CelebA and Cars.
- Since it is possible to generate class-conditional 3D-aware images, we provide Fréchet Inception Distance (FID) of each label in AFHQ and Cars datasets.

2. Related Work

Implicit neural representation and rendering: The use of deep learning techniques (LeCun et al., 2015) to represent three-dimensional space has received considerable attention in recent years. Among the various approaches that have been proposed, implicit neural representations (Sitzmann et al., 2019; Tulsiani et al., 2020; Rajeswar et al., 2020) have shown particular promise. NeRFs have been proposed by combining an implicit neural representation with volume rendering (Drebin et al., 1988) to enable the synthesis of novel views that are not explicitly represented in the training data. As a result, NeRF is capable of synthesizing 3D-consistent images with fine details. This is accomplished by mapping a 3D point and direction onto an RGB color and corresponding volume density using Multi-Layer Perceptrons (MLPs) (Pinkus, 1999) which learn an implicit representation function (Mescheder et al., 2019; Chabra et al., 2020; Yariv et al., 2020). However, one downside to using NeRFs is that they require highly constrained images for supervision during training. Another concern is that each instance of a NeRF can only represent a single object rather than multiple objects simultaneously. It has been proposed that generative NeRFs may be able to alleviate these problems.

Generative NeRF: There has been some recent progress in NeRF-based methods for generating 3D-aware images from 2D unconstrained image datasets. In these methods, generative models are trained to ensure continuous 3D geometric consistency. For instance, the GRAF (Schwarz et al., 2020) and pi-GAN (Chan et al., 2021) both proposed a generative NeRF, and showed promising results. GIRAFFE is another method that is more closely related to our work and improves on GRAF by separating an object from its background scene. However, none of these methods can control the conditional generation of images, which enables various feature manipulation in generated images. To address this issue, Jo et al.(2021) proposed CG-NeRF; however, since CG-NeRF utilizes pi-GAN structures instead of GIRAFFE, it removes background scene in most of their datasets while

training. Specifically, CG-NeRF takes two types of conditional information. First, conditional data forms are used to translate one image into another; while our study aims to generate novel images directly from noise vectors instead. Second, CG-NeRF takes texts as conditions with CLIP (a natural language processing technique) (Radford et al., 2021); whereas our model uses numerical conditions, which can be interpolated by varying values, allowing for continuous feature manipulation in image synthesis that represents intensity changes in relation to the given values.

3. Methods

Our goal is to make a framework for conditionally controllable 3D-aware image synthesis that guarantees representations of the intensity of conditions with continuous values. Given a labeled real-world 2D image dataset, the 3D-aware image generator G takes conditions and camera pose, which are denoted by $\mathbf{x}$ and $\mathbf{d}$, respectively, and latent vectors for representing shape and appearance, i.e., $\mathbf{z_s}$ and $\mathbf{z_a}$. Then, $\mathbf{G}$ produces an image $\mathbf{I}$, corresponding to the input condition. At training time, real images from the dataset and $\mathbf{I}$ are directed to the discriminator $\mathbf{D}$, which is comprised of residual modules (He et al., 2016a), while conventional convolutional layers (O'Shea & Nash, 2015) and fully-connected layers (Pinkus, 1999) have been used in other generative NeRF models in recent years. Figure 2 shows an overview of our model.

3.1. Conditional Neural Radiance Fields

A conventional neural radiance field maps a 3D coordinate $\mathbf{x} = (x, y, z)$ and viewing direction $\mathbf{d} = (\theta, \phi)$ to a volume density σ and a view-dependent RGB color value R, G, and B with fully-connected layers. However, several studies (Rahaman et al., 2019) observed that deep learning techniques are difficult to represent high-frequency details, especially with low-dimensional inputs. To diversify the inputs, NeRFs introduced positional encoding (Tancik et al., 2020) to the inputs, $\mathbf{x}$ and $\mathbf{d}$, before the fully-connected layers:

$$\begin{aligned}\gamma(p, L) =&(\sin(2^0\pi p), \cos(2^0\pi p), \sin(2^1\pi p), \\ &\cos(2^1\pi p), \cdots, \sin(2^{\mathrm{L}-1}\pi p), \cos(2^{\mathrm{L}-1}\pi p)),\end{aligned} \tag{1}$$

where $\gamma(\cdot)$ indicates positional encoding, L is the dimensionality of positional encoding, and p is a scalar value as a component of x and d. To extend to generative neural radiance fields, shape and appearance codes, $\mathbf{z_s}$ and $\mathbf{z_a}$, are fed into the MLP as follows:

$$(\gamma(\mathbf{x}), \gamma(\mathbf{d}), \mathbf{z_s}, \mathbf{z_a}) \mapsto (\sigma, \mathrm{R}, \mathrm{G}, \mathrm{B}). \tag{2}$$

In GIRAFFE, the generative neural radiance field was substituted to generative neural feature fields by extending the dimensionality of color (Niemeyer & Geiger, 2021), which was originally three-dimensional, to a feature space having a dimension of M_f as:

$$\begin{aligned}\mathrm{h}_\theta : \mathbb{R}^{\mathrm{L_x}} \times \mathbb{R}^{L_\mathrm{d}} \times \mathbb{R}^{M_\mathrm{s}} \times \mathbb{R}^{M_\mathrm{a}} &\mapsto \mathbb{R} \times \mathbb{R}^{M_f} \\ (\gamma(\mathbf{x}), \gamma(\mathbf{d}), \mathbf{z_s}, \mathbf{z_a}) &\mapsto (\sigma, \mathbf{f}).\end{aligned} \tag{3}$$

where h_θ represents a generative neural feature field, $L_\mathbf{x}$ and $L_\mathbf{d}$ indicate dimensionalities of positional encoding output, and $M_\mathbf{s}$ and $M_\mathbf{a}$ are dimensionalities of latent encodings of the shape and appearance, respectively.

In this work, we project conditions to the latent vectors (Miyato & Koyama, 2018), $\mathbf{z_s}$ and $\mathbf{z_a}$, by element-wise production. Before the projection, conditional vectors $\mathbf{c}$ having a dimension of $M_\mathbf{c}$ are encoded by a fully-connected layer to make the dimension the same as the dimensionalities of $\mathbf{z_s}$ and $\mathbf{z_a}$. The shape and appearance conditional encodings, $\mathbf{c_s}$ and $\mathbf{c_a}$, are constructed as

$$\begin{aligned}&\mathbf{c_s} = L_\mathbf{s}(\mathbf{c}) * \mathbf{z_s}, \quad \mathbf{c_a} = L_\mathbf{a}(\mathbf{c}) * \mathbf{z_a}, \\ &L_\mathbf{s} : \mathbb{R}^{M_\mathbf{c}} \mapsto \mathbb{R}^{M_\mathbf{s}}, \; L_\mathbf{a} : \mathbb{R}^{M_\mathbf{c}} \mapsto \mathbb{R}^{M_\mathbf{a}}\end{aligned} \tag{4}$$

where $L_\mathbf{s}$ and $L_\mathbf{a}$ are the encoding layers for the shape and appearance, respectively, and $*$ denotes element-wise multiplication. We employ these conditional projections as a replacement for conventional latent vectors in generative feature fields:

$$\begin{aligned}h_\theta : \mathbb{R}^{L_\mathbf{x}} \times \mathbb{R}^{L_\mathbf{d}} \times \mathbb{R}^{M_\mathbf{s}} \times \mathbb{R}^{M_\mathbf{a}} &\mapsto \mathbb{R} \times \mathbb{R}^{M_f}, \\ (\gamma(\mathbf{x}), \gamma(\mathbf{d})), \mathbf{c_s}, \mathbf{c_a} &\mapsto (\sigma, \mathbf{f}).\end{aligned} \tag{5}$$

3.2. Scene compositions

As our model is motivated by GIRAFFE, it can separate scenes and individual objects with multiple feature vectors (Niemeyer & Geiger, 2021). Each entity requires a generative feature field and is encoded by a respective feature vector. Consequently, in the model, there are N generative feature fields when $N-1$ objects and a background scene exist in an image. To composite N entities, the composition operator $C(\cdot)$ composites all feature fields from the N entities. As previously described, each single entity $h_{\theta i}^i$ outputs a volume density θ_i and a feature vector $\mathbf{f}_i$. Following the method in GIRAFFE, we composite each component of entities with density-weighted mean-based composition. The mathematical expression of the composition operator can be represented as

$$C(\mathbf{x}, \mathbf{d}, \mathbf{c}) = \Big(\sigma, \sum_{i=1}^{N} \frac{\sigma_i \mathbf{f}_i}{\sigma}\Big), \tag{6}$$

where $\sigma = \sum_{i=1}^{N} \sigma_i$.

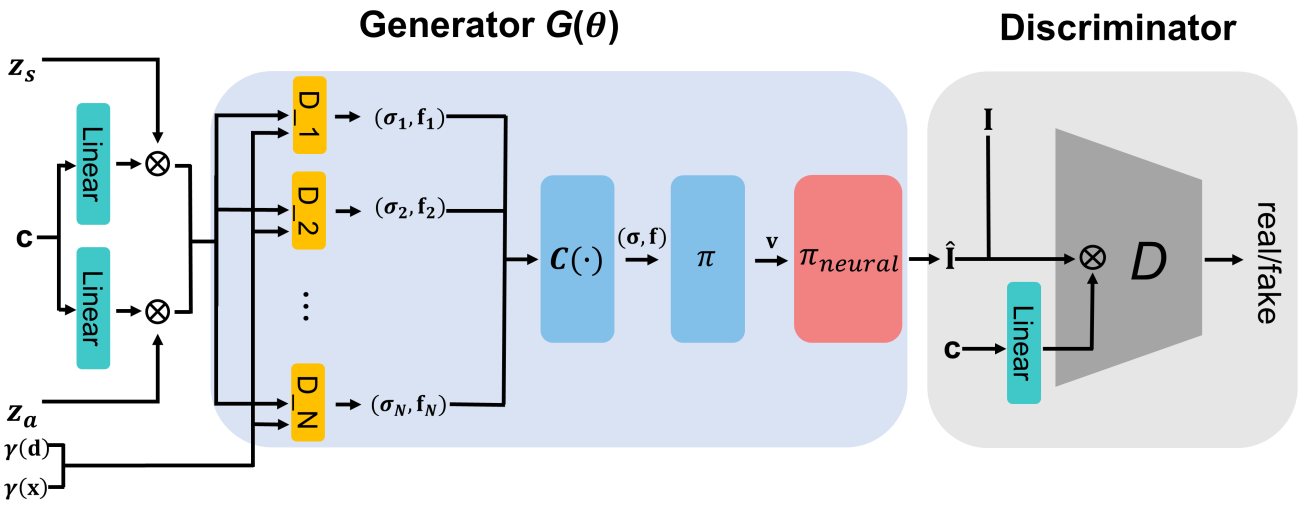


Figure 2. **Overview of the proposed C^{3}G-NeRF.** Since our model is inspired by the architecture of GIRAFFE, our model generates $N-1$ objects and the background with N decoders and a composition operator. D_i indicates ith decoder and $C(\cdot)$ represents the composition operator. The decoders take a 3D coordinate vectors of positional encoding $\gamma(\mathbf{x})$ and viewing direction $\gamma(\mathbf{d})$, where γ indicates positional encoding functions. In addition, the decoders take conditional vectors $\mathbf{c}$, which are encoded by linear layers, shape codes $\mathbf{z_s}$, and appearance codes $\mathbf{z_a}$. By compositing the outputs of each decoders with the composition operator $C(\cdot)$ and then volume-renders the result. Consequently, a composited feature vector $\mathbf{v}$ is produced. The feature vector $\mathbf{v}$ passes the neural rendering module π_{neural}. In this process, the generator $G(\theta)$ synthesizes a fake image $\hat{I}$. The discriminator D takes a real image I or the fake image $\hat{I}$ projected by the conditional labels $\mathbf{c}$.

3.3. Volume rendering and neural rendering

For scene rendering, previous studies (Schwarz et al., 2020) have volume-rendered a camera ray $r(t) = \mathbf{o} + t\mathbf{d}$ directly to RGB colors, while GIRAFFE volume-renders it to feature vectors (Niemeyer & Geiger, 2021), and subsequently, neural-renders the feature vectors to generate synthetic images. Our approach basically follows a discretized form of volume rendering methods used in NeRF (Mildenhall et al., 2021); nonetheless, detailed methods follow those of GIRAFFE to render features to images, which can be represented as

$$\mathbf{v} = \sum_{j=1}^{N_s} T_j(1 - e^{-\sigma_j \delta_j}) f_j, \text{where } T_j = \prod_{k=1}^{j-1} e^{-\sigma_j \theta_j}, \quad (7)$$

where $\mathbf{v}$ represents a final feature vector, T denotes an accumulated transmittance along the cast ray, and $N_\mathbf{N}$ is the number of sample points along a cast ray for an arbitrary camera pose ξ. By sampling points along the camera ray, we utilize a feature vector f_j and a density σ_j corresponding to each point. Furthermore, the feature images have a 16^2 resolution, i.e., $H_V \times W_V$ for cost-effectiveness. Existing studies, including GRAF and NeRF (Schwarz et al., 2020; Mildenhall et al., 2021), commonly adopted the volume rendering approach for 3D-to-2D projections. However, in our model, an additional 2D neural rendering network is required since the volume rendering composes feature images, not colored high-resolution images. The additional neural rendering network,

$$\pi_{neural} : \mathbb{R}^{H_V \times W_V \times M_f} \mapsto \mathbb{R}^{H \times W \times 3}, \quad (8)$$

maps outputs of the volume rendering to a synthetic image, $\hat{\mathbf{I}} \in \mathbb{R}^{H \times W \times 3}$, by upsampling the feature images. The architecture of neural rendering network is similar to a neural rendering operator in existing studies (Niemeyer & Geiger, 2021); however, we hire residual modules instead of conventional convolutional networks to enhance training speed and performance.

3.4. Training details

At training time, we randomly sample latent vectors $\mathbf{z_s}$ and $\mathbf{s_a}$, as well as camera pose ξ from prior distributions $p_\mathbf{s}$, $p_\mathbf{a}$, and p_ξ. Prior distributions $p_\mathbf{s}$ and $p_\mathbf{a}$ are defined as Gaussian distributions, and camera pose distribution ξ are set to a uniform distribution. In the generator G, we project conditional labels to latent vectors (Miyato & Koyama, 2018) $\mathbf{z_s}$ and $\mathbf{z_a}$. Similarly, conditional labels are projected to real images $\mathbf{I}$ or synthesized images $\hat{\mathbf{I}}$, in the discriminator D. Real images $\mathbf{I}$ are randomly sampled from the training dataset, which follows distribution $p_\mathbf{I}$. We use a GAN loss with R1 gradient

penalty (Mescheder et al., 2018) as follows:

$$\mathcal{L}(G,D) = \mathbb{E}_{\mathbf{z_s}\sim p_s, \mathbf{z_a}\sim p_a, \xi\sim p_\xi}\big[-\log(D(G(\mathbf{z_s},\mathbf{z_a},\xi,\mathbf{c})))\big] + \mathbb{E}_{\mathbf{I}\sim p_\mathbf{I}}\big[-\log(1-D(\mathbf{I})) + \lambda\|\nabla D(\mathbf{I})\|^2\big]. \tag{9}$$

We train the generator and discriminator of the proposed model by competing for a zero-sum game with the above loss function. We use the RMSprop optimizer (Ruder, 2016) with a learning rate of 3×10^{-4} and 1×10^{-4} for the generator and the discriminator, respectively. The model is trained on one A6000 GPU with 48GB memory with a batch size of 32. We set $M_f = 128$ for a 64^2 resolution and a 128^2 resolution, while we set $M_f = 256$ for a 256^2 resolution. We utilize ReLU activations (Nair & Hinton, 2010) as activation functions used in our model, except for the final layer in the neural renderer, which uses a sigmoid function (Ramachandran et al., 2017).

3.5. Accelerating training with residual modules

We observe elaborate conditional generation with GIRAFFE is not feasible without residual modules (He et al., 2016a; Mescheder et al., 2018). In the experiments of this study, we demonstrate that conditional generation with plain networks shows lagging performance. We argue that this result is because conditionally projected latent vectors are too far from the discriminator, which causes gradient vanishing. Therefore, residual modules support conveying the information to conditionally projected latent vectors from the discriminator. Moreover, the range of varying latent vectors expands since we projected conditional labels to latent vectors, which is challenging to learn with plain networks. We apply residual modules to our model in the decoder, neural rendering network, and discriminator. We validate the efficiency of this contribution in the Section 4.

4. Experiments

In this section, we evaluate our C^{3}G-NeRF on three real-world datasets: CelebA (Liu et al., 2015), AFHQ (Choi et al., 2020), and Cars (Ashrafi, 2022). CelebA contains approximately 200,000 face images with 40 conditional binary labels. AFHQ contains 16,000 high-resolution animal faces of 7 species; Cars contains 9,737 car images of 13 car models with various camera views. We train and evaluate 128^2 resolution for all datasets, 64^2 for CelebA and AFHQ, and 256^2 for AFHQ and Cars.

We first evaluate the conditionally controlling 3D-consistent image generations. We then evaluate the quality of generation by FIDs (Heusel et al., 2017). Furthermore, we show training speed by comparing with synthesized images at each iteration in the training process. Finally, we include an evaluation of residual modules to validate the efficiency of residual modules adopted in our model.

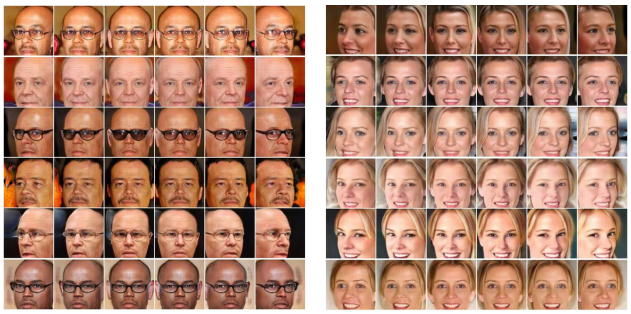


Figure 3. **Class conditional synthetic object rotation generated by C^{3}G-NeRF trained with CelebA.** Each row represents a single object with the same latent vectors. Each column indicates rotation angles. In (a), we fixed the input conditions as a bald man, whereas we fixed the conditions as a blonde smiling woman in (b). All images have a resolution of 128^2. Using C^{3}G-NeRF, 3D-consistent image generation is successful under the given conditions.

4.1. Controllable features in 3D object generation

We evaluate our model by rotating, translating depth and horizontal, and adding objects. Figures 1, 3 and 4 demonstrates that C^{3}G-NeRF successfully learns 3D-consistency with fine details for AFHQ, CelebA, and Cars, respectively, since the performed rotation and translations of objects are conserving the spatial consistency. Also, we confirm that features of the generated images successfully follow conditional inputs.

Furthermore, in Figure 4, we assess C^{3}G-NeRF using out-of-distribution images with translating depth and horizontal at test time. This means C^{3}G-NeRF can generate beyond the distribution of training images by using extended rotation and transition values over the training sets. Moreover, we can finely control each object and scene in the generated images by C^{3}G-NeRF; for instance, while adding the objects in a scene, each object can be controlled by translating and rotating in 3D space.

4.2. Continuously controllable features in 3D object generation

To show the ability to manipulate each condition, we analyze the model by interpolating 40 conditional binary values of the CelebA dataset, such as chubby, smiling, blonde, and pale skin. While the range of conditional values is zero and one in the training procedure since the labels are binarily encoded in the CelebA dataset, in the test procedure, we further provide extrapolated values by expanding the range to zero to three, as shown in Figure 5. This experiment can demonstrate whether each face feature is mapped into each input label of the generator. With various conditional label values, C^{3}G-NeRF shows superior performances in interpolation (Davis, 1975) and extrapolation (Brezinski & Zaglia, 2013) of each condition. For example, chubby and

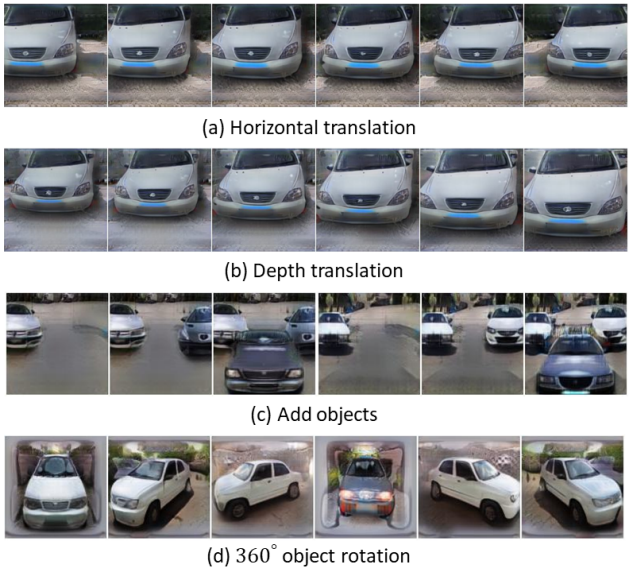


Figure 4. **Visualization of various 3D-aware manipulation in Cars.** By controlling the horizontal and depth translation, the disentanglement of the objects and background are shown in (a) and (b). After training with unstructured 2D images with a single object, we can generate $N-1$ objects in one scene by replicating N decoders as in (c). Furthermore, our model enables 360 ° object rotation for each object in the scene as in (d).

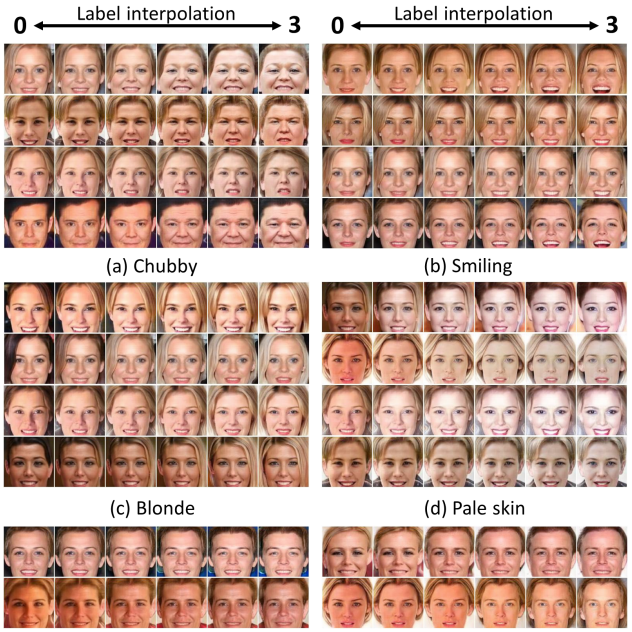


Figure 5. **Interpolation and extrapolation on conditional input values.** Each row represents a single object. We present the diverse conditional results according to conditional input values with the range of zero to three. Note that, in training time, the features are trained only with the two values of zero and one, which indicate the existence of the corresponding feature. The features in the face images with interpolated and extrapolated input values smoothly change, which means the continuous conditional learning is adequately progressed.

smiling conditions (Liu et al., 2015) smoothly change according to conditional values regardless of the extrapolation range. This result also indicates the ability of our model to generate out-of-distribution images; for instance, exaggerated features can be generated with a high input label value exceeding one.

We experiment with non-characteristic (implicit) labels, corresponding to categories (classes) of AFHQ. This class label is more challenging to learn since the image features of each class are not obvious. We generate inter-class images with interpolated class-conditional input values as shown in Figure 6. We observe that features of each class coexist at the intermediate state of class-conditional values. This result indicates that the ability of feature manipulation can be applied to implicit class labels, which is generally more challenging to be trained.

4.3. Diversity of generation

We verify that our model can synthesize diverse images under the same conditions. To verify diverse generations, we fix the conditional input values and interpolate the values of the latent vectors (Shmelkov et al., 2018). With varying the latent vectors, our model generates various intermediate images with conserving the fixed conditional features, as shown in Figure 7. This is due to our generator can disentangle the condition to the latent vectors and magnificently learn the distribution of each conditional case.

Table 1. **Quantitative comparison with FIDs (↓) with three datasets.** The baseline is set to the conditional GIRAFFE with plain networks. The 64^2, 128^2, and 256^2 are the image resolutions of the generated images and real images.

Model	AFHQ			CelebA		Cars	
	64^2	128^2	256^2	64^2	128^2	128^2	256^2
Baseline	212.74	226.51	239.01	55.90	83.89	244.55	266.51
Ours	26.72	25.79	28.58	5.60	7.64	43.29	31.63

4.4. Quantitative evaluation

We evaluate the image quality by measuring FIDs with 20,000 randomly sampled real images and 20,000 generated images, which is one of the conventional methods to assess generative NeRFs. We furnish two evaluations to demonstrate the quality of generation on each dataset as well as each label in the dataset. We evaluate images generated with random conditional inputs by FID using different image resolutions and datasets.

As can be seen in Table 1, C^{3}G-NeRF achieves prominent FIDs in the conditional 3D-consistent generation. We observe that FIDs of C^{3}G-NeRF outperform the conditional GIRAFFE irrespective of resolution; this signifies that C^{3}G-

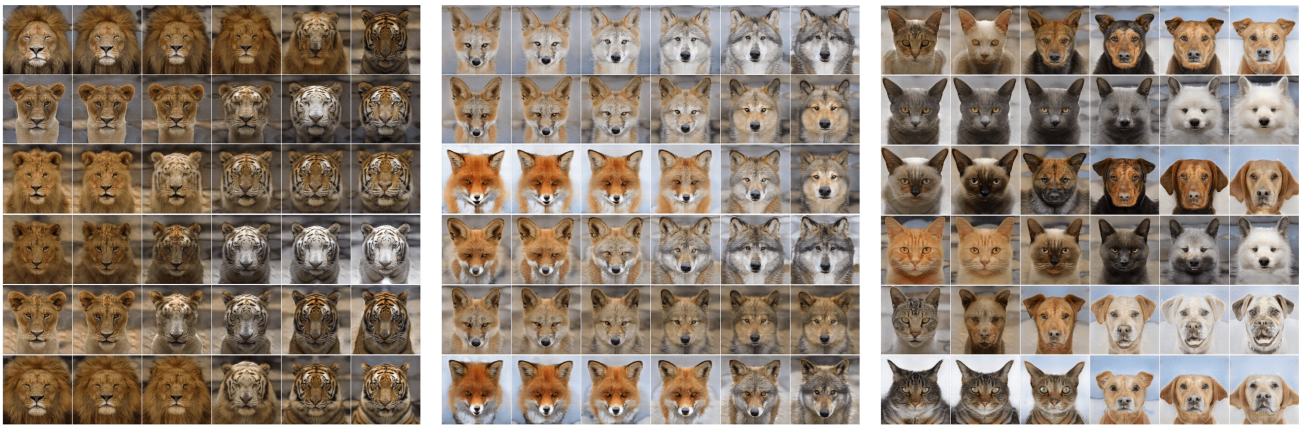


Figure 6. **Class interpolation on conditional input values with the AFHQ dataset.** Each row and column represent the same latent vectors (identical object) and the same class-conditional values, respectively. By interpolating the values of each class, features of each category coexist at the intermediate state of class-conditional values.

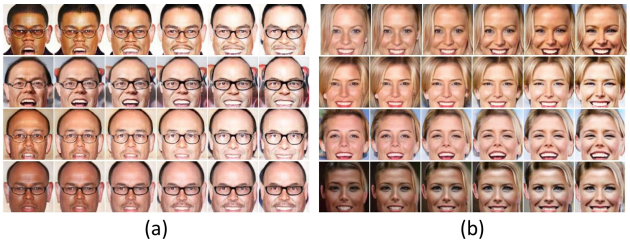


Figure 7. **Interpolating latent vectors.** In (a) and (b), the leftmost images and rightmost images are generated with the same conditional inputs but different latent vectors. Intermediate images are generated with the interpolated values of the two latent vectors. By interpolating values of the latent vectors, our model offers diverse images with a certain condition (i.e. a smiling blonde woman or a glassed man). The conditional input labels of (a) and (b) are fixed as a glassed man and a smiling blonde woman, respectively. We observe that various images can be synthesized with the same condition, which demonstrates the properties of the generative model.

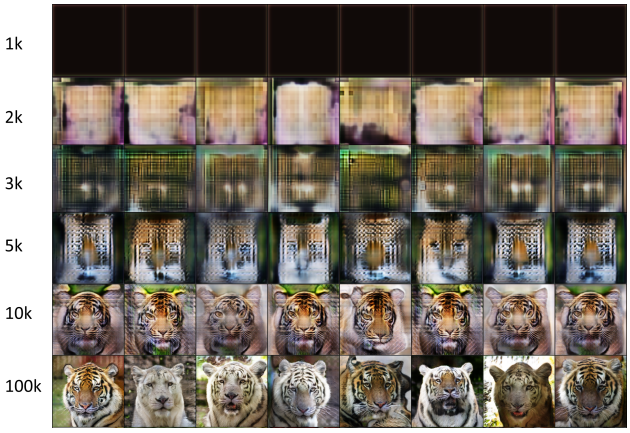


Figure 8. **Training results in terms of iteration.** By employing residual modules, our model produces diverse and high-quality images with fine details within a small number of iterations.

NeRF is robust to training in high resolutions. The conditional GIRAFFE exhibits considerably high FIDs, indicating training problems. For example, in the CelebA face image generation with a 128^2 resolution, C^{3}G-NeRF demonstrates an FID value of 7.64, reducing the value by 90.9% compared to the baseline. However, in the Car dataset, we notice that FIDs are relatively inferior to the other datasets; we attribute this to a difference in view degree (360° in Car dataset vs limited degrees in other datasets). Consequently, we believe that the FID evaluation may not be appropriate for different views when comparing results across datasets.

The FIDs for each label in Table 2 show the level of conditional disentanglement that has been achieved by C^{3}G-NeRF. We can see that the FIDs for each label are similar to, or better than, the FIDs for all labels combined. This suggests that C^{3}G-NeRF is able to satisfactorily learn the features associated with each condition across all datasets.

In addition, even in cases where some labels have very few examples (e.g., foxes, lions, and tigers in AFHQ), we still observe comparable FIDs suggesting that C^{3}G-NeRF is adept at learning from data regardless of its unposed distribution. In the DOG class of the AFHQ dataset, the FIDs are inferior to other classes. This result is due to the variety of dog images while the other classes consist of similar images.

4.5. Evaluation of residual modules

We compare C^{3}G-NeRFs with residual modules and conditional GIRAFFE with a plain network in terms of generated image quality. This experiment demonstrates the effect of adopting residual modules (He et al., 2016a; Mescheder et al., 2018) in C^{3}G-NeRF architecture. As shown in Figure 8, C^{3}G-NeRF generates high-quality images with fine details after a small number of iterations. In contrast, the

Table 2. **Quantitative comparison with FIDs (↓) with each class of AFHQ and Cars.** The 64^2, 128^2, and 256^2 are the image resolutions of the generated images and real images. The FIDs of 64^2 resolution of the Cars dataset are not evaluated due to training challenges with full-degree images under a low resolution.

Category	AFHQ		
	64^2	128^2	256^2
CAT	10.38	13.65	15.48
DOG	31.64	43.37	51.73
LEOPARD	17.20	14.98	13.42
FOX	25.97	28.01	22.50
LION	8.76	12.20	7.61
TIGER	12.78	8.96	5.93
WOLF	28.39	30.82	14.85
Category	**Cars**		
	64^2	128^2	256^2
PEYKAN	-	67.47	68.57
QUIK	-	62.71	49.64
SAMAND	-	50.53	61.01
PEUGEOT-PARS	-	66.27	46.45
PEUGEOT-207I	-	71.06	52.99
PRIDE-111	-	62.28	68.84
PRIDE-131	-	57.70	65.43
TIBA2	-	61.79	55.57
RENAULT-L90	-	65.51	76.84
NISSAN-ZAMIAD	-	88.83	133.77
PEUGEOT-206	-	61.55	128.10
PEUGEOT-405	-	66.15	71.45
MAZDA-2000	-	77.23	104.53

conditional GIRAFFE using plain networks shows poor generation quality as seen in Figure 9 with more (five times larger) iterations. Therefore, we can confirm that it is necessary to use the proposed residual modules in the NeRF structures in order to generate high-quality conditional images.

We further compare the FIDs of each model in three datasets. As can be seen in Table 1, C^{3}G-NeRF demonstrates significantly better FIDs than conditional GIRAFFE with plain networks in the qualitative comparison. This confirms our hypothesis that residual modules improve the generation quality and is extremely helpful for the conditional 3D-aware generation. As previously described, the residual modules assist the learning of conditional information by helping the gradient flow from the output of the discriminator to the conditional inputs of the generator.

5. Conclusions

We presented a novel model, C^{3}G-NeRF, for conditional and continuous feature manipulation in 3D-aware image generation. Our core idea is projecting conditional labels with encoding layers to the latent vectors of the generator and an intermediate layer of the discriminator in order to disentangle features in a dataset. C^{3}G-NeRF incorporates residual modules for facilitating 3D-aware conditional training. By interpolating and extrapolating the conditional input values, we demonstrated 3D-consistent image generation with elaborate manipulations of the intensity of features. We believe that our method pioneers a new conditional generation with NeRF.

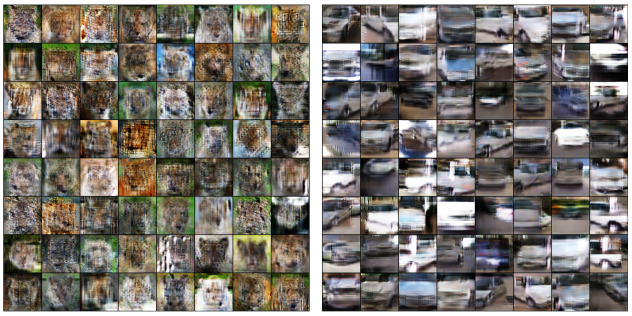


Figure 9. **Generated objects by the conditional GIRAFFE with plain networks trained with 500,000 iterations.** (a) and (b) show the synthesized images with plain networks-based conditional GIRAFFE trained in AFHQ and Cars, respectively. Despite enough iterations (five times larger compared to our model), the generator synthesizes low-quality images.

References

Ashrafi, Y. Iran cars, Aug 2022. URL https://www.kaggle.com/datasets/usefashrfi/iran-used-cars-dataset.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (8):1798–1828, 2013.

Brezinski, C. and Zaglia, M. R. *Extrapolation methods: theory and practice*. Elsevier, 2013.

Chabra, R., Lenssen, J. E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., and Newcombe, R. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision*, pp. 608–625. Springer, 2020.

Chan, E. R., Monteiro, M., Kellnhofer, P., Wu, J., and Wetzstein, G. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5799–5809, 2021.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems*, 29, 2016.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, pp. 8789–8797, 2018.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8188–8197, 2020.

Davis, P. J. *Interpolation and approximation*. Courier Corporation, 1975.

Deng, Y., Yang, J., Xiang, J., and Tong, X. Gram: Generative radiance manifolds for 3d-aware image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10673–10683, 2022.

Drebin, R. A., Carpenter, L., and Hanrahan, P. Volume rendering. *ACM Siggraph Computer Graphics*, 22(4): 65–74, 1988.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645. Springer, 2016b.

Henzler, P., Mitra, N. J., and Ritschel, T. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9984–9993, 2019.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Jo, K., Shim, G., Jung, S., Yang, S., and Choo, J. Cg-nerf: Conditional generative neural radiance fields. *arXiv preprint arXiv:2112.03517*, 2021.

Kajiya, J. T. and Von Herzen, B. P. Ray tracing volume densities. *ACM SIGGRAPH Computer Graphics*, 18(3): 165–174, 1984.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and pattern Recognition*, pp. 8110–8119, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pp. 4114–4124. PMLR, 2019.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pp. 3481–3490. PMLR, 2018.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Miyato, T. and Koyama, M. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 2010.

Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., and Yang, Y.-L. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7588–7597, 2019.

Nguyen-Phuoc, T. H., Richardt, C., Mai, L., Yang, Y., and Mitra, N. Blockgan: Learning 3d object-aware scene representations from unlabelled images. *Advances in*

Neural Information Processing Systems, 33:6767–6778, 2020.

Niemeyer, M. and Geiger, A. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11453–11464, 2021.

Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*, pp. 2642–2651. PMLR, 2017.

O'Shea, K. and Nash, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

Pinkus, A. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pp. 5301–5310. PMLR, 2019.

Rajeswar, S., Mannan, F., Golemo, F., Parent-Lévesque, J., Vazquez, D., Nowrouzezahrai, D., and Courville, A. Pix2shape: Towards unsupervised learning of 3d scenes from images using a view-based representation. *International Journal of Computer Vision*, 128(10):2478–2493, 2020.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Ruder, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Schwarz, K., Liao, Y., Niemeyer, M., and Geiger, A. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33: 20154–20166, 2020.

Shmelkov, K., Schmid, C., and Alahari, K. How good is my gan? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 213–229, 2018.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33: 7537–7547, 2020.

Tulsiani, S., Kulkarni, N., and Gupta, A. Implicit mesh reconstruction from unannotated image collections. *arXiv preprint arXiv:2007.08504*, 2020.

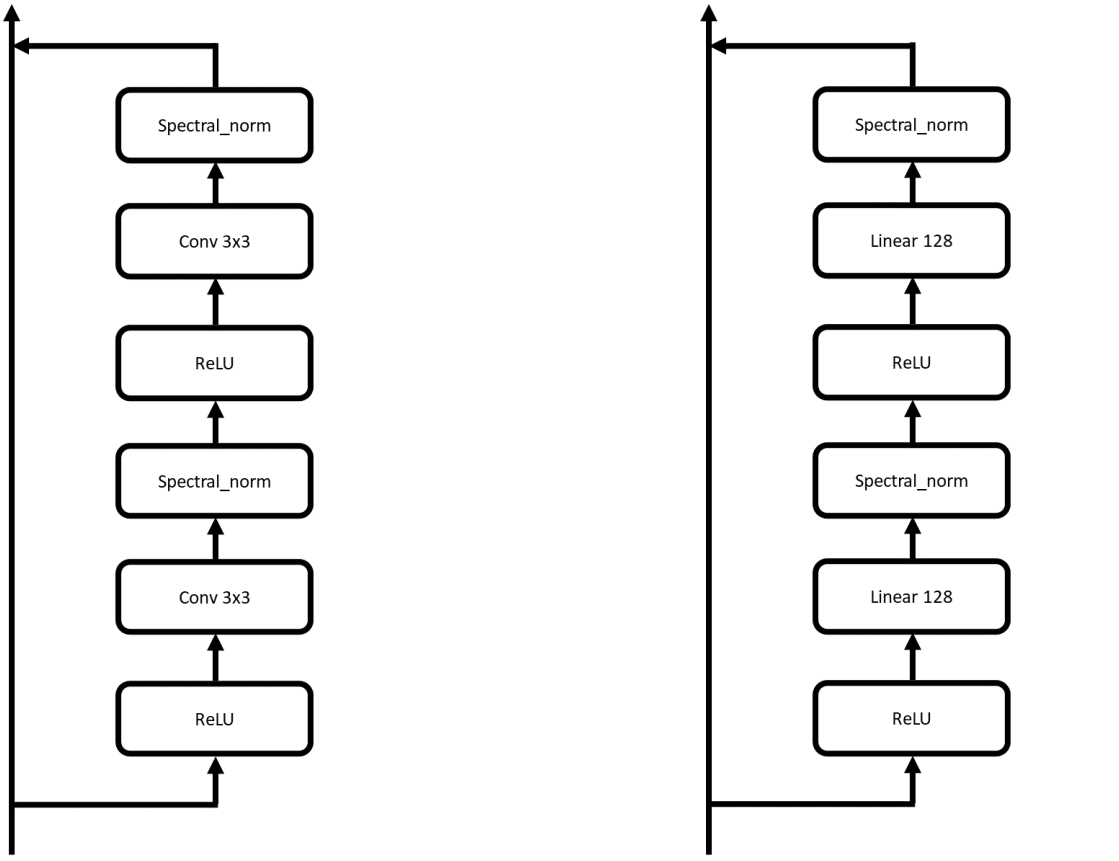
Van Den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pp. 1747–1756. PMLR, 2016.

Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in Neural Information Processing Systems*, 29, 2016.

Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., and Lipman, Y. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33: 2492–2502, 2020.

A. Model details

In this section, we discuss our detailed network architectures with residual modules. We adopt the residual modules to the discriminator, decoder, and neural renderer as Figures 10 to 12.



(a) ResBlock architecture with Convolutional layers.

(b) ResBlockFC architecture with Linear layers.

Figure 10. **Architectures of ResBlocks used in our model.** We employ two types of residual modules, the ResBlock and ResBlockFC. While the ResBlock is comprised of convolution layers, the ResBlockFC is based on linear layers. Therefore, we utilize the ResBlock for the discriminator and the neural renderer, which require spatial information, whereas the ResBlockFC is utilized for the decoder, which maps a 3D coordinate, a viewing direction, and conditional encodings to a feature space and a volume density with linear layers.

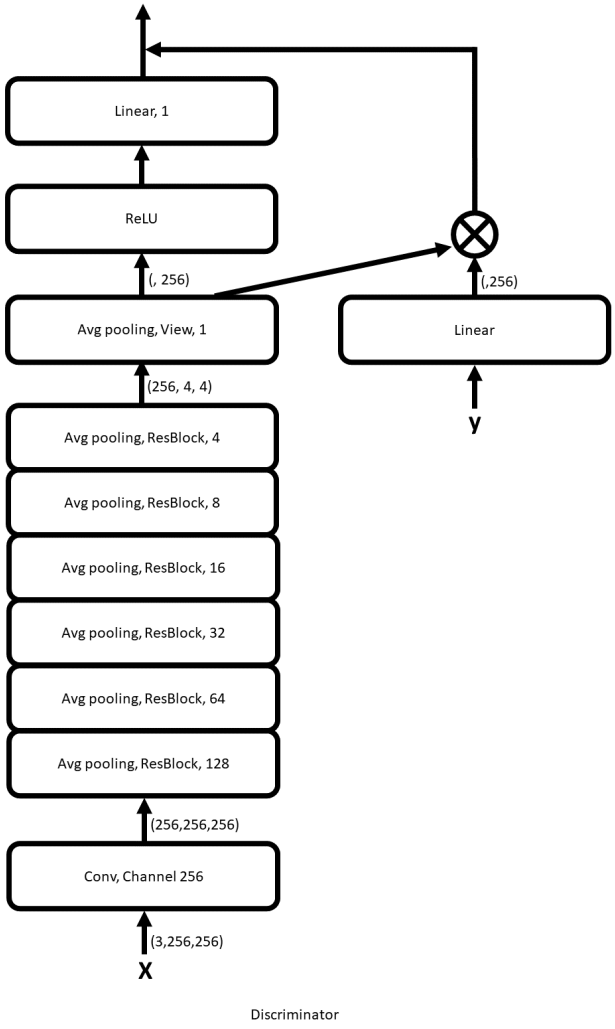


Figure 11. **Architecture of the discriminator.** The **x** and **y** indicate an input image of the discriminator and a conditional label, respectively. Each **x** and **y** are embedded with the convolutional layer and the linear layer. The embedded image is encoded by passing several average pooling layers and ResBlocks. The encoded image is projected to the embedded conditional label. By the projection, we can fuse information of the image and conditional label.

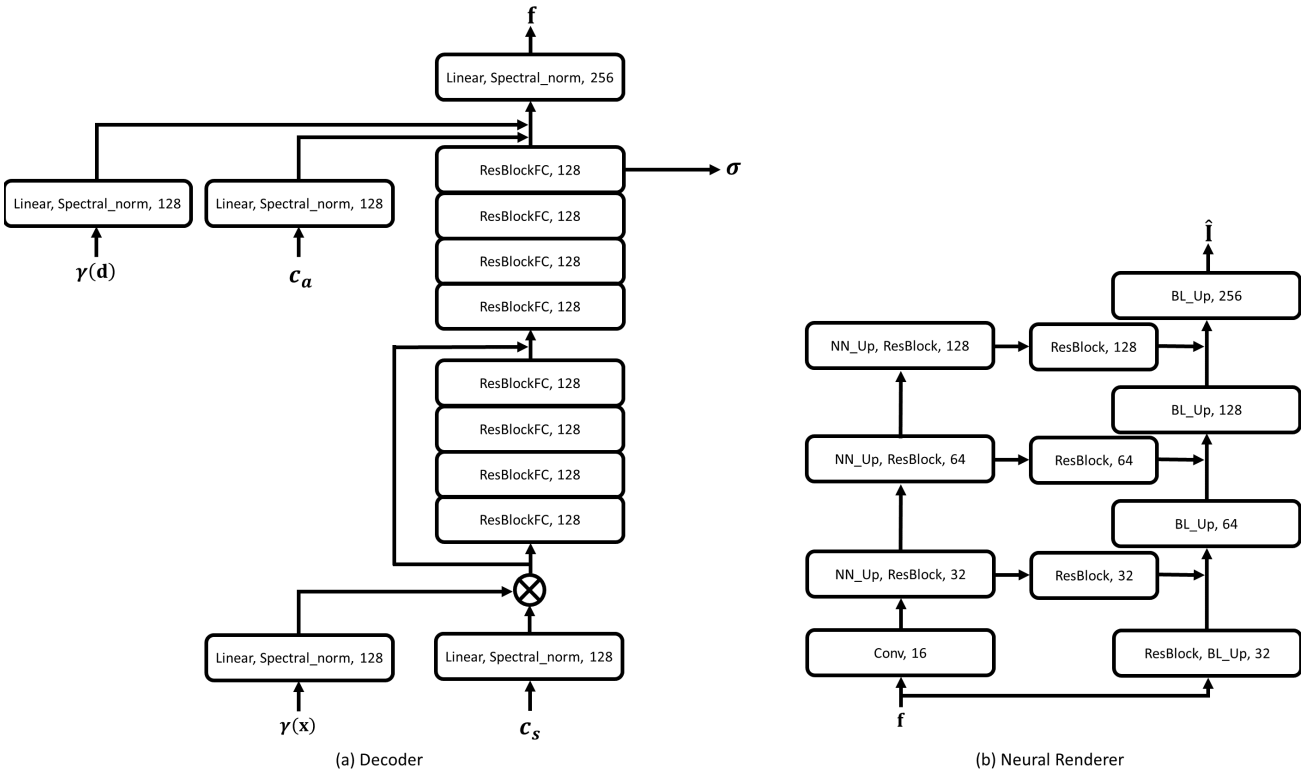


Figure 12. **Architectures of the decoder and neural renderer.** The (a) and (b) show the architecture of the decoder and neural renderer, respectively. In (a), the shape conditional encoding $\mathbf{c}_\mathbf{s}$ and the positional encoded 3D point $\gamma(\mathbf{x})$ are embedded and multiplied. We use 8 blocks of the ResBlockFC and one skip-connnection. After passing the blocks, the volume density σ is estimated as the output of the final ResBlockFC. By adding the appearance conditional encoding $\mathbf{c}_\mathbf{a}$ and the positional encoded viewing direction $\gamma(\mathbf{d})$ to the volume density and passing the linear layer, the decoder outputs the feature $\mathbf{f}$. In (b), BL_Up and NN_Up represent a bilinear upsampling and a nearest neighbor upsampling, respectively. The neural renderer takes the feature $\mathbf{f}$ as an input and outputs a synthesized image $\hat{\mathbf{I}}$.

B. Controllable features in 3D object generation

In this section, we report additional 3D-aware generations with controlling the features as well as a negative result as Figures 13 and 14.

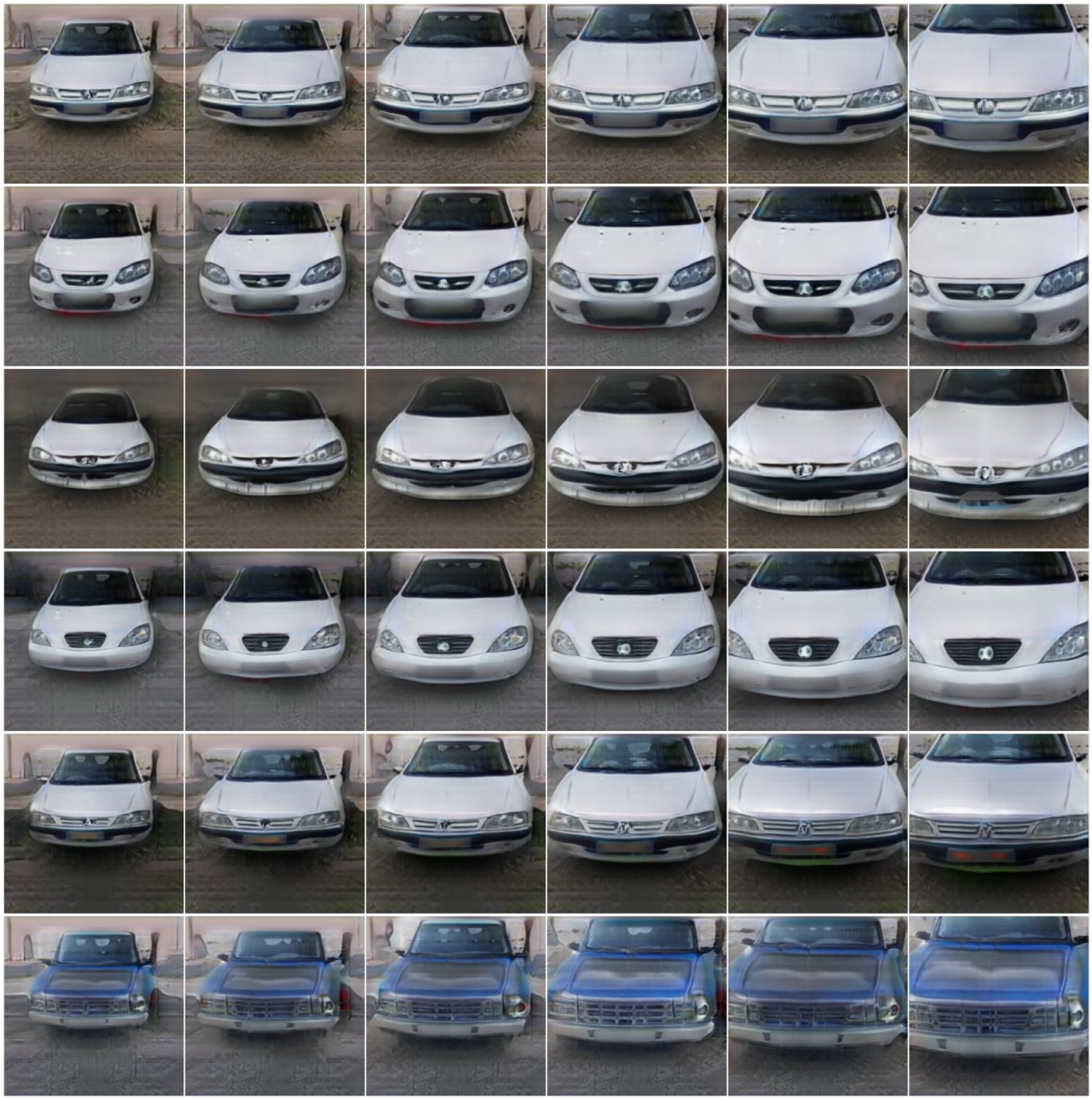


Figure 13. **Visualization of manipulating the scaling.** Each row and column indicate a single object with the same conditional class and the same scale value, respectively. Our model generate a object with various scales, including out-of-distribution images.

Figure 14. **Negative result.** Each column indicates different rotation angles. Near to the edge of columns represents more rotation. While we trained with 60° rotation angle, we produce the results with 120° rotation angle, corresponding to out of distribution. The excessive out of distribution occurs negative results as the edge of column images.

C. Continuously controllable features in 3D object generation

We report additional interpolation and extrapolation results on CelebA with manipulations of different labels as Figures 15 to 20.

Figure 15. **Interpolation and extrapolation results with the glass condition.** Each column indicates generated images with the corresponding label values between zero and three.

Figure 16. **Interpolation and extrapolation results with the aging condition.** Each column indicates generated images with the corresponding label values between zero and three.

Figure 17. **Interpolation and extrapolation results with the make-up condition.** Each column indicates generated images with the corresponding label values between zero and three.

Figure 18. **Interpolation and extrapolation results with the rosy cheek condition.** Each column indicates generated images with the corresponding label values between zero and three.

Figure 19. **Interpolation and extrapolation results with the bald condition.** Each column indicates generated images with the corresponding label values between zero and three.

Figure 20. **Interpolation and extrapolation results with the mustache condition.** Each column indicates generated images with the corresponding label values between zero and three.

Figure 21. **Random Samples on AFHQ with a 256^2 resolution.**

Figure 22. **Random Samples on CelebA with a 128^2 resolution.**

Figure 23. **Random Samples on Cars with a 256^2 resolution.**