CtrlNeRF: The Generative Neural Radiation Fields for the Controllable Synthesis of High-fidelity 3D-Aware Images.

Liu, Jian[a,*], Yu, Zhen[b]

[a]School of Computer and Artificial Intelligence, Zhengzhou University, No.100 Science Avenue, Zhengzhou, Henan, 450001, China
[b]Department of Electrical and Computer Engineering, California State Polytechnic University, Pomona, CA 91768, USA

ARTICLE INFO

Keywords:
Implicit Representation
Novel View Synthesis
Generative Adversarial Network (GAN)
Neural Radiation Field (NERF)
Controllable Image Generation
3D-Aware Images

ABSTRACT

The neural radiance field (NERF) advocates learning the continuous representation of 3D geometry through a multilayer perceptron (MLP). By integrating this into a generative model, the generative neural radiance field (GRAF) is capable of producing images from random noise z without 3D supervision. In practice, the shape and appearance are modeled by z_s and z_a, respectively, to manipulate them separately during inference. However, it is challenging to represent multiple scenes using a solitary MLP and precisely control the generation of 3D geometry in terms of shape and appearance. In this paper, we introduce a controllable generative model (*i.e.* **CtrlNeRF**) that uses a single MLP network to represent multiple scenes with shared weights. Consequently, we manipulated the shape and appearance codes to realize the controllable generation of high-fidelity images with 3D consistency. Moreover, the model enables the synthesis of novel views that do not exist in the training sets via camera pose alteration and feature interpolation. Extensive experiments were conducted to demonstrate its superiority in 3D-aware image generation compared to its counterparts.

1. Introduction

In 2014, Goodfellow et al. proposed a generative adversarial network (GAN) [1], which is a deep generative model inspired by game theory. Subsequently, various GAN-derived models were developed for image generation and translation tasks [2]. A typical GAN comprises a generator and discriminator that compete with each other to attain Nash equilibrium. The purpose of the generator is to produce as much synthetic data as possible that aligns with the potential distribution of real data, whereas the discriminator's aim is to accurately differentiate between genuine and fabricated data. The architecture of the GAN prototype is illustrated in **Fig**.1. The input of the generator is random noise, denoted by z, which is mapped into a new data space using function G(z). The discriminator serves as a binary classifier that differentiates between real samples taken from the dataset and fake samples generated by the generator. During adversarial training, the objective function aims to maximize generator loss and minimize discriminator loss. When the discriminator cannot distinguish between real and fake data, it reaches an optimal state. At this point, the generator successfully learns the distribution of the real data.

Although GANs have achieved significant success in 2D image synthesis, the generated images cannot preserve the 3D consistency. In contrast, the neural radiation field (NERF) [3], which is briefly summarized as the use of an MLP network to learn a 3D geometric representation from a set of posed images, enables the rendering of images from an arbitrary view because it is a continuous 3D presentation of 2D images with camera poses. Due to the inherent features of the radiance fields, rendered images can enforce multiview consistency. Currently, neural radiation fields have been successful in applications of reverse rendering, novel view synthesis, 3D object editing, digital human bodies, and image/video processing. The primary limitations of NERFs are that they require posed images for training and are unable to learn multiple scenes using a single MLP.

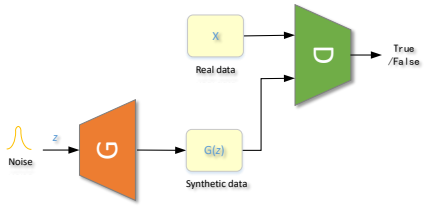

Figure 1: The architecture of generative adversarial networks (GANs). G refers to a generator, and D refers to a binary discriminator.

By integrating a neural radiance field into the generator, a generative radiance field (GRAF) [4] was implemented to produce 3D-aware images from random noise with a Gaussian distribution, and the model was trained on unstructured datasets without 3D supervision. The conditional radiance field in the generator uses 5D coordinates with the spatial location (x, y, z) and viewing direction (θ, ϕ) as inputs, and novel views are synthesized by projecting the output color c and density θ into an image using differential volume rendering. A patch-based discriminator was employed to distinguish between fake and real images. Furthermore, the shape and appearance are modeled by z_s and z_a respectively, to manipulate them separately during the inference. The shape variable z_s and the appearance variable z_a were obtained separately by sampling a Gaussian distribution.

GRAF is capable of disentangling shapes from appearance using shape and appearance codes and taking precise

liujian10@zzu.edu.cn (L. Jian)
ORCID(s): 0000-0003-2981-2128 (L. Jian)

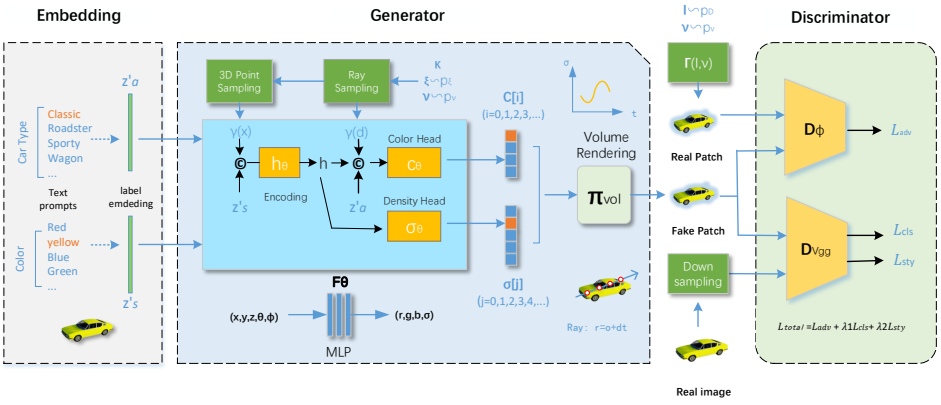

Figure 2: The framework of the generative neural radiation field (CtrlNeRF), which includes three main components: embedding, generator, and discriminator.

control of the camera pose for novel view synthesis, and does not require posed images for training. However, a bottleneck is that one MLP merely represents a scene, resulting in high memory overhead in the case of multiple scenes. Furthermore, GRAF cannot provide sophisticated control over the shape and appearance of generated objects. To address these issues, we introduce **CtrlNeRF**, a generative model based on neural radiance fields that allows precise control of image synthesis according to class labels (shape) and color labels (appearance). The framework is illustrated in **Fig**.2. The generated images preserved 3D consistency because of the intrinsic features of the radiance field in the generator. Moreover, the model allows for novel view synthesis by manipulating the camera pose. Specifically, we make the following contributions.

- We modified the input and output of the MLP and added a VGG-based discriminator to differentiate class and color.
- We represented multiple scenes using a single MLP, reducing storage consumption and increasing inference efficiency.
- We achieved explicit control over the 3D-aware image generation according to the class and color labels.
- We synthesized novel views nonexistent in the dataset through camera pose alteration and feature interpolation.

The remainder of this paper is organized as follows. Section II introduces related work on 2D/3D image generation. Section III briefly reviews backbone models and explains the proposed method. Section IV presents the experimental settings and evaluation metrics. Section V presents a qualitative and quantitative analysis of the results. Finally, Section VI concludes the paper.

2. Related Works

2D Image Synthesis: Generative adversarial networks (GANs) are deep generative models that perform advanced unsupervised tasks [5] such as image generation, image superresolution, and text-to-image synthesis, *etc*. For unconditional GANs, for example DCGAN [6], the input of the generator is random noise, which is an unrestricted input that probably leads to low quality images in some cases, and the formation of an image is uncontrollable due to the randomness of the input. In contrast, conditional GANs, such as InfoGAN [7], CGAN [8], and ACGAN[9], incorporate conditional variables (labels and text) into the generator and discriminator, allowing the generation of high-quality images with control. To stabilize the training process, the Wasserstein generative adversarial networks [10][11][12] used the Earth-Mover (EM) distance to optimize the objective function, producing a better gradient behavior than other distance metrics.

For a typical GAN, obtaining high-resolution images is challenging because the discriminator can easily distinguish between false and true images at high resolution. Several strategies have been implemented to enhance the stability of the training process and progressively improve image resolution [13] [14] [15]. For example, a GAN HD pixel-to-pixel [16] can produce high-resolution images up to 2048 × 2048 pixels. To improve image generation control, several studies [17][18][19][20] have been conducted to disentangle the underlying factors of variation. Two-dimensional images are essentially projections of three-dimensional objects. However, they cannot ensure multiview consistency owing to the absence of 3D geometric constraints.

Implicit Representation: Implicit representations of 3D geometry are popular for deep learning 3D reconstruction [21]. The advantages of voxel-based [22][23] [24][25] or mesh-based methods [26][27][28][29] are that implicit representations are continuous and are not restricted to topology. Recently, hybrid grid representations [30][31] have

been extended to large-scale scenes, but all of the above methods require 3D input without considering texture. To overcome the limitations of 3D supervision, some studies [32][33][34][35] presented differentiable rendering techniques to learn continuous shape and texture representations from 2D-posed images. Mildenhall et al. [3] proposed neural radiance fields, in which they combined an implicit neural model with volume rendering for novel view synthesis. NeRF requires multi-view images with camera poses for supervision and trains a single network per scene. The neural radiance field is implemented by an MLP, which is a fully connected multilayer network with a 5D input of spatial location $\mathbf{x}(x, y, z) \in \mathbb{R}^3$ and viewing direction $\mathbf{d}(\theta, \phi) \in \mathbb{R}^2$, 4D output of volume density $\sigma \in \mathbb{R}^+$ and view-dependent color $\mathbf{c}(r, g, b) \in \mathbb{R}^3$.

3D-Aware Image Generation: To date, neural scene representations have been integrated into generative models to enable the synthesis of 3D-aware images from latent code. Voxel-based GANs [36][37][38] learn textured 3D voxel representations from two-dimensional images using differentiable rendering techniques. However, such voxel-based models are memory intensive, impeding high-resolution image synthesis. Radiance field-based methods [39] achieve higher quality and better 3D consistency, but have difficulties in training high-fidelity images due to the cost of the rendering process.

Mildenhall et al. [3] proposed neural radiance fields (NERF) that can implicitly represent 3D geometries and synthesize novel views using volume rendering. NERF and their variants are valuable tools for generating 3D-aware images. Despite their strengths, they are limited by slow training and inference, inability to handle dynamic scenes, generalization shortcomings, and the necessity for a great number of perspectives. To address these challenges, Garbin et al. [40] introduced FastNeRF, a method that can generate high-quality images at a rate of up to 200 Hz. To apply NERF to unknown scenes, studies on this issue include pixelNeRF [41] and IBRNet[42]. Furthermore, J. Gu et al. [43] proposed styleNeRF to synthesize high-resolution images at interactive rates, allowing control of camera poses and different levels of styles. Huang et al. [44] designed a framework for stylizing 3D scenes through 2D-3D mutual learning.

Taking advantage of both GAN and NeRF, Schwarz *et al.* [4] introduced generative neural radiance fields (GRAF). Although the shape and appearance are disentangled in the model, they are restricted to a single-object scene without explicit control of the image synthesis. Niemeyer *et al.* [45] presented GIRAFFE to learn 3D representation of a compositional scene as synthetic neural feature fields, which employs MLP to represent each object in the scene and reconstruct them afterwards, significantly increasing memory consumption and computational cost. Most recently, several SOTA generative models inspired by GRAF have been introduced. HeadNeRF [46] is a facial rendering method that combines NeRF and facial parameterization models. Its outstanding advantages lie in real-time performance and support for separate control of camera pose, facial identity, expression, and appearance. GRAM [47] is an innovative method designed to control point sampling and learning of radiance fields on 2D manifolds, represented as a collection of implicit surfaces within a 3D volume. Clip-NeRF [48] is a versatile framework that enables intuitive manipulation of NeRF through brief text prompts or exemplar images. It combines NeRF's capability for novel view synthesis with the controllable manipulation of latent representations in generative models.

3. Method

The neural radiance field (NERF) has achieved impressive results in a novel view synthesis using a set of posed images. Combined with the generative model, the generative radiance field (GRAF) has been successfully employed in 3D-aware image synthesis from latent code. The generated images preserve multiview consistency due to the benefits of the neural radiance field. Moreover, the GRAF prototype can be trained using unposed images and provides explicit control over the camera pose. The shapes and appearances in GRAF were disentangled using the shape code z_a and the appearance code z_s. However, shapes and appearances are subject to a certain level of unpredictability because of the randomness of latent codes. Our approach employs a single MLP to learn multiple scenes and achieves precise control over the synthesis of 3D images based on labels. To support the rationale behind our model design, we initially present the fundamentals of NERF and GRAF.

Neural Radiance Fields (NERF): The radiance field is a continuous representation of a scene, denoted by the function F_Θ, which takes the 3D location $\mathbf{x}$, the viewing direction $\mathbf{d}$ as input and the color $\mathbf{c}$ along with the volume density values σ as output. The mapping function F_Θ:($\mathbf{x}$, $\mathbf{d}$) →($\mathbf{c}$, σ) was implemented using a fully connected network that optimizes the weights to map each of the 5D coordinate inputs to their appropriate density and color. Due to the bias of deep networks towards lower frequency functions, the function F_Θ, when applied directly to the 5D coordinate input, proved inadequate to capture high frequency variations. Hence, positional encoding $\gamma()$ is used to translate a 3D location and viewing direction into a high-dimensional space, thus facilitating F_Θ to approach a high-frequency function with greater ease, formally defined in Equation 1.

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

The function $\gamma()$ is applied independently to the three coordinate values (x,y,z) of the position $\mathbf{x}$ and two components of the unit vector of the viewing direction $\mathbf{d}$. The MLP network assigns the resulting characteristics to the color value $c \in \mathbb{R}^3$ and the volume density $\sigma \in \mathbb{R}^+$, as shown in Equation 2. Here, L_x=10 and L_d=4.

$$\begin{aligned}&\gamma(\mathbf{x}), \gamma(\mathbf{d}) \mapsto (\mathbf{c}, \sigma)\\ &\mathbb{R}^{L_x} \times \mathbb{R}^{L_d} \rightarrow \mathbb{R}^3 \times \mathbb{R}^+\end{aligned} \tag{2}$$

The neural radiance field is a representation of a scene as the volume density and emitted radiance at every point in space. The volume density, denoted by σ, can be thought of as the probability differential of a ray terminating at an infinitesimal particle at a specific location $\mathbf{x}$. The expected color $C(r)$ of the camera ray: $r(t) = o + td$ is defined in Equation 3, with near and far bounds t_n and t_f.

$$\begin{aligned} C(r) &= \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d))dt \\ where \quad T(t) &= \exp\left(-\int_{t_n}^{t} \sigma(r(s))ds\right) \end{aligned} \tag{3}$$

where the function $T(t)$ denotes the accumulated transmittance along the ray from t_n to t and the probability that the ray travels from t_n to t without hitting any other particles. Rendering a 2D image from a neural radiance field requires estimating the integral $C(r)$ for each camera ray r traced through the pixels of a virtual camera.

The integral of $C(r)$ is typically estimated using a deterministic quadrature, which inherently restricts the resolution of rendered images because the MLP is merely interrogated at a discrete points. A stratified sampling approach was implemented to divide the data into uniformly spaced intervals, from which a single representative sample was randomly selected from each interval. This method computes the integral value by aggregating data points from a discrete collection of samples. Moreover, a hierarchical volume sampling technique was used to enhance rendering efficiency.

Generative Radiance Fields (GRAF): Generative Radiance Field (GRAF) is a generative model comprising a generator based on the radiance field and a multi-scale patch discriminator. This model enables the synthesis of 3D-aware images from random noise, and is trained on unposed datasets.

(1) **Generator:** The inputs of the generator is made up of the intrinsic camera parameter K, the camera pose ξ, the sampling pattern ν, the shape code z_s, and the appearance code z_a. The generator generates predicted image patches, denoted as $P^{'}$, as its output. The pose of the camera, denoted ξ, is randomly selected from the pose distribution, denoted by p_{ξ}. The center (u, s) and scale of the virtual patches are determined using a uniform distribution. Furthermore, the shape and appearance codes, denoted by z_s and z_a, are drawn from the shape and appearance distributions, denoted by p_a and p_s.

Ray Sampling:The real patch $P(u, s)$ is determined by utilizing the 2D image coordinates that specify the position of each pixel in the image domain. The corresponding rays are determined by these coordinates, the intrinsic camera parameter K, and the camera pose ξ.

3D Point Sampling: The sampling method involves sampling N points $\{\mathbf{x}_r^i\}_{i=1}^N$ along each ray r for the numerical integration of the expected color $C(r)$. Instead of using a single network to represent the scene, stratified sampling optimizes two networks: one 'coarse' and one 'fine' simultaneously. This procedure allocates more samples to the visible region to increase the quality of the images.

Conditional Radiance Field: The conditional radiance field is implemented by a fully connected neural network with parameter θ. More than a regular radiance field, it is subject to the inputs of shape code z_s and appearance z_a. The encoding for shape h is obtained by concatenating the positional encoding $\gamma(x)$ and shape code z_s and is subsequently converted to the volume density σ through a density head σ_θ. Nevertheless, in order to separate the shape and appearance, the volume density was independently predicted without employing the view direction d and appearance code z_a during the inference process. To estimate the predicted color $\mathbf{c}$, a concatenating vector comprising the shape encoding h, positional encoding of the direction $\gamma(d)$, and appearance code z_a is fed into the color head c_θ for further inference.

Volume Rendering: The acquisition of the color $\mathbf{c}$ and volume density σ of N points along the ray (c_r^i, σ_r^i) was achieved using volume rendering. The synthesized patch $P^{'}$ was obtained by combining the result of every sampling ray, and the value of the color c_r was calculated using equation 4.

$$\begin{aligned} c_r = \sum_{i=1}^{N} T_r^i \alpha_r^i c_r^i \qquad T_r^i = \prod_{j=1}^{i-1}(1 - \alpha_r^j) \\ \alpha_r^i = 1 - \exp\left(-\sigma_r^i \delta_r^i\right) \end{aligned} \tag{4}$$

The transmittance (T_r^i) and alpha value (α_r^i) of sample point i along the ray r are denoted by T_r^i and α_r^i, respectively, and the distance between neighboring sample points is defined by $\delta_r^i = \left\| x_r^{i+1} - x_r^i \right\|_2$.

(2) **Discriminator:** The development of a discriminator involves the construction of a deep convolutional neural network (CNN) with ReLU as an activation function. The discriminator accelerates both training and inference by comparing the synthesized patch $P^{'}$ with the real patch P, which is obtained by accessing a real image at 2D coordinates $P(u, s)$ through bilinear interpolation, referred to as $\Gamma(I, \nu)$. The discriminator was adequate for all patches randomly sampled on various scales. The size of the patch determines its receptive field, where larger receptive fields are utilized to capture global content, and smaller receptive fields are used to progressively discern local details.

(3) **Training and inference:** In adversarial training, the generator $G(\theta)$ seeks to minimize the function $V(\theta, \phi)$, while the discriminator $D(\phi)$ seeks to maximize it. The non-saturating objective function $V(\theta, \phi)$ with R1 regularization

is defined in Equation 5.

$$V(\theta,\phi) = \mathbb{E}_{z_s\sim p_s, z_a\sim p_a, \xi\sim p_\xi, v\sim p_v}[f(D_\phi(G_\theta(z_s,z_a,\xi,v)))] + \mathbb{E}_{I\sim p_D, v\sim p_v}[f(-D_\phi(\Gamma(I,v))) - \lambda\|\nabla D_\phi(\Gamma(I,v))\|^2] \quad (5)$$

Where $f(t) = -\log(1+\exp(-t))$, I signifies an image sampled from the data distribution p_D, and p_v refers to the distribution over random patches. Furthermore, the parameter λ controls the level of regularization. The discriminator utilizes both spectral normalization and instance normalization.

CtrlNeRF: Actually, GRAF can generate a 3D geometry from latent code; however, its shape and appearance are not easily manipulated. To address this issue, we developed a GRAF-derived model (i.e. **CtrlNeRF**). This model allows us to use a single MLP to learn multiple 3D representations and to explicitly control object formation. We modified the output of the MLP to disentangle the shape and appearance and added an extra discriminator to distinguish between the object category and style.

(1) **Generator:** Based on the GRAF generator, we manipulated the input of the MLP by embedding label codes in shape and appearance codes, allowing the generator to utilize the label-embedded codes to generate a 3D geometry with precise controls of shape and color.

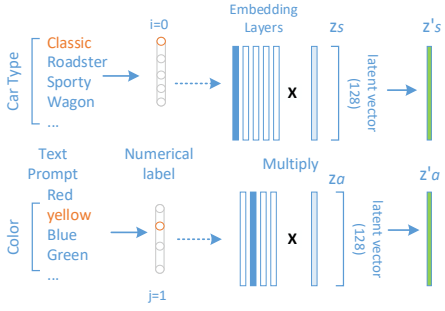

Figure 3: Scheme for incorporating label codes into a latent code through multiplication.

Input: The latent code z is typically separated into two components: shape code z_s and appearance code z_a. The text labels were initially translated into numerical labels (i=0, 1, 2, ...), and the corresponding vector was extracted from the embedding layers using the label index as a reference. The label-embedded codes $z_s^{'}$ and $z_a^{'}$ were derived by multiplying the latent codes z_s and z_a by the feature vectors.
Output: Unlike the GRAF, the output of MLP in the model consists of a volume density array $[\sigma(i)]_{i=0}^{N-1}$ and a color array $[c(j)]_{j=0}^{M-1}$. In this context, M denotes the number of classes, and N denotes the number of styles. The label i/j refers to the numerical representation of the class/color. The

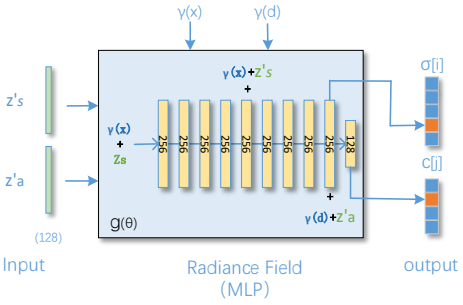

Figure 4: The architecture of a conditional radiance field (MLP) comprises inputs of $z_s^{'}$ and $z_a^{'}$, as well as $\gamma(x)$ and $\gamma(d)$. The output of the model consists of a volume density array $\sigma[]$ and color array c[].

label embedding technique is illustrated in Fig.4.
Conditional Radiance Field: The inference for the volume density and color resembles that of the GRAF prototype. However, the MLP in the generator is conditional on the inputs of the label-embedding latent code, and the outputs are the density and color arrays associated with class and style. The structure of the proposed conditional radiance field is depicted in **Fig**.3. In this design, $\gamma(\mathbf{x})$ and $\gamma(\mathbf{d})$ refer to the positional encoding for the coordinates $\mathbf{x}$ in the 3D space and the directions $\mathbf{d}$ of the rays associated with each point, respectively. $z_s^{'}$ and $z_a^{'}$ are the label-embedded latent codes.

(2) **Discriminator**: In addition to using the typical discriminator in GRAF to evaluate the generated patch $P^{'}$ compared to the real patch P, we employed a discriminator based on VGG16 [49] to effectively classify various classes and styles of objects. The VGG network is well known for its exceptional performance in multi-classification tasks. Initially, the discriminator D_{vgg} was trained using annotated images $I^{'}$ that were down-scaled from the real image I. The pre-trained discriminator was utilized as an auxiliary classifier for the patches generated $P^{'}$. To further improve image quality, we adopted posed images for training and replaced adversarial loss with reconstruction loss.

(3) **Training and Inference**: During supervised learning, the network parameters are optimized using loss functions. In particular, the discriminator was trained by a real patch P and a generated patch $P^{'}$ to improve computational efficiency. The discriminator D_{vgg} was trained in real images resized $I^{'}$. The loss function to update the weights of the network is defined in Equation 6, and the pseudocode for training is shown in **Algorithm** 1.

$$L(G(\theta)) = L_{adv}(D(\phi)|P_{i,j}^{'}) + \lambda_1 L_{cls}(D_{vgg}|P_{i,j}^{'}) + \lambda_2 L_{sty}(D_{vgg}|P_{i,j}^{'}) \quad (6)$$

where, L_{adv} refers to adversarial loss between P and $P^{'}$, L_{cls} and L_{sty} refer to the loss of class and style, respectively,

and λ_1 and λ_2 denotes weights for L_{cls} and L_{col}. In the experiment, RMSprop was used as optimizer and the weights of these losses were λ_1=2.0, λ_2=3.0, with a batch size of 8.

Algorithm 1 CtrlNeRF training algorithm

Input: real images I with labels $(\hat{i}, \hat{j})$.
Initialization: camera intrinsic K, camera pose ξ, and sampling pattern ν.
do *iterations*
  $random\ (i,j) \in (\hat{i}, \hat{j})$
  $z_s^{'}, z_a^{'} \leftarrow z_s, z_a$
  $P_{i,j}^{'} \leftarrow G(z_s^{'}, z_a^{'})$:
    for *M points along N rays* :
    $\sigma[i], \mathbf{c}[j] \leftarrow F_\Theta(\mathbf{x}, \mathbf{d})$)
    $P_{i,j}^{'} \leftarrow \pi(\sigma[i], \mathbf{c}[j])$
    end for
  $P_{\hat{i},\hat{j}} \leftarrow \mathcal{T}(I)$
  $L_{adv} \leftarrow D_\phi(P_{i,j}^{'}, P_{\hat{i},\hat{j}})$
  $L_{cls}, L_{sty} \leftarrow D_{vgg}(P_{i,j}^{'}, I)$
  $Loss = L_{adv} + \lambda_1 L_{cls} + \lambda_2 L_{sty}$
  $update\ G$
  $update\ D_\phi, D_{vgg}$
end do

4. Experiments

4.1. Datasets

In this study, due to the lack of annotations in the datasets used for generative models such as GRAF and GIRAFFE, we began by developing a synthetic dataset named **CARs** (I) utilizing 3D editing software. First, a car was situated at the origin of the coordinate system, and a virtual camera was placed on the surface of the upper hemisphere oriented towards the origin. where θ and ϕ represent the pitch and yaw angles of the camera, respectively. The camera was placed in a hemisphere with a radius of r. By manipulating the pose of the virtual camera, we could obtain the views of an object with variable respect. The captured images with a size of 800x800 were automatically labeled by class, color, and pose. Four types of cars (classic, roadster, sporty, and wagon) were included in the CARs dataset, each of which was presented in four color modes: red, green, blue, and yellow. In addition, we used publicly accessible NERF datasets [3], specifically **Synthetic** (II) and **LLFF** (III), for demonstration.

4.2. Baselines

To demonstrate its excellence in 3D-aware image generation, we compared our model with the latest NeRF-based generative models, including CLIP-NeRF. NeRF can learn the continuous representation of 3D geometry from posed images using neural radiance fields and render a novel view using differentiable volumetric rendering. GRAF [4] is a generative model capable of producing images with 3D consistency using latent codes related to shape and appearance, without requiring 3D supervision. GRIFFEE [45] is another generative model derived from the GRAF model, which uses multiple MLPs to represent compositional scenes. CLIP-NeRF is the SOTA multimodal 3D object manipulation method for neural radiance fields using a short text prompt. For these generative models, qualitative and quantitative analyzes were performed to evaluate the performance of the proposed model by comparing it with other generative models.

4.3. Evaluation Metrics

The FID score, which comprises human assessments of realism and diversity, has been widely used to evaluate the quality and variety of the generated images. This metric was first introduced by Kanazawa et al. in 2018 [50]. The lower the FID score, the better the model performance. FID score was derived from Equation 7.

$$\begin{aligned} \mathcal{FID} &= d^2((m_r, C_r), (m_g, C_g)) \\ &= ||m_r - m_g||_2^2 + \mathcal{T}r(C_r + C_g - 2(C_r C_g)^{1/2}) \end{aligned} \tag{7}$$

The pair (m_r, C_r) corresponds to real images, while the pair (m_g, C_g) corresponds to generated images. In each pair, m represents the mean and C represents the covariance. The KID score [51], which is an unbiased estimate that does not require a normal distribution hypothesis, was introduced for image evaluation.

In addition, to measure the reconstructed shapes and their closest shapes in the ground truth, the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) [52] were used for a quantitative comparison between real and synthetic images.

5. Results

5.1. Controllable 3D-aware Image Synthesis based on the Labels.

The model was trained on the **CARs**(I), **Synthetic** (II), and **LLFF**(III) datasets. Label-associated latent codes $z_s^{'}$ and $z_a^{'}$ are the input of the MLP in the generator, and the view direction $d(\theta, \phi)$ is sampled within a specific pose range. In the experiment, we sampled 1024 rays for an image and 64 points along each ray. Therefore, 1024×64 points were sampled, each with 3D coordinates and ray directions. The RMSprop optimizer was used, with a learning rate of 0.0001 for the discriminator and 0.0005 for the generator, and class and color labels were used for the prediction. The synthesized images are shown in **Fig**.5, 6, 7 for the three datasets. Optimization for multiple scenes typically requires approximately 100–200k iterations to converge on a single NVIDIA A4500 GPU(approximately 14 h).

The FID score reflects the similarity and variance between real and synthesized images, which is an essential metric to quantitatively evaluate the performance of the generative model. The FID scores of the images generated in dataset I, grouped by class and color, are shown in **Fig**.8. When a single MLP handles multiple scenes within the

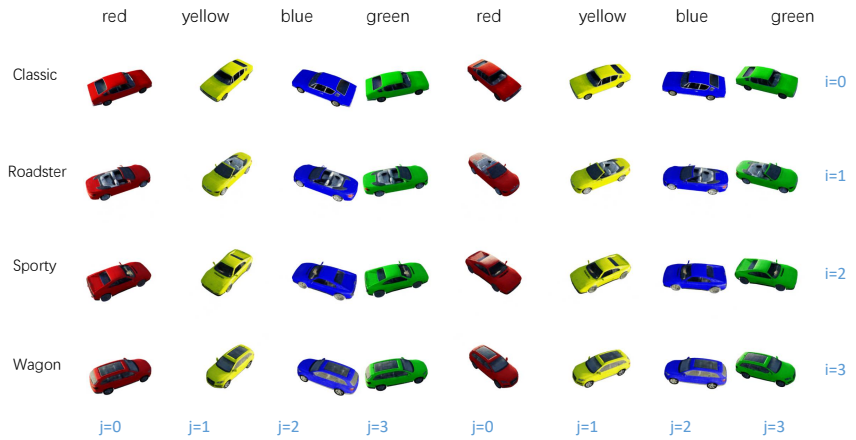

Figure 5: Samples of the synthesized images (400x400) on **CARs**(I) dataset.

Figure 6: Samples of the synthesized images (400x400) on **Synthetic**(II) dataset.

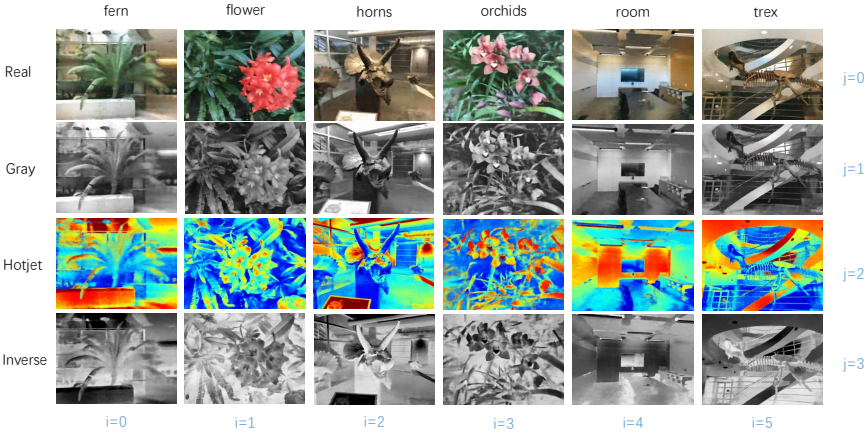

Figure 7: Samples of the synthesized images (504x378) on **LLFF**(III) dataset.

model, it is evident that the image quality decreases as the number of scenes increases. The mean FID scores of the model trained on Datasets (I), (II), and (III) are presented in **Fig.**9. As shown in **Fig.** 10. Image quality decreased with an increase in the number of classes and styles.

Furthermore, the model performance on the "LLFF" dataset was noticeably poorer compared to the "Car" dataset, likely due to the higher complexity of the images in the "LLFF" dataset, particularly when dealing with multiple scenes. Higher complexity implies greater entanglement

when using a shared-weight MLP to represent multiple scenes.

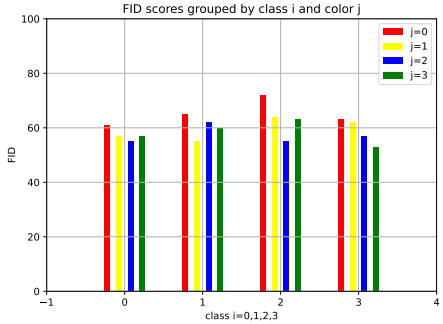

Figure 8: The diagram of FID scores of the generated images grouped by class (i=0,1,2,3), and color (j=0,1,2,3)

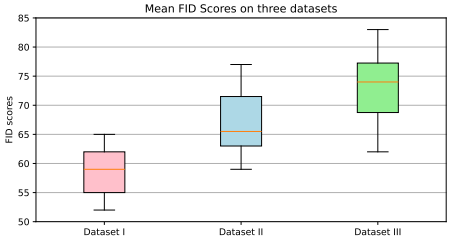

Figure 9: The mean FID scores of the generated images on CARs(I), Synthetic(II) and LLFF(III) datasets, respectively

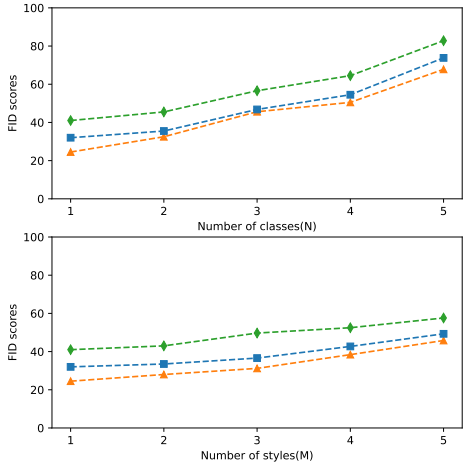

Figure 10: The diagrams of FID scores of the generated images with increasing the number of classes and styles on CARs(orange), Synthetic(blue) and LLFF(green) datasets.

5.2. Novel View Generation via Camera Manipulation.

As shown in the following three figures, novel views of an object can be obtained by alternating the poses of the rendering camera. For example, in **Fig.**11, the virtual camera captures images of the object with the poses of $\theta \in [-180°, 180°]$ and $\phi \in [0°, 90°]$. In **Fig.**12, we changed the radius of the rendering sphere stepwise, ranging from 3.5 to 5.0, with an interval of 0.5. Finally, we performed horizontal translation of the synthesized objects within the range of (-1.0, 1.0) in **Fig.**13.

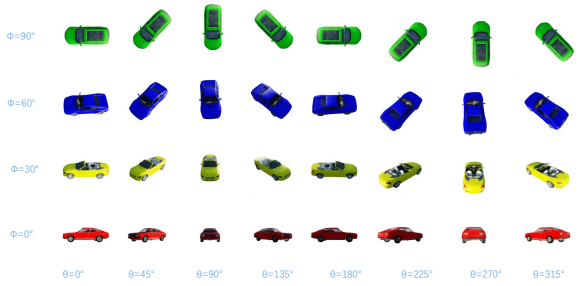

Figure 11: Novel views synthesized by manipulating the rendering camera. θ, ϕ denotes the pitch or yaw angle, respectively.

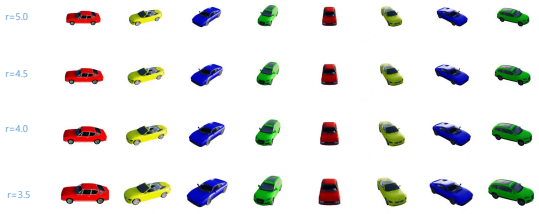

Figure 12: The depth translation of the synthetic object with the radius r from 3.5 to 5.0. Here, r is the distance between the origin of coordination and the camera.

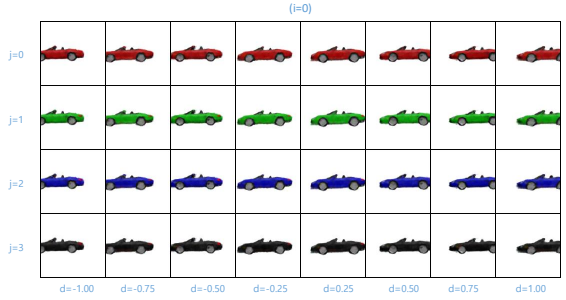

Figure 13: The horizontal translation of the synthetic object with the distance d from -1.0 to 1.0. Here, d is the horizontal shift of the camera.

5.3. New Feature Synthesis via Linear Interpolation.

As shown in **Fig.**14, the new color of the car, which is unseen in the training set, is synthesized using the color interpolation: c=(1.0-λ)c[i]+λc[j]. where λ is a linear coefficient ranging from 0 to 1. In the same way, we can

Figure 14: The color of the car is synthesized via color linear interpolation. λ is a linear coefficient that varies from 0 to 1.

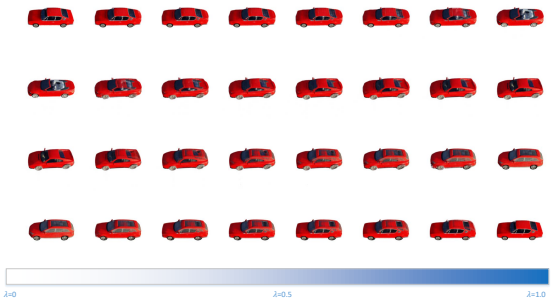

Figure 15: The shape of the car is altered via density linear interpolation. λ is a linear coefficient ranging from 0 to 1.

also simulate other features, such as texture, material, and environmental illumination. As shown in **Fig.**15, the shape of the car can also be altered step by step through density interpolation.

5.4. Ablation Studies

The development of the proposed model entailed adaptation of the GRAF prototype by altering the input and output components of the MLP and integrating an additional discriminator. Consequently, in our ablation studies, we compared the results of our model with those of Models I, II, and III, which eliminated the specific modifications for the input, output, and VGG discriminators. In Table 1, we present a quantitative comparison of the FID and KID scores of Models I, II, and III with those of our model, indicating that the manipulation of the MLP output in the GRAF prototype plays a significant role in our model because training does not converge without it. Using density and color arrays, we effectively deployed the output to multiple slots corresponding to classes and styles. Then these outputs were used to render the images independently. We also observed that the image quality degraded without embedding labels for the input and the VGG discriminator. The two strategies not only increased the quality of the generated images, but also shortened the training time.

5.5. Comparison to SOTA Methods

Generative radiance fields combine GAN and NERF techniques to synthesize 3D-aware images. In both qualitative (**Fig**. 16) and quantitative (**Tab**. 2) comparisons with state-of-the-art generative methods, our approach yields results on par with the CLIP-NERF method and exceeds the GRAF and GIRAFFE methods in terms of PSNR and SSIM. During the experiment, we noticed that the generative models facilitate creating new views with 3D consistency using the unposed dataset, but our method can store multiple scenes in a single MLP without significantly sacrificing image quality and also allowing for manipulation of 3D-aware image creation based on the given labels and camera pose. Although GRAF and GIRAFFE are both generative models, GRAF is unable to represent multiple scenes within a single MLP. On the other hand, GIRAFFE employs MLPs to represent each object in a composite scene, leading to substantial memory consumption in multiobject scenes. As anticipated, our model falls short of CLIP-NERF with respect to PSNR and SSIM due to the use of a single MLP for implicit representation of multiple scenes simultaneously. Furthermore, to highlight the advantages of our model over CLIP-NERF, we performed a quantitative analysis in **Tab**. 3, concentrating on storage requirements and computational costs. CLIP-NERF allows for the manipulation of the shape and color of objects according to text or image prompts, but it cannot learn multiple scene representations in a single model and requires separate training for each scene. As the number of scenes to be represented grows, both the model storage and the training time expand proportionally. In contrast, the demands of our model remain unchanged.

6. Conclusion

The study aimed to achieve a sophisticated level of control in 3D-aware image synthesis. To achieve this goal, we improved the GRAF to allow for precise manipulation of 3D object creation in terms of pose, class, color style, and other attributes. By modifying the input and output of the MLP as well as the incorporation of an additional discriminator, we successfully entangled and disentangled the label codes into and out of the latent code, thereby enabling 3D-aware image generation from label prompts during the inference phase. Using our model, various scenes can be implicitly represented using a single MLP with shared weights, which significantly minimizes memory usage when handling multiple scenes. Additionally, it has been demonstrated that while the image quality produced by our model surpasses that of the NeRF-based generative models, it is marginally less impressive than CLIP-NERF, which is attributed to the shared weights within the MLP. Another limitation of the model is that the image quality diminishes as the quantity and complexity of the scenes increase.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 2014.

[2] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. Recent progress on generative adversarial networks (gans): A survey. *IEEE Access*, pages 36322–36333, 2019. DOI:10.1109/ACCESS.2019.2905015.

Table 1
The results of FID/KID scores in the ablation studies.

Model/Class FID/KID	Classic	Sporty	Roadster	Wagon	Mean
Model I	68.83	71.07	80.18	74.92	**73.75**
(w/o input)	0.072	0.083	0.078	0.075	**0.077**
Model II	–	–	–	–	no
(w/o output)	–	–	–	–	converge
Model III	62.13	55.25	58.14	63.96	**56.87**
(w/o VGG)	0.065	0.049	0.060	0.068	**0.061**
Ours	42.54	48.06	51.43	44.37	**46.60** ↓
	0.048	0.052	0.057	0.045	**0.050** ↓

Figure 16: Some synthesized images by GRAF, GIRAFFE, CLIP-NERF and our model on the **Synthetic**(II) dataset.

[3] 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. 2020. DOI:10.48550/arXiv.2003.08934.

[4] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *arXiv e-prints*, 2020. DOI:10.48550/arXiv.2007.02442.

[5] Xian Wu, Kun Xu, and Peter Hall. A survey of image synthesis and editing with generative adversarial networks. *Tsinghua Science Technology*, 22(006):660–674, 2017. DOI:10.23919/TST.2017.8195348.

[6] Yang Yu, Zhiqiang Gong, Ping Zhong, and Jiaxin Shan. Unsupervised representation learning with deep convolutional neural network for remote sensing images. In *International Conference on Image Graphics*, 2017.

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. 2016. DOI:10.48550/arXiv.1606.03657.

[8] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *Computer Science*, pages 2672–2680, 2014. DOI:10.48550/arXiv.1411.1784.

[9] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. 2016. DOI:10.48550/arXiv.1610.09585.

[10] Martin Arjovsky, Soumith Chintala, Lxe, and On Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 2017. DOI:10.1088/1742-6596/2586/1/012157.

[11] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. 2017. DOI:10.48550/arXiv.1704.00028.

Table 2
The quantitative assessment of our model with state-of-the-art methods in terms of PSNR and SSIM.

Model	GRAF[4]		GIRAFFE[45]		CLIP-NERF[48]		OURS	
/Class	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Hot dog	22.17	0.980	22.67	0.981	34.38	0.998	32.21	0.997
Chair	23.76	0.983	21.85	0.976	32.63	0.997	30.76	0.996
Lego	24.53	0.986	23.16	0.985	30.45	0.996	28.36	0.995
Mic	21.48	0.976	22.64	0.980	31.38	0.996	29.57	0.995
Ficus	23.50	0.985	21.58	0.972	32.86	0.997	31.85	0.996
Mean	**23.08**	**0.983**	**22.38**	**0.979**	**32.34**	**0.997**	**30.55**	**0.996**

Table 3
The comparison of our model with CLIP-NERF [48] in storage demands and computational expenses.

Model	CLIP-NERF				OURS			
/Num. of Scenes (n)	n=1	n=2	n=3	n=4	n=1	n=2	n=3	n=4
Model Storage (MB)	13.6	27.2	40.8	54.4	14.1	14.1	14.1	14.1
Training time (hours)	32.6	65.8	98.5	131.2	8.3	8.1	8.4	8.5
Inference time (seconds)	7.75	7.58	7.61	7.63	7.63	7.78	7.68	7.71

[12] Henning Petzka, Asja Fischer, and Denis Lukovnicov. On the regularization of wasserstein gans. 2017 .DOI:10.48550/arXiv.1709.08894.

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. 2017. DOI:10.48550/arXiv.1710.10196.

[14] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *IEEE*, 2017. DOI:10.1109/ICCV.2017.629.

[15] Zhang Han, Xu Tao, Li Hongsheng, Zhang Shaoting, Wang Xiaogang, Huang Xiaolei, and Dimitris N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017. DOI:10.1109/TPAMI.2018.2856256.

[16] Ting Chun Wang, Ming Yu Liu, Jun Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. 2017. DOI:10.48550/arXiv.1711.11585.

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. DOI:10.1109/TPAMI.2020.2970919.

[18] Wonkwang Lee, Donggyun Kim, Seunghoon Hong, and Honglak Lee. High-fidelity synthesis with disentangled representation. 2020. DOI:10.1007/978-3-030-58574-7_10.

[19] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debhath, and Anima Anandkumar. Semi-supervised stylegan for disentanglement learning. 2020. DOI:10.48550/arXiv.2003.03461.

[20] S. Reed, K. Sohn, Y. Zhang, and H. Lee. Learning to disentangle factors of variation with manifold interaction. In *International Conference on Machine Learning*, 2014.

[21] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, and Stephen Lombardi. Advances in neural rendering. *arXiv e-prints*, 2021. DOI:10.48550/arXiv.2111.05849.

[22] Andrew Brock, Theodore Lim, J. M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *Computer Science*, 2016.

[23] Danilo Jimenez Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images, 2016.

[24] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017 .DOI:10.1109/CVPR.2017.701.

[25] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. 2016. DOI:10.48550/arXiv.1610.07584.

[26] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Atlasnet: A papier-mché approach to learning 3d surface generation. 2018. DOI:10.48550/arXiv.1802.05384.

[27] Yiyi Liao, Simon Donné, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. DOI:10.1109/CVPR.2018.00308.

[28] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2020. DOI:10.1109/ICCV.2019.01006.

[29] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. *In Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. DOI:10.1007/978-3-030-01252-6_4.

[30] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition(CVPR)*, 2020. DOI:10.1109/CVPR42600.2020.00604.

[31] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *In Proc. of the European Conf. on Computer Vision (ECCV)*, 2020 .DOI:10.1007/978-3-030-58580-8_31.

[32] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition(CVPR)*, 2019. DOI:10.1109/CVPR42600.2020.00209.

[33] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. DOI:10.1109/CVPR42600.2020.00356.

[34] Vincent Sitzmann, Michael Zollhfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. 2019. DOI:10.48550/arXiv.1906.01618.

[35] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2020. DOI:10.1109/ICCV.2019.00278.

[36] Philipp Henzler, Niloy Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. DOI:10.1109/ICCV.2019.01008.

[37] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. *IEEE*, 2019. DOI:10.1109/ICCVW.2019.00255.

[38] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong Liang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. 2020. DOI:10.48550/arXiv.2002.08988.

[39] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. DOI:10.1109/CVPR46437.2021.00574.

[40] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. 2021. DOI:10.48550/arXiv.2103.10380.

[41] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. 2020. DOI:10.48550/arXiv.2012.02190.

[42] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. 2021. DOI:10.1109/CVPR46437.2021.00466.

[43] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. 2021. DOI:10.48550/arXiv.2110.08985.

[44] Yu-Jie Yuan Yu-Kun Lai Yi-Hua Huang, Yue He and Lin Gao. Stylizednerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[45] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. DOI:10.1109/CVPR46437.2021.01129.

[46] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022 DOI:10.48550/arXiv.2112.05637.

[47] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022 DOI:10.48550/arXiv.2112.08867.

[48] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. *arXiv e-prints*, 2021 DOI:10.48550/arXiv.2112.05139.

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. DOI:10.48550/arXiv.1409.1556.

[50] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. 2018. DOI:10.48550/arXiv.1806.07755.

[51] Mikoaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. 2018. DOI:10.48550/arXiv.1801.01401.

[52] Z. Wang, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux. Image quality assessment: From error visibility to structural similarity. 2013.