Tri-plane Diffusion Models

Zhaoning Wang Jeffrey Chan Mubarak Shah
University of Central Florida
{zhaoning, jeffrey}@ucf.edu

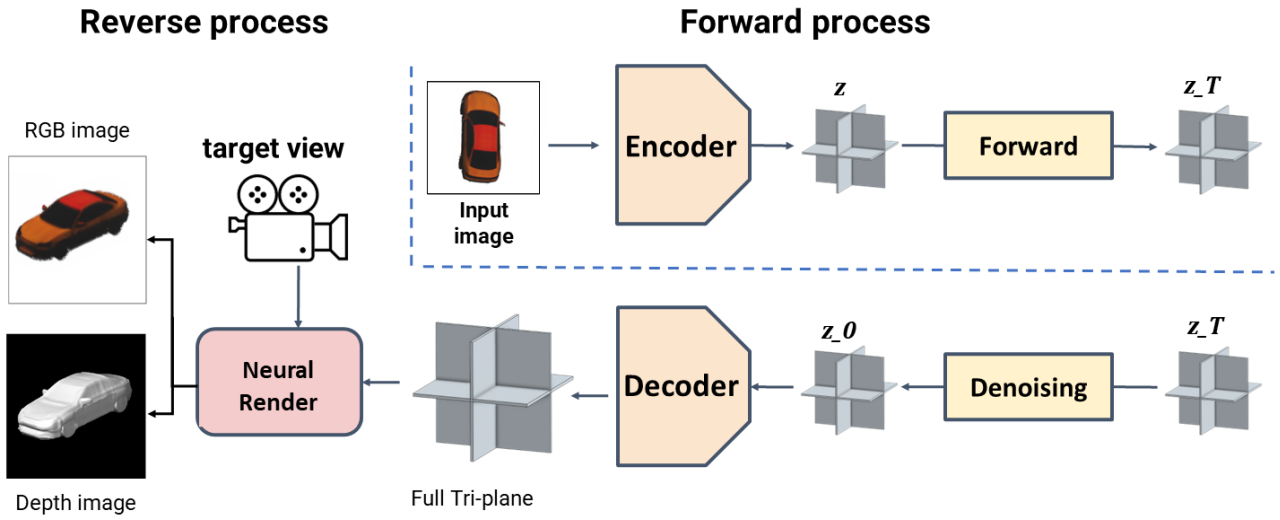

Figure 1. Overview of our model: We trained a VAE that converts an image into a 3D tri-plane representation, with a downscaled small tri-plane as a latent intermediate, which can be coupled with a diffusion model. In the reverse process, we diffusion the latent and decode back to the full tri-plane, which can be neural rendered to new views or depth models.

Abstract

In recent years, 3D computer vision has made tremendous progress with the emergence of Neural Radiance Fields (NeRF), enabling free-viewpoint rendering with implicit representations. However, the generation and editing of a NeRF scene is a trivial task. Existing work in creating 3D representation either requires large computing resources or training data. In this work, we take advantage of the generation and editing capability of diffusion models and use it with an efficient tri-plane 3D representation of NeRF. We propose tri-plane diffusion model, which diffuses on the tri-plane representations to create novel, high-quality 3D scenes with minimal computational overhead. Our approach circumvents the limitations of previous methods by leveraging the inherent flexibility and efficiency of latent diffusion models for seamless generation and editing of NeRF scenes. Through our experiments and evaluations, we propose a new 3D-consistent training scheme and demonstrate that our model can learn 3D tri-plane representations from a dataset of 2D images, and can be coupled with diffusion models for arbitrary generation and editing, paving the way for more accessible and interactive 3D content creation and manipulation. Unlike 2D diffusion models, our method gives a full scene, facilitating efficient unrestricted view synthesis and shape creation. Moreover, our diffusion-centric technique inherently allows for conditional generations like masked completion or single-view 3D synthesis during the inference process.

1. Introduction

Diffusion models have shown extraordinary performance in generating 2D images [50, 51]. The next logical step is the synthesis of 3D scenes. 3D Scenes are essential for developing embodied AI training environments, generating

virtual worlds for virtual reality applications, and creating immersive experiences in gaming and other applications. Making just one 3D asset requires a lot of human effort and is time-consuming. 3D scene synthesis can speed up the process significantly, reducing human labor, time, and cost. In this work, we study the task of 3D scene synthesis using diffusion models.

However, moving from 2D to 3D is not straightforward. First, 3D assets storage complexity increases exponentially. Second, processing 3D examples requires more complex models, which are hard to train. Finally, even if we assume unlimited storage and computing resources, another problem is the availability of large-scale 3D datasets, which are crucial for achieving good performance. Scaling a 3D dataset to the scale of ImageNet will be an enormous task for the computer vision community.

A 3D scene can be represented with point clouds, meshes, Voxels, and Neural Radiance Fields (NeRF). Point clouds, meshes, and voxels are storage-intensive and hard to scale. On the other hand, with NeRF, a 3D scene can be compressed in the weights of a neural network, reducing the storage considerably. However, in order to store a scene, a neural network needs to be trained per scene, which is compute-intensive even for one scene. Recently, the community has been moving to use a triplane representation that effectively balances both storage and computing [1, 6, 12, 17, 20].

To bypass the dependency of large-scale 3D dataset availability, the community has built methods that leverage large-scale 2D datasets to train a model for a 3D task [1, 6, 12, 53]. Generative models like generative adversarial networks(GANs) have been extended from the 2D domain to 3D- aware neural field generation [6, 53]. One example is Eg3D [6], where they trained a StyleGAN to produce images conditionate to pose. They showed that even though their model was trained on 2D images, their model could capture 3D information and construct a 3D scene.

This work proposes a diffusion model which generates 3D scene triplane representations efficiently with high quality. We also designed a training Loss that enables the model to pick up 3D information from 2D images. Our method can increase the diversity of examples on 3D datasets by learning priors and generating new 3D object scenes.

2. Related Work

Generative Models There have been many efforts in machine learning to generate high-quality, photorealistic images, and there is a great number of works have been proposed in recent years. Notable among which are the generative adversarial networks (GANs) [18, 25, 26, 28, 54], VAE [30, 60], and auto-regressive models [59]. Particularly, GANs have been utilized extensively due to the high quality of the generated image, and have a wide range of applications including attribute editing [32, 34, 46, 56], data augmentations [31, 37], and Style transfer [45, 46, 67].

More recently, the success of diffusion models has led to a number of techniques for more realistic generation and demonstrated the wide distribution they can generate [14, 16, 24, 43, 48, 52, 57]. The Denoising Diffusion Probabilistic Models (DDPM) [23] is the most widely used diffusion model; however, it suffers from expensive computing resources and time. Thus, authors in [50] have migrated the diffusion into latent space and made them more efficient while preserving the generation quality. As the computational overhead is even more in the 3D regime, where our focus lands, efficiency is pivotal to our research. Thus, our work is based mainly on [50] to take advantage of performance.

3D representation There is an increasing trend in computer vision recently to use neural fields as 3D representations of scenes [2, 3, 40, 42, 49, 64], using implicit representations. In recent years, there are endless efforts on improving quality and performance. For example, PixelNeRF [64] significantly decreases the number of images to fit a NeRF model by using a convolutional prior. In a more recent work, instant-ngp [42] uses a hash-table approach to reduce the fit time and capability of NeRF.

On the other hand, more explicit neural fields [6, 21, 63] consider the tri-plane to be the preferred type of neural field representation. [21] maps the neural fields into point cloud for meshed object generation. Multiple tri-planes can be utilized simultaneously for 4D rendering [4] to create novel views of an object moving through space and time. Tri-planes can use planar factorization [4] for volumetric rendering that allows 2D, 3D static, 3D dynamic, and 4D dynamic rendering. A tri-plane can also be used with a 3D-aware CDM [20] to produce high-quality images. A 3D-aware GAN from [6] and tri-planes have been used in a 3D-aware video editing system for facial images [63]. A conditional NeRF-based decoder can reconstruct image latents into a tri-plane representation [53] then frozen and used with an autoregressive transformer to generate novel views. A 2D CDM and optimized radiance field [39] can reconstruct 3D objects from a single 2D image, and [1] provides an example of incorporating a tri-plane NeRF directly into the function of a diffusion model. Nonetheless, directly applying the tri-planes on DDPM is time-consuming and hardly has any control over the content.

3D Generation With the recent development of generative models, there has been exponential progress in the computer vision community on 3D content generation. There are great works utilizing GANs [5, 6, 19, 53, 55, 66]. Nonetheless, with the more powerful diffusion models, more possibilities have been explored, such as 3D object generation [1, 9, 21, 41], text-to-object [33, 36, 47], scene

generation [4, 10, 29, 58], novel view synthesis [7, 11, 20, 36, 39, 61], scene editing [22], and Radiance Field generation [15].

The authors in [62] formulate 3D awareness as a multi-view 2D image generation task, but their method does not explicitly determine the underlying 3D structure. An architecture combining a Point Cloud variational autoencoder (VAE), a 3D aware U-Net, diffusion model, tri-plane features, and MLP decoder [21] explicitly determines 3D structure. Some models [33, 47] use 2D text-to-image diffusion to perform text-to-3D synthesis. Similarly, a conditional diffusion model (CDM) as used in [39] takes advantage of previous work [47] in text-to-image diffusion to build scene data for a radiance field using text prompt diffusion constraints to produce plausible novel views. These image-augmented text prompts can substitute for the alternate views normally coming from an underlying data set. The technique in [12] uses a 2D label map to explicitly determine 3D structure and generate different viewpoints, but does not control camera position of the rendered output.

For 3D scenes from 2D images far from the target view, a tri-plane based Neural radiance field (NeRF) and CDM can be combined [20] to mitigate occluded views and produce detailed 3D objects. A diffusion model can be directly integrated with an encoder, tri-plane latent, and MLP [1] to become a 3D-aware denoiser, using intermediate denoising steps to create inductive bias toward 3D scene consistency. Input views can be reverse projected from 2D image features into a 3D volumetric space [7] from which projection back to 2D space and camera parameters can create viewing position specific novel views. A diffusion model can be trained on optimized neural field MLP weights under the guidance of a transformer-based network [15] to enable diffusion modeling over 3D shapes and 4D mesh animations. Novel views of human faces [63] through video editing using 3D-aware methods from [6] decompose and reconstruct facial images for novel views. In [4], 3D scene data is stored for reference by an MLP, decoupling data storage capacity from speed. This method uses a spatial and temporal structure composed of six NeRF feature planes to compute a feature vector for a 4D point in spacetime to enable novel view synthesis of objects in motion, but similar to [12, 20, 39, 63] offers no explicit camera control. Volumetric radiance fields can be generated directly [41] by using a voxel grid representation of an object and a denoising formulation allowing a diffusion model to learn from a data set of 3D objects represented as radiance fields. However, they are resource-demanding and not efficient.

3. Preliminary

3.1. Diffusion Model

Diffusion models are generative models inspired by non-equilibrium thermodynamics. They are usually composed

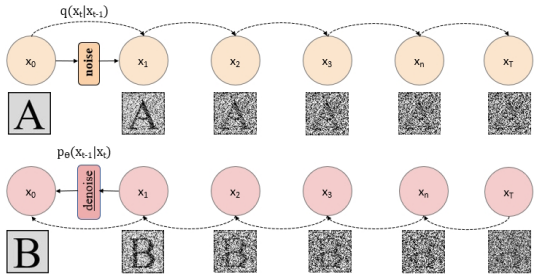

Figure 2. The forward process of diffusion is represented by Eq. 1 and its reverse process is represented by Eq. 2, which may be represented simply by *q* and *p*.

of two main components: the forward process (also known as the diffusion process) and the reverse process (referred to as the reverse diffusion process). In the forward process, data (typically an image) is gradually introduced to noise, while the reverse process involves converting the noise back into a sample that originates from the target distribution.

During the forward process, Gaussian noise is added to the image iteratively and incrementally via a Markov chain until the input image becomes nothing but Gaussian noise and all information from the input image is lost. At this point, the input image x_0 has been mapped into x_T, the final image containing nothing but Gaussian noise. The forward process can be parameterized with β_t as the variance schedule:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \quad (1)$$

On the other hand, the reverse process is to learn to turn the fully noised image x_T back into an image x_0 and thereby learn to start with *any* noised sample x_T plus a seed to then generate an x_0 which is representative of the data set used in training:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \quad (2)$$

The forward and reverse diffusion process is depicted in Fig. 2.

Latent Diffusion Models. In Latent Diffusion Models (LDM) [50], the diffusion model maps the input image into a latent space with a pre-trained VAE. The latent space has less data overhead in the diffusion process. compared to the DDPM. The latent space is usually in height of H/f and width of W/f, where the H and W are the original height and width of the input image. During inference, the VAE decoder is used to decode the diffused latent to desired images. The authors in [50] demonstrate that the latent diffusion models have the capability of generating high-resolution images while avoiding excessive compute demands.

3.2. Neural Radiance Field (NeRF)

A Neural Radiance Field [40] is an implicit, volumetric representation of a 3D object scene given as a density field σ and RGB color field ξ defined over a 3D space. As discussed in [40], the 3D scene is calculated as a 5D function taking the coordinates (x,y,z) of a point and the direction (θ,ϕ) of a ray of light passing through that point and outputting a vector of color and density (c, σ) which tells us what we need to know about light, shadow, and transparency of a pixel at a location; thus, allowing the generation of a 2D view from a specific angle and viewport.

To render a view from a specific camera pose (denoted by camera parameters p), we can consider each pixel on the rendered image to be a ray shooting into the scene. With ray casting logic, one can use volume rendering [38] to render the RGB color $C(\mathbf{r})$ along the given ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ in the radiance field following the equation below:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \tag{3}$$

where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$. and t_n and t_f are the near and far bounds of rendering. The overall flow is shown below on Fig. 3

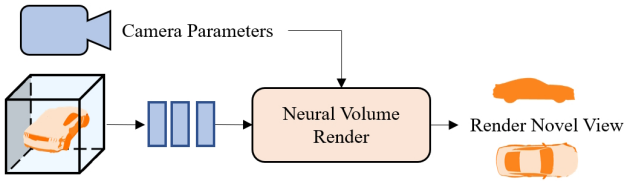

Figure 3. Overall NeRF rendering flow.

For the purposes of this paper, NeRF provides evidence that a 3D representation can be built up from 2D images via some sort of ray projection and volumetric rendering. We can also see this volumetric rendering and pixel calculation in a different form in the method of tri-plane features.

3.3. Tri-Plane Features

Although NeRF has an amazing ability to represent a 3D scene, it has a series of drawbacks, one of which is that it relies on *implicit* representations, i.e. MLP. There is no explicit control over the scene. Another pitfall is that for each pixel in the 3D, while ray casting, it has to go over 8 layers of the Neural Network to get the color and density (c, σ) pair, which is time-consuming and inefficient. On the other hand, the voxel representation [35], which stores all voxels of color and density in a cube, has speed and explicit control. Nevertheless, it requires a cubed amount of memory to store the values, which makes it resource-demanding. To overcome such limitations, the authors in [6] proposed a hybrid tri-plane representation of the 3D scene, occupying the advantages of both.

In [6], instead of using a full cube of voxels for explicit representation, they use feature planes along three axis-aligned orthogonal planes. The XYZ planes comprise the tri-plane representation, denoted as $\mathbf{S}$. For any 3D position $x \in \mathbb{R}^3$, it can be projected onto each of the three feature planes, retrieving the corresponding feature vector (F_{xy}, F_{xz}, F_{yz}) via bilinear interpolation. Then the features can be put together through summation and fed into a lightweight MLP to interpret 3D features F as color and density (c, σ). The overall structure is shown in Fig 4.

The tri-plane representation has been increasingly popular in recent months as 3D representations [6, 17], generations [1, 20], and interpretation [4], proving its efficiency and effectiveness.

4. Method

Our overall method consists of two stages approach similar to Latent Diffusion models. In stage 1, an encoder-decoder is trained to learn a z intermediate triplane representation. Where the encoder is a CNN that receives an image as input and outputs an intermediate triplane representation, the decoder consists of a CNN and a neural render model. The CNN receives an intermediate latent representation and outputs an upscaled triplane representation which is fed to the neural render to generate an image given a camera parameter. Figure 5 illustrate stage 1 training. In stage 2, a diffusion model is trained to generate the intermediate triplane representations. Figure 6 illustrate stage 2 training.

4.1. Stage 1: Triplane Intermediate Representation

Let $\mathbf{S}$ be a triplane representation of a 3D scene, and x^p an RGB image for camera parameters p. The objective of stage 1 is to learn a model f such that

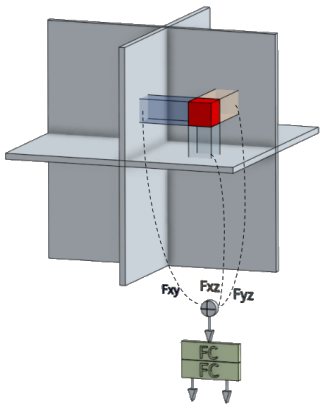

Figure 4. Tri-plane overview. The red cube indicates a 3D position x aggregated from three feature planes

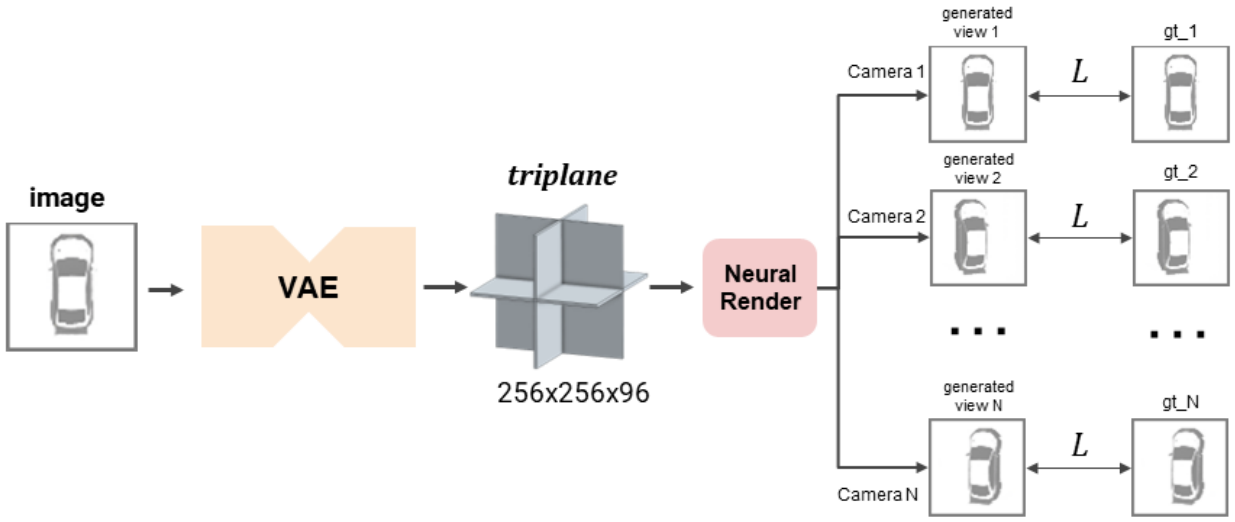

Figure 5. Stage 1 Training. An image is fed to an encoder-decoder to learn a tri-plane representation. The tri-plane representation is fed to the neural render to generate views given the camera parameters. Then the loss is computed regarding the generated view with the ground-truth gt.

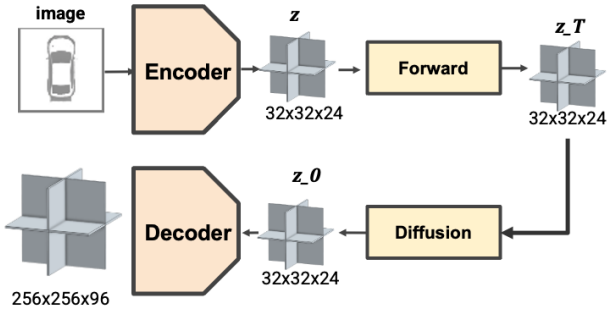

Figure 6. Stage 2 Training. An image is encoded to get the intermediate triplane representation. The forward process adds noise to compute $z_1, \dots z_T$. The diffusion process denoises the $z_t, \dots, z_0$. The decoder upscales the intermediate triplane representation.

$$f(x^p) = \mathbf{S} \tag{4}$$

In this work, the model f consists of an autoencoder architecture where the image encoder $\mathcal{E}$ is trained to output the intermediate triplane representation z, and the decoder $\mathcal{D}$ is a triplane decoder trained to output the triplane representation $\mathbf{S}$. Note that z is a low dimensional representation of $\mathbf{S}$, and the purpose of the decoder $\mathcal{D}$ is to upscale the intermediate representation z to $\mathbf{S}$.

$$\mathcal{E}(x^p) = z; \quad \mathcal{D}(z) = \mathbf{S} \tag{5}$$

In order to force the models to learn 3D information, a neural render function ψ defined as

$$\psi(S, p) = x^p \tag{6}$$

where S is the triplane representation and p is the camera parameters for a target view x^p. Given a set of images with different camera parameters $x^{p_0}, x^{p_1}, \dots, x^{p_N}$ our loss function is defined as

$$\mathcal{L} = \sum_{j}^{N} \sum_{i}^{N} \mathcal{L}_p(x^{p_i}, \psi(f(x^{p_j}), p_i)) \tag{7}$$

where $\mathcal{L}_p$ is the perceptual loss [65]. In other words, for a given triplane $f(x^{p_j})$, multiple views $p_0, p_1, ..., p_N$ can be computed with ψ. To minimize the loss, $\psi(f(x^{p_j}))$ should be similar to x^{p_i} for $i \in 1..N$. This will encourage the model to encode 3D information in its representation. In practice, not all views p_i are used for computing the loss due to memory resources. In our case, we used N = 2 in our experiments due to limited resources. Figure 5 illustrates how the loss $\mathcal{L}$ is computed.

4.2. Stage 2: Diffusion Training

The purpose of stage 2 is to increase the diversity of the synthesis of triplanes. In order to reduce computation and training time, the diffusion model is applied at the intermediate triplane representation z. Note that using the full triplane representation for the diffusion model will result in increased computation time due to the size of the data.

Let $z_0 = \mathcal{E}(x^{p_i})$ where z_0 is the intermediate triplane representation from the encoder. Let $\epsilon_\theta(z_t, t)$ be our denoising network, a U-Net-based model KL-regularized similar to [50]. We optimize our denoise network using the Latent Diffusion loss introduced by [50], which is defined as:

$$L_{LDM} = \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[||\epsilon - \epsilon_\theta(z_t, t)||_2^2\right] \quad (8)$$

4.3. Sampling images given a camera parameter

At inference, a noisy intermediate triplane representation is sampled by $z_t \sim \mathcal{N}(0, 1)$, and it is denoised iterative by the diffusion model $\epsilon(z_t, t)$ until it reaches z_0. In order to accelerate the sampling, we used a DDIM [57] at the denoising process. The denoised z_0 is fed into $\mathcal{D}$ to compute the upscaled version of the triplane S. This triplane S contains the 3D representation of the scene, from which multiple images from different views can be computed. A image x^{p_i} can be computed using $\psi(S, p_i)$ by giving the target camera parameter p_i.

4.4. Implementation details

For our experiments, three datasets were used FFHQ [27], ShapeNet Cars [8] and a Triplane dataset. FFHQ dataset contains images of faces taken from a diverse variety of camera parameters. To accelerate experimentation, image resolution was down-sampled to 64x64. For this dataset, camera parameters, intrinsic and extrinsic, were computed using an off-the-shelf [13] model as described at [6]. The ShapeNet car dataset is a 3D assets dataset for cars. For each 3D asset, 50 views from different camera parameters were extracted with a resolution 128×128. Each image is associated with its respective camera parameter. The triplane dataset was created by using the Eg3D model trained on the ShapeNet car dataset. We sampled 3000 triplane representations of size $256 \times 256 \times 96$.

This work was built on top of the official repository of LDM [1]. Stage one uses a VAE trained with standard KL-regularization. The neural render is an MLP with two layers, the first with an input of 64 units and the second with four units (RGB + Density). Stage 1 is optimized with the loss in Equation 7. Stage 2 uses a latent diffusion model unconditional unless it is specified. The Stage 2 architecture is a UNet with attention to resolutions 8, 4, and 2.

5. Experimental Results

In this section, we discuss our experiments. First, we present single-view training, our first approach, and show that DM cannot learn 3D priors from a single image by just applying our methods. In section 5.2, we use a triplane dataset constructed by the eg3D method to show that DM can learn triplane generations. In section 5.3, we aim to learn our model to encode 3D information without relying on another method to get the triplane representation.

5.1. Experiment 1: Single View Training

This experiment aims to evaluate if the diffusion model can learn a 3D-aware representation by using a single-view image during training. In order to achieve this, we condition the diffusion model to the camera parameters of the target image. We relied on the prior from the diffusion model to generate 3D information not visible in the single-view image. We hypothesized that the DM could exploit the 3D prior from images at the generation of triplane representations.

We trained our model with the FFHQ dataset using 64x64 image resolution. Each image has a camera parameter associated, which was computed using an off-the-shelf model. This dataset does not contain multi-view images for each identity; therefore, the loss function for stage 1 was adapted as follows:

$$\mathcal{L} = \sum_i \mathcal{L}_p(x^{p_i}, \psi(f(x^{p_i}), p_i)) \quad (9)$$

Figure 8c and 8d show the results of our model. These examples were generated by sampling a triplane from the trained DM and generating images with a target camera parameter by using the same triplane. However, the diffusion model was able to synthesize a photorealistic image of a person but failed to learn 3D information. The triplane collapses into a single plane or flat image.

We also sampled images using DDIM from the same noise and changed the DM condition with their target camera parameters. The idea is that the X_0 should capture the content or identity of the generation, and the camera parameters should be independent of the identity. Figure 8a and 8b show the images generated by this procedure. These images show that the model entangled both the identity and the camera parameters; therefore, when generating, it produces a new identity at each camera parameter.

With this experiment, we notice two critical issues:

- There is no training signal that enforces 3D information at the triplane; therefore, the model produces flat images.
- The diffusion model conditionated with camera parameters will entangle both content and camera parameters. Also, this approach will be slow to run.

5.2. Experiment 2: Triplane Training

After experiment 1, we want to investigate whether learning triplane representations using DM was even possible. This experiment aims to evaluate DM trained on precomputed triplanes using encoding from other approaches. We hypothesize that the DM can generate triplanes keeping diversity in their reconstructions.

[1] https://github.com/CompVis/latent-diffusion

Figure 7. Some generation results for shapeNet dataset [8], showing diversity in both types and texture in the generated cars

(a) Conditioned DM (left view) (b) Conditioned DM (right view)

(c) Fix Triplane (left view) (d) Fix Triplane (right view)

Figure 8. Examples from the single-view model with the conditioning diffusion model.

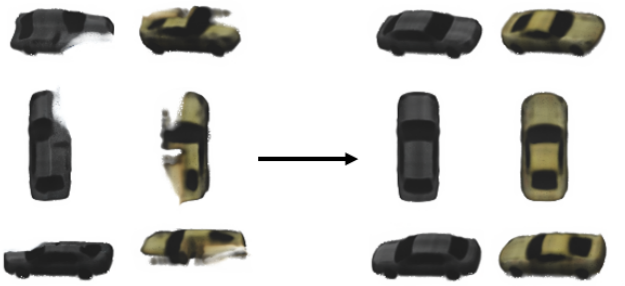

Figure 9. Our diffusion model successfully fills the missing parts of car models. A small cylinder is used to mask in the left, and the right is masked with a thin long cuboid.

We created a dataset using the Eg3D model trained from the ShapeNet car datasets. 3000 generations were sampled. For each generation, we store a triplane with a dimension of 256x256x96. For this experiment, the input and output of stage 1 is the triplane with dimensions 256x256x96. The encoder will reduce the triplane representation into a latent triplane representation 32x32x24. This representation will be fed into the decoder to upscale the triplane. No Neural render model was used at stage 1 for this experiment. For stage 2, the diffusion model is applied to the latent triplane representation. This approach results in faster training due to the reduction of size. At inference, an eg3D neural render model trained on the ShapeNet car dataset is used to compute images from triplanes.

Examples of this model are shown in Figure 7. This figure shows that the model was able to generate a sample successfully capturing diversity in their generation. These results demonstrate that diffusion models can learn triplane representation and that applying the diffusion model to the latent triplane representation is effective.

We also explore the task of triplane inpainting. For this, we removed partial content from the triplane and fed it to the network to generate the missing content. Figure 9 shows that our model successfully reconstructed the missing content of the triplane.

From this experiment, we learn that the DM can learn the 3D structure of the triplane. These methods showed that they could generate 3D representation and shows diversity in their generations. We showed that our model achieves 3D information understanding and can fill in missing content.

5.3. Experiment 3: Multi-view training

Motivated by the previous experiment, we want to remove the requirement for a triplane representation dataset. This experiment aims to train our model to generate triplane representation. From experiment 1, we learned that learning triplane from single-view training is challenging; therefore, we decided to use multi-view training for this experiment. The idea is to train an encoder to learn 3D consistent information by enforcing the reconstruction of multiple views from a single triplane.

To warm up the model, we trained the model to recon-

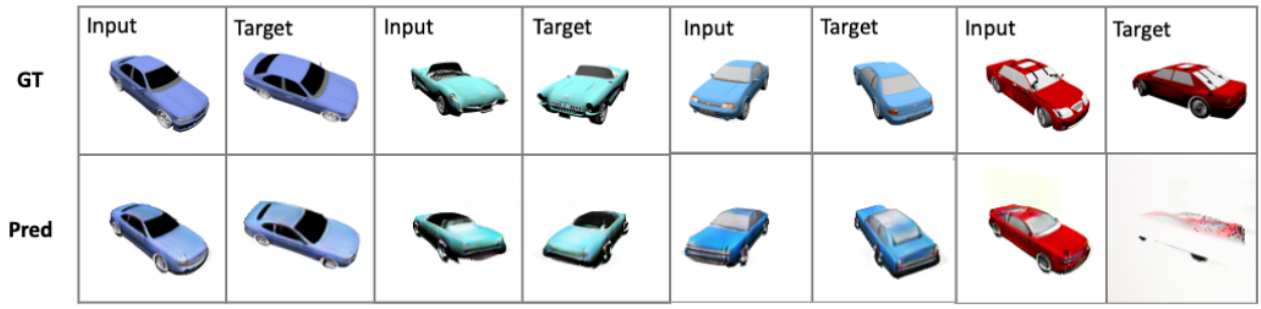

Figure 10. Examples of the multi-view model. Each sample has an input camera parameter (*left*) and a target camera parameter (*right*), which ground truth at the top. Model reconstructions are top with their corresponding camera parameters.

struct a single view as in experiment 1. During this warm-up, the model will learn to encode visual features in the triplane to reconstruct the input image. Later this model is trained to reconstruct the two target views. The first is the same as the input image, and the second is a different view from the input image.

We trained a VAE with KL regularization on the ShapeNet Car dataset during 32 epochs for the warmup and 2 epochs for fine-tuning. Figure 10 shows the reconstruction results for both the input and image from a different view. This Figure shows that our model could learn 3D information at the triplane representation and render images from challenging views. The Figure's third example shows that it can generate a view from the other side of the current view. But this model is imperfect; the fourth example in Figure 10 shows that the model fails to generate the entire 3D object.

This section showed that our model could learn 3D information with multi-view training. Although this model is not perfect, we believe that this model can improve by fine-tuning the model for more epochs. In the future, a diffusion model can be trained on the intermediate triplane representation.

6. Discussions, Limitations, and future work

In experiment 5.1, we showed that the triplane could not learn 3D information from a single view with our approach and that conditioning our diffusion model to camera parameters leads to entangling both identity and camera parameters at the denoising process. This proves the conditional-discriminator in Eg3D [6] may not be applicable to all models or tasks, while our multi-view training forces explicit 3D consistency and thus is more robust in maintaining 3D information.

Although our works show promising 3D results generated from the model, there are multiple paths to expand the architecture. First and foremost, our current rendering resolution is 128x128, which is relatively small. Artifacts and poor details arise if rendered with higher resolution. This can be overcome by utilizing a 3D-aware super-resolution block as in [6]. Our model also requires prior knowledge other than normal training images, e.g. camera parameters. Although there exist off-the-shelf models like [13] to create pseudo-labels, or one can learn the pose distribution on the fly [44], it is observed that the results can be unreliable [19]. The true independence of camera poses from camera parameters remains an open challenge.

Limited by time and computing resources, we are not able to train the diffusion model with conditioning, such as images or text. We also look forward to extending to more domains or even to domains of general 3D objects. In the future, we plan to work on these improvements to better controlling of the 3D diffusion model with maximum generality.

7. Conclusion

Diffusion models and Neural Rendering are topics of growing interest in the computer vision community. In this work, we leverage the generative power of diffusion models to generate 3D representations. We proposed our approach to learning a DM for generating 3D representations.

Our experimental results suggest that by using multi-view training, our model can learn a triplane representation that learns 3D information. We also showed that the diffusion model can be trained on triplane representations to increase sample diversity. This may enable a better training strategy for 3D information, and thus lead to more controllable 3D synthesis and innovative methods for reconstructing 3D shapes and novel views.

References

[1] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. *arXiv preprint arXiv:2211.09869*, 2022.

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields, 2021.

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields, 2023.

[4] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. *arXiv preprint arXiv:2301.09632*, 2023.

[5] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *arXiv*, 2020.

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

[7] Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. *arXiv preprint arXiv:2304.02602*, 2023.

[8] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[9] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion NeRF: A unified approach to 3D generation and reconstruction. Apr. 2023.

[10] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. SceneDreamer: Unbounded 3D scene generation from 2D image collections. Feb. 2023.

[11] Congyue Deng, Chiyu Max, Jiang, Charles R Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, and Stanford University. NeRDi: Single-View NeRF synthesis with Language-Guided diffusion as general image priors.

[12] Kangle Deng, Gengshan Yang, Deva Ramanan, and Jun-Yan Zhu. 3d-aware conditional image synthesis. *arXiv preprint arXiv:2302.08509*, 2023.

[13] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2019.

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[15] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. *arXiv preprint arXiv:2303.17015*, 2023.

[16] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[17] Sara Fridovich-Keil, Giacomo Meanti, Frederik Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. *arXiv preprint arXiv:2301.10241*, 2023.

[18] IJ Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, and Y Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[19] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations*, 2022.

[20] Jiatao Gu, Alex Trevithick, Kai-En Lin, Josh Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. *arXiv preprint arXiv:2302.10109*, 2023.

[21] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023.

[22] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. 2023.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[24] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022.

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

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[27] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, 43(12):4217–4228, dec 2021.

[28] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[29] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. NeuralField-LDM: Scene generation with hierarchical latent diffusion models. Apr. 2023.

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[31] Wei Li, Wei Ding, Rajani Sadasivam, Xiaohui Cui, and Ping Chen. His-gan: A histogram-based gan model to improve data generation quality. *Neural Networks*, 119:31–45, 2019.

[32] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5871–5880, 2020.

[33] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022.

[34] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems*, 34:16331–16345, 2021.

[35] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

[36] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. Mar. 2023.

[37] Jinqi Luo, Zhaoning Wang, Chen Henry Wu, Dong Huang, and Fernando De la Torre. Zero-shot model diagnosis. *arXiv preprint arXiv:2303.15441*, 2023.

[38] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[39] Luke Melas-Kyriazi, I Laina, C Rupprecht, and Andrea Vedaldi. Realfusion: 360 reconstruction of any object from a single image. 2023.

[40] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[41] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. *arXiv preprint arXiv:2212.01206*, 2022.

[42] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[43] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[44] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[45] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020.

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

[47] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[49] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021.

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[51] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022.

[52] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[53] Kyle Sargent, Jing Yu Koh, Han Zhang, Huiwen Chang, Charles Herrmann, Pratul Srinivasan, Jiajun Wu, and Deqing Sun. Vq3d: Learning a 3d-aware generative model on imagenet. *arXiv preprint arXiv:2302.06833*, 2023.

[54] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[55] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[56] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9243–9252, 2020.

[57] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[58] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. DiffuScene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis. Mar. 2023.

[59] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[60] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[61] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022.

[62] Jianfeng Xiang, Jiaolong Yang, Binbin Huang, and Xin Tong. 3d-aware image generation using 2d diffusion models. *arXiv preprint arXiv:2303.17905*, 2023.

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

[64] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021.

[65] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society.

[66] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis. 2021.

[67] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.