DOI: 10.1111/cgf.14930

Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

Learning to Generate and Manipulate 3D Radiance Field by a Hierarchical Diffusion Framework with CLIP Latent

Jiaxu Wang[1], Ziyi Zhang[1] and Renjing Xu[1]

[1]Hong Kong University of Science and Technology (Guangzhou)

Abstract

3D-aware generative adversarial networks (GAN) are widely adopted in generating and editing neural radiance fields (NeRF). However, these methods still suffer from GAN-related issues including degraded diversity and training instability. Moreover, 3D-aware GANs consider NeRF pipeline as regularizers and do not directly operate with 3D assets, leading to imperfect 3D consistencies. Besides, the independent changes in disentangled editing cannot be ensured due to the sharing of some shallow hidden features in generators. To address these challenges, we propose the first purely diffusion-based three-stage framework for generative and editing tasks, with a series of well-designed loss functions that can directly handle 3D models. In addition, we present a generalizable neural point field as our 3D representation, which explicitly disentangles geometry and appearance in feature spaces. For 3D data conversion, it simplifies the preparation pipeline of datasets. Assisted by the representation, our diffusion model can separately manipulate the shape and appearance in a hierarchical manner by image/text prompts that are provided by the CLIP encoder. Moreover, it can generate new samples by adding a simple generative head. Experiments show that our approach outperforms the SOTA work in the generative tasks of direct generation of 3D representations and novel image synthesis, and completely disentangles the manipulation of shape and appearance with correct semantic correspondence in the editing tasks.

CCS Concepts

• ***Computing methodologies*** → ***Shape modeling; Image manipulation;***

1. Introduction

Neural Radiance Field (NeRF) [MST*21] has gained a lot of attention in recent years due to its promising capability to synthesize photo-realistic novel views from arbitrary views. The dominant performance has the potential to significantly impact and revolutionize applications in many fields including computer graphics, computer vision, augmented reality/virtual reality, etc. On account of the success of NeRF, a number of extended works have been proposed in various fields, such as dynamic scene, generalization, unconstrained scene exploration [DBS*21], and scene understanding [FZC*22, KGY*22].

The conventional NeRF pipeline is primarily limited to specific, individual scenes and is unable to be edited. Recently, researchers have brought the Generative Adversarial Network (GAN) into NeRF representation for generating and manipulating 3D scenes. In contrast to 2D GAN, they use 3D-aware GAN which incorporates a 3D-structure-aware inductive bias within the generator network architecture. As for editability, a widely used approach in 3D-aware GAN is to separate latent codes for shape and appearance and feed them to different layers of the generators [CMK*21, GLWT21, ZXNT21, CLC*22, HMR19]. Users can specifically manipulate the neural scene by controlling the disentangled latent codes, such as appearance and shape transfer. Furthermore, other studies incorporate condition settings into the generative pipeline to perform 3D-aware image manipulation for a given condition [JSJ*21, WCH*22]. These methods achieve better and more detailed editing but still cannot fully disentangle the changes in shape and appearance because the predicted shape and color still share some hidden features in the shallow layers of the generators. In addition, 3D-aware GANs consider NeRF pipelines as regularizers and do not directly operate with 3D models, which results in imperfect 3D consistencies. More importantly, similar to 2D GAN, 3D GAN also suffers from issues including mode collapse and complex training paradigms. ChangeIt3D [AHS*23] and LADIS [HAZ*22a] edit 3d shapes through parsing language and do not edit colors.

Meanwhile, diffusion models have recently advanced to many 2D generation tasks including image generation [GCB*22, NDR*21, RLJ*23, KSH22, SHC*22], image super resolution [RBL*22, SHC*22, DMH21], inpainting [LDR*22, XZL*23], and image manipulation [ALF22, MHS*21, KZL*23, YGZ*23, ZHG*23], and outperform GAN-based models on several benchmarks. This is because diffusion models are much easier to train and offer better diversity. In addition, it can be observed that diffusion models deliver promising performance on multimodal data [RLJ*23, TRG*22, PVG*21, SPH*22, ALF22, BNX*23, GCB*22, NDR*21, PJBM22, XWC*23, LGT*23] and conditional generations [NSL*23, ZLW*23, BNX*23, GCB*22, NDR*21]. Moreover, several studies have introduced diffusion models into 3D geometry generations such as point cloud [VWG*22], while extending them to directly generate neural implicit representation remains difficult. The reasons lie in three aspects: **1**. Implicit representations often rely on the parameters of neural networks, which can hardly be generated. **2**. Training the 3D diffusion models requires plenty of groundtruth samples of 3D assets. In the context of NeRF, obtaining such a number of groundtruth data is impractical because per-scene NeRF optimization requires a very long time. Even if there are some volumetric radiance field approaches that can converge faster [SSC22], it still needs more days to prepare a dataset consisting of a few thousand samples. On the other hand, some studies have investigated optimizing a volumetric NeRF by score-based distillation from a pretrained 2D image generation diffusion model [PJBM22, XWC*23, LGT*23, HCO*23, WDL*23, CSL*23, RKP*23]. These approaches are time-consuming and memory-inefficient, and constrained by the quality and consistency of generated 2D images.

In this paper, we reformulate the 3D NeRF generation task and disentangle the editing task into a unified framework by a purely hierarchical diffusion-based framework, which includes three main stages. The first is responsible for 3D generation, and the last two are employed for geometry and appearance manipulation respectively. In contrast to GAN-based models, the presented diffusion model directly generates and manipulates 3D assets, maintaining full 3D consistency. Different from the previous works which use volumetric radiance field as 3D representation, we propose a novel disentangled neural points field as our 3D neural representations. This representation helps us explicitly separate shape and appearance in our diffusion-based pipeline and ensures fully independent changes. Besides, the neural representation is trained under the generic setting rather than single scene optimization, which significantly reduces the difficulties of 3D dataset conversion. For our three-stage diffusion model, the first stage learns the generative model in the CLIP latent space for directly generating new 3D assets in an unconditional way. The second stage is conditioned on the CLIP latent vector and generates neural points with geometry features. The third stage produces the appearance features for the given point scaffold and another CLIP latent vector. Thanks to the high-level semantic information encoded in the CLIP vector, our method can faithfully learn the semantic correspondences in a disentangled manner but does not require different shape-appearance pairs. Additionally, we surpass the complex inverse GAN techniques in the recreation of realistic content because our pipeline is directly conditioned on the image/text prompts and performs manipulation in a forward manner without the need to reproject the reference image/text to the original Gaussian space.

In summary, our contributions are in the following:

- We propose a three-stage diffusion-based unified framework for both 3D generation and text-image driven manipulation tasks, achieving diverse generation and shape and appearance control in a disentangled manner. To the best of our knowledge, our approach is the first work purely based on diffusion models for editing neural scenes, avoiding basic problems associated with GAN-based approaches.
- We are the first to propose and use the generic neural point field as the representation for the target task, which can explicitly ensure fully independent changes and simplify the pipeline of 3D dataset preparations.
- We establish remarkable results for both generation and manipulation by improving over the previous SOTA performance on generation and manipulation tasks.

2. Related Work

2.1. Neural radiance field editing.

Over the last few years, studies of implicit representations experience significant progress, which represents 3D models by deep neural networks. Among them, Neural Radiance Fields (NeRF) [MST*21] achieved the most impressive rendering results by optimizing the 5D neural radiation field of the scene. The impressive performance of NeRF has inspired several subsequent works that have extended its capabilities such as dynamic scene [PCPMMN21, TZFR23, LNSW21, CJ23, TTG*21], generalization [WWG*21, CXZ*21, JLF22, LPL*22], acceleration [WLC*23, CJ23, GKJ*21, LLZ*22, LLW*23] and few-shot reconstruction [JTA21, CLH*22, YPW23]. Despite improving the quality of reconstructing challenging scenes, they are not capable of explicitly controlling and editing the scene.

Editing 3D scenes has received significant interest in the computer graphics community. In order to enable the editability of NeRF, researchers have investigated a range of hybrid semi-implicit neural representations and the integration of generative models into the NeRF rendering framework. For the former, Neumesh [YBZ*22] and NeRF-Editing [YSL*22] store implicit neural features into an explicit mesh scaffold to edit the neural representation by controlling the container. But they are only able to make local-scale edits, not possible to replace the entire shape or appearance. In contrast, generative-based methods achieve global-level edit. GRAF [SLNG20] is a 3D aware generative model and first adopts shape and appearance codes to conditionally synthesize NeRF. GIRAFFE [NG21] can learn individual objects from unprocessed image collections without requiring additional guidance. CodeNeRF [JA21] splits shapes and textures by learning to embed them individually and edit them by changing the latent codes. EditNeRF [LZZ*21] was developed to address the challenge of editing 3D models represented by millions of network parameters. However, the above approaches either require time-consuming optimization processes or cannot ensure fully independent changes. CLIP-NeRF [WCH*22] have attempted to address these problems by using separate latent codes for shape and appearance and feeding them to different layers of the NeRF generator. Furthermore, GDRF [WDY*22] learns to edit specific categories by deforming and recoloring a template space, thereby significantly reducing the training complexity. Nevertheless, these two methods are still based on 3D-aware Generative Adversarial Network (GAN). The 3D consistency cannot be ensured and the details are blurry. This is because the NeRF pipeline is only regarded as the regularization in their generation pipeline rather than directly generating 3D assets. Our proposed method combines the two main branches, explicitly disentangles shape and appearance and directly generates 3D neural scene representation, which effectively overcomes the above issues.

2.2. Diffusion models.

Diffusion models [SDWMG15], originally proposed by Sohl-Dickstein *et al.*, are mainly trained to denoise data that has been perturbed by Gaussian noise. Diffusion model can be classified into two main categories: namely denoising diffusion probabilistic models (DDPM) [HJA20] and noise-conditioned score networks (NCSN) [SE19]. The two types of approaches can be generalized by the score stochastic differential equations theory [SSDK*20]. To develop further based on the diffusion model, researchers mainly focus on three primary areas: sampling acceleration, likelihood maximization, and data generalization. [CHIS23] summarizes the current progress in computer vision.

Diffusion model has emerged the most advanced deep generation model and has been applied in a wide range of fields, including image super resolution [RBL*22, SHC*22, DMH21], image inpainting [LDR*22, XZL*23], image editing [MHS*21, KZL*23, YGZ*23, ZHG*23], semantic segmentation [HAZ*22b, BRV*21, BKC*22, GMJS22], video generation [HNM*22, HCS*22, ZCP*22, QCZ*23], natural language processing [AJH*21, GLF*22, HKT22, LTG*22], point cloud completion [LWYL22, LH21, VWG*22, ZDW21] and multi-modal generation [RLJ*23, TRG*22, PVG*21, SPH*22, ALF22, BNX*23, GCB*22, NDR*21, PJBM22, XWC*23, LGT*23], as well as interdisciplinary applications in fields such as and medical image reconstruction [CSY22, CY22, PGZ*22a, PGZ*22b]. Notably, in the area of high-resolution image generation, the impact of diffusion models has surpassed that of GANs. It overcomes the vulnerability of GANs, i.e. mode collapse, has more stable training processes, and produces more diverse and meaningful results. Despite the original diffusion model having longer training and sampling time, many recent studies have aimed to reduce them.

2.3. 3D generation.

Generative adversarial approaches [GPAM*20] are achieving noteworthy results in the field of 3D generation, such as generating voxel-based representations [CCS*19, KAAL22], meshes [**?**, GSW*22], or NeRFs [CMK*21, GLWT21, ZXNT21, CLC*22, HMR19]. Pi-GAN [CMK*21] have introduced a GAN-based approach to the standard NeRF [MST*21] model, incorporating a form of stochastic conditioning and trained using an adversarial loss. CIPS-3D [ZXNT21] have addressed the issue of high memory costs and lengthy training times by having their volume rendering components output low-resolution 2D feature maps. These maps are then upsampled using efficient convolutional networks to generate the final images. StyleNeRF [GLWT21] has addressed the issue of 3D inconsistency by carefully designing the convolution stage to minimize such inconsistencies. EG3D [CLC*22] is a hybrid explicit-implicit network that real-time synthesis of high-resolution images that maintain consistency across multiple viewpoints. However, the previously mentioned GAN-based methods are not capable of supporting conditional generation and lack full 3D consistency. In contrast, the image condition generation model is better at reasoning ambiguity and producing diverse and meaningful content.

More recently, the diffusion model has been introduced into 3D generation. These generation methods are mainly divided into two branches. The first is to distill the 2D pre-training diffusion model into a 3D NeRF representation, which mainly is a voxel-based neural representation [PJBM22, XWC*23, LGT*23, HCO*23, WDL*23, CSL*23, RKP*23]. This type of methods often requires long optimization time and under-averaged quality due to the inconsistency of the images generated by the 2D diffusion models. The other is to directly generate 3D assets through the denoising diffusion model. DiffRF [MSP*23] is the first to generate 3D neural representations from random noises in a straightforward manner. However, the use of the volumetric radiance field leads to low resolution and inefficient memory consumption. Our method is also able to produce 3D entities straightforwardly, and we achieve 3D shape and texture manipulation by image/text prompts in a disentangled style.

3. Generic neural point field.

Different from previous works in which the volumetric radiance field is widely adopted to represent 3D scenes, we aim to simplify the dataset preparation pipeline and explicitly separate shape and appearance representations. It is noted that the goal of the 3D-aware GAN is to synthesize novel images rather than the entire 3D representation of objects. Therefore, we propose the generic disentangled neural point field to represent 3D objects.

Figure 1 shows the workflow about how to build the neural point field from existing images and the related volumetric rendering process. For the dataset containing various multi-views,here we assume the number of views is n. We use two different heads to extract geometry and appearance features respectively. The corresponding depth maps (which can be estimated from multiview images if the dataset does not provide them) are transformed into point clouds following the camera parameters, and the extracted two feature maps are projected to the point cloud and transformed to 8 dimensions by an MLP. The n feature vectors are summed and weighted by the associated binary mask weights ($f = SUM(w_i * f_i), i = 1,..,n$). Each point contains two feature vectors associated with the shape and appearance information. Next, the normal volume rendering technique (Equation 1) is conducted to access such neural point field via sampling points along rays.

$$c = \sum_{k=1}^{M} \tau_k (1 - exp(-\sigma_k \delta_k)) c_k, \tau_k = exp(- \sum_{t=1}^{k-1}) \sigma_t \delta_t, \qquad (1)$$

The sampled points aggregate features from their nearby neural points by inverse distance weights that is to regulate the degree of influence of the nearest K neural points $p_j \mid j = 1, 2, \ldots, K$.

$$g_x = \textstyle\sum_{j=1}^{K} \frac{w_j}{\sum w_j} g_{j,x}, w_j = \frac{1}{\left\| p_j - x \right\|} \qquad (2)$$

Then we use two independent decoders to regress transmittance *sigma* and colors c. After integrating colors along rays, images

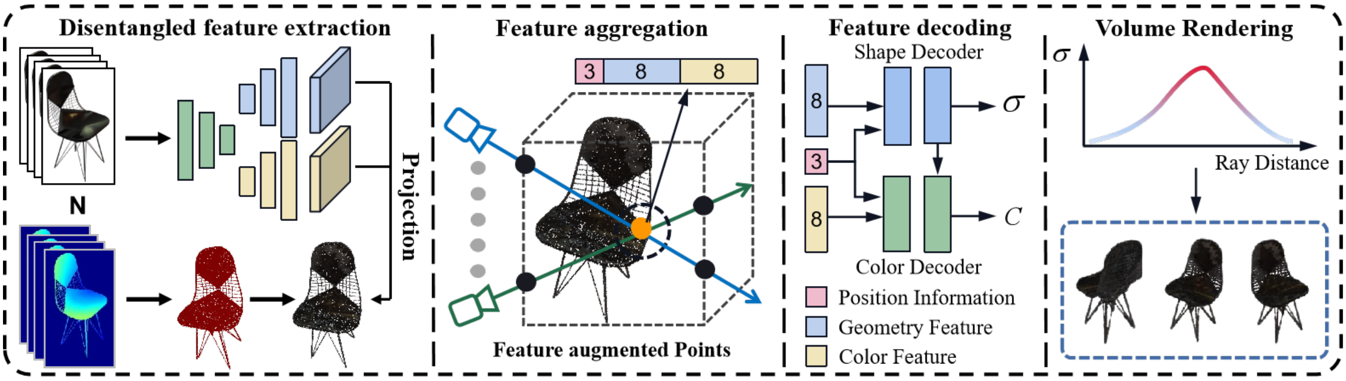

Figure 1: ***Overview of neural point rendering process.*** *The volumetric rendering based on the proposed neural point field is depicted. Multiview images are input to UNet with two output heads for disentangled feature extraction. The features are fetched to the point scaffold transferred from the depth map. We aggregate features for sampling points from neighboring neural points and separately input them to the shape and color decoder to obtain* σ *and c.*

of specific viewpoints are synthesized and we compute L2 loss to drive the training of the framework.

More importantly, the disentangled neural point field is trained on multiple scenes hence it can be generalized to unobserved scenarios. In other words, for preparing 3D datasets, it is easier to convert datasets with multiview images into neural representations because per-scene optimization is not required.

4. Diffusion model.

Diffusion models progressively destroy samples by injecting noise sampled from multivariate Gaussian distribution, then learn to reverse this process and generate the result by iterative denoising the random initialization. Traditional DDPM [HJA20] is based on the Markov chain method. Its diffusion process generates a sequence of $x_1, x_2, \cdots, x_T$ from a given 3D object radiance field x_0 by the forward Markov process. Each diffusion step is carried out by the pre-defined Gaussian perturbation $q\left(x_t \mid x_{t-1}\right)$ and the whole process can be described by

$$q\left(x_{1:T} \mid x_0\right) = \prod_{t=1}^{T} q\left(x_t \mid x_{t-1}\right) = \prod_{t=1}^{T} \mathcal{N}\left(x_t \mid \sqrt{1-\beta_t}, \beta_t I\right), \quad (3)$$

where $\beta_t \in (0,1)$ is a chosen hyperparameter to implement a schedule for the injected noise variance. This process can also express the conditional distribution of x_t given x_0 in a straightforward manner by using properties of Gaussian distribution, resulting in Equation 4.

$$q\left(x_t \mid x_0\right) = \mathcal{N}\left(x_t \mid \sqrt{a_t} x_0, (1-a_t) I\right), \quad (4)$$

where $a_t = \prod_{k=0}^{t}(1-\beta_k)$. This transformation can be employed to efficiently generate x_t for arbitrary time steps given a predetermined x_0.

The generation process, also known as the denoising process, reverses the diffusion process by learning Gaussian transition kernels parameterized by deep neural networks. Song et al. [SME20] prove that if $q(x_t \mid x_{t-1})$ satisfies a Gaussian distribution and β_t is small enough, $q(x_{t-1} \mid x_t)$ is still a Gaussian distribution. Thus the distribution $P(x_t - 1 \mid x_t)$ can be estimated by Equation 5.

$$p_\theta\left(x_{t-1} \mid x_t\right) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_t\right), \quad (5)$$

where the μ_θ and $\sum_\theta$ refer to the mean and variance of the Gaussian distribution $P(x_t - 1 \mid x_t)$. In the original DDPM, the variance can be explicitly derived from the diffusion coefficients.

$$\sum = \frac{1-\alpha_{t-1}}{1-\alpha_t} \beta_t I \quad (6)$$

Furthermore, μ_θ can also be reparameterized by the noises at t timestamp, yielding Equation 7.

$$\mu_\theta\left(x_t, t\right) = \frac{1}{\sqrt{1-\beta_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-a_t}} \varepsilon(x_t, t)\right), \quad (7)$$

Denoising diffusion implicit model (DDIM) [SME20] replaces the Markov-based forward process in DDPM using a Non-Markovian forward process. The edge distribution of DDIM forward process also satisfies the form of edge distribution introduced under Markov assumption, then it is still possible to reuse the optimization objective of DDPM. When the form of the objective distribution of the Non-Markovian is supported at any time step, the objective distribution is

$$q\left(x_{\hat{t}} \mid x_t, x_0\right) = \mathcal{N}\left(\sqrt{\alpha_{\hat{t}}} x_0 + \sqrt{1-\alpha_{\hat{t}}-\sigma^2} \frac{x_t - \sqrt{\alpha_t} x_0}{\sqrt{1-\alpha_t}}, \sigma^2 I\right), \quad (8)$$

where $\hat{t}$ refers to $t-1$, σ is the parameter that describes noises added in the forward process. The free noise variable σ determines the randomness of the generative process. When $\sigma = 0$, the generative process becomes deterministic.

5. Three-stage disentangled diffusion framework.

Our framework is composed of three generative diffusion models that are built on and extended from DDIM. The first diffusion model aims for interpolation in CLIP latent space, and the generated latent vector can be the condition for the subsequent diffusion modules. The second diffusion module is trained to denoise a

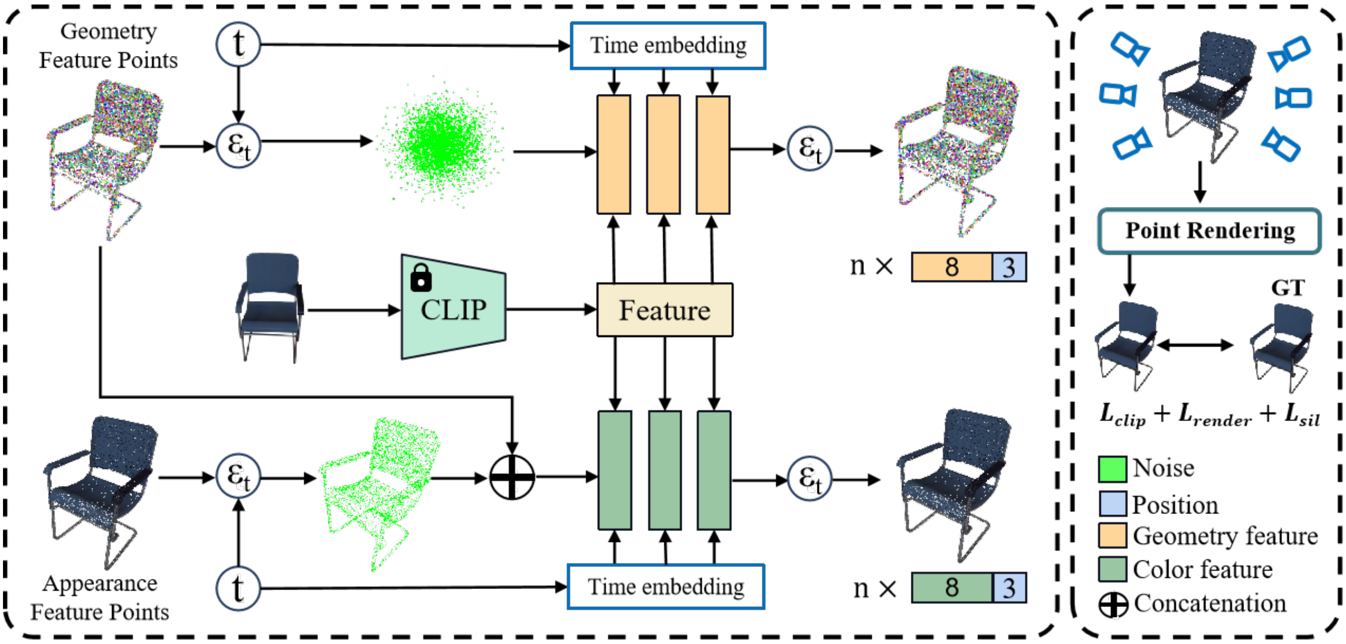

Figure 2: *Training pipeline of the two-stage conditional diffusion model for disentangled manipulation. We separate the neural point field into two parts to guide the training processes of the two diffusion models, namely geometry and appearance. To clarify, we omit the denoising loss which is common for diffusion models.*

high-dimensional spherical Gaussian initialization to produce the neural point representations only with geometry features. The third diffusion step is conditioned on a CLIP vector as well as a point cloud scaffold to generate semantically corresponding appearance features for the given points. Each module is trained independently and jointly worked in the inference stage. In our case, our target is to generate 3D objects in the form of our proposed neural points field and separately manipulate shape or appearance driven by image or text prompts via running corresponding diffusion modules.

At first, we introduce how we separately manipulate the shape and appearance in the two-step diffusion manner. Then we present how the framework generates new samples under the unconditional setting.

5.1. Disentangled training paradigm for scene manipulation modules.

The 3D neural point field can be summarized as a point cloud scaffold with two disentangled geometry and appearance feature vectors for each point. Assume the neural point set as $\{p_i\}, p_i \in R^{19}, i = 1, 2, ..., N$ where 19 denotes the XYZ coordinates (3), the 8-dimensional shape feature vectors and the 8-dimensional appearance vectors. As we presented in section 3, the shape and appearance of rendering results only rely on the associated features and coordinates. Therefore, we explicitly apply two diffusion models to separately generate the two types of feature vectors. In this way, the changes in shape and color will be fully independent and do not share any hidden layer features. The whole training pipeline is presented in Figure 2. As stated in the figure, each denoising model is conditioned on a provided CLIP vector which is encoded from a corresponding image. This image is associated with the x_0, and randomly selected from all given viewpoints in the dataset.

Shape Generation Module. The training process of shape generation starts with the noise injection into the 3D tensors represented by $(Batch, numberofpoints, 11)$, following Equation 5. The denoising model learns to predict the noise at time t (ε_t) for recovering 3D entities. For each step of prediction, the denoising model also receives an associated CLIP latent vector as the conditional input. This CLIP vector can be encoded from either the input images to reconstruct the neural points, as we introduced in section 3, or the corresponding text description from our prepared text library. The denoising is reformulated as $\varepsilon(x_t, t, z)$ to replace the $\varepsilon(x_t, t)$ in Equation 8. The diffusion loss is depicted by Equation 9. The denoising model is implemented by the permutation invariant Set Transformer [LLK*19], which is designed specifically for unordered sequence data such as point clouds. To model the conditional denoising, we feed the time t and the conditional CLIP vector to each layer of the network. The time t is firstly embedded by an MLP.

For each layer, we introduce an additional MLP to project the condition embedding to the hidden dimension. We replace Layer Normalization with Adaptive Group Normalization to pose these conditions. In our experiments, we also try to concatenate all conditions in the input layer, such as in [NJD*22], we found similar performance but slightly lower running speed. After this step, we only obtain the geometry features, thus it is impossible to render colorful images as supervision. By contrast, we are able to obtain the object silhouette by accumulating transmittance along each ray. Therefore, except for the diffusion loss, we additionally design a silhouette loss to supervise the model training (Equation 10). M_a

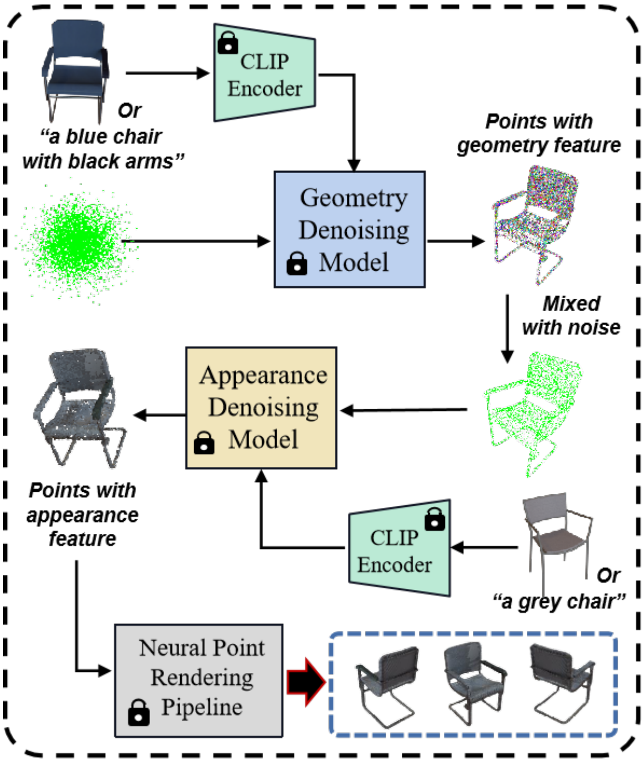
CLIP Encoder
Or
"a blue chair with black arms"
Points with geometry feature
Geometry Denoising Model
Mixed with noise
Appearance Denoising Model
Points with appearance feature
CLIP Encoder
Or
"a grey chair"
Neural Point Rendering Pipeline

Figure 3: *Illustration of disentangled editing by our two-stage manipulation modules after these models are trained. The edit can be simply performed by inputting corresponding CLIP conditions.*

is the rendering function for obtaining the silhouette of the object, in which the $\bar{x}_0$ denotes the denoised neural representation and v refers to the viewpoint associated with the CLIP condition image. We compute the binary cross entropy loss between the groundtruth mask and the prediction to encourage the reconstruction of the geometry part of x_0.

$$L_{diff} = ||\varepsilon_t - \varepsilon_\theta(t, x_t, z)||^2 \tag{9}$$

$$L_{sil} = -(M_{gt} log M_a(\bar{x}_0, v) + (1 - M_{gt}) log(1 - M_a(\bar{x}_0, v))) \tag{10}$$

However, fully denoising all samples requires burden computation resources. Inspired by the forward process of the diffusion model, arbitrary x_t can be directly obtained from x_0 and t via Equation 4. Inversely, if we obtain the ε_t, we can roughly compute the corresponding x_0 by Equation 11. Even though this x_0 is not the optimal result, it is still useful to push the $\varepsilon(x_t, z, t)$ toward the real value.

$$\bar{x}_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \sqrt{1-\alpha_t}\varepsilon(x_t, z, t)) \tag{11}$$

In addition, the rough approximation becomes more accurate as the timestamp is close to 0. Hence, it should be scaled by an exponential decay annealing function $w_a(t) = exp(-0.005 * t)$. The final loss to train the geometry generation module is $L_{geo} = L_{diff} + w_a(t)L_{sil}$.

Appearance Generation Module. The third diffusion stage is to generate appearance features for a given point scaffold and a conditioned CLIP latent vector. The input to this denoising model is 3D noises sampled from a standard multivariate Gaussian distribution $(Batch, Numberofpoints, 8)$. It shares the same Set Transformer architecture with the geometry denoiser. For the CLIP latent vector and the time embedding, it also adopts adaptive group normalization to implement condition injections. Different from the geometry diffusion model, it requires a point cloud scaffold to be a condition for guiding appearance generation. In the training stage, the point cloud comes from the dataset, but in the inference stage, it can be generated by the above geometry diffusion model. We concatenate the point cloud condition with the input noise along the last dimension in the input layer. Thereafter, the input to the appearance denoiser changes to $(Batch, Numberofpoints, 11)$. After the appearance features are generated, they can be concatenated with the corresponding geometry features and the input point cloud scaffold to recover an intact neural point representation. This representation can be used to synthesize images in arbitrary views with provided camera parameters. Therefore, similar to the silhouette loss in training the geometry diffusion model, we adopt the rendering image loss by computing the L2 distance between the rendered images and the groundtruth. This loss is described in Equation 12 where the $R(x, s)$ means the rendering function to render the neural radiance field x from the provided camera parameter s. The $\bar{x}_0$ is derived by Equation 11 as well. In addition, inspired by [WCH*22], the CLIP loss is employed to improve the semantic correctness of the generated samples by Equation 13. The last two auxiliary losses are multiplied by the above exponential decay weights because both of them utilize the roughly approximated $\bar{x}_0$. Thus the total loss is described as $L_{appe} = L_{diff} + w_a(t)(L_{rend} + \gamma L_{clip})$.

$$L_{rend} = ||R(\bar{x}_0, s) - I_{gt}||_2^2 \tag{12}$$

$$L_{clip} = Dist_{clip}(R(\bar{x}_0, s) - I_{gt}) \tag{13}$$

5.2. Inference paradigm for manipulations.

After the above two generative diffusion models for geometry and appearance are trained, we simply conduct shape or appearance swaps in a two-step manner without any additional setting. The only thing to control this pipeline is to change the CLIP conditions. Figure 3 clearly illustrates the whole process for manipulating neural scenes by changing the CLIP condition for each module. In the first step, our approach generates the point cloud and associated geometry neural features based on the reference images that we expect to follow its shape. The "points with geometry features" in this figure represent the generation results of the geometry diffusion model, in which the point scaffold will be concatenated with a 3D noise matrix and input to the next module. In contrast to the training stage, the appearance CLIP condition is not identical to the geometry condition but is replaced by other reference images or texts that users prefer to maintain the appearance style. As indicated in this figure, the shape description "chair with arms" is maintained while the original blue-style color is transferred to the grey appearance from the second exemplar image.

It can be experimentally observed that even if our method is not

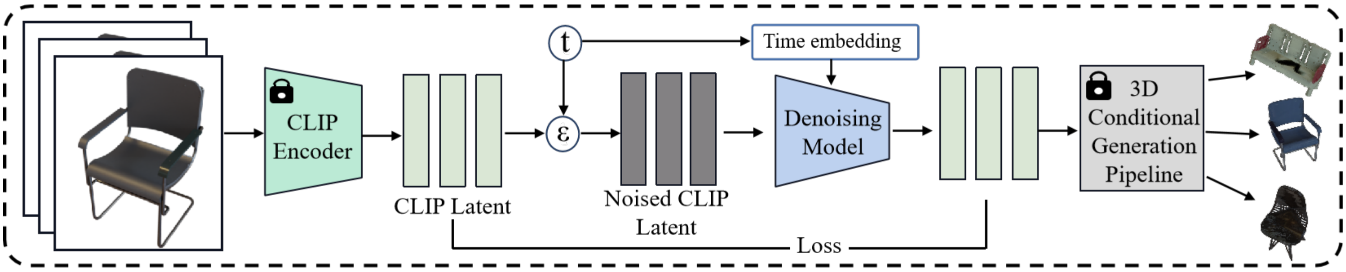

Figure 4: *Illustration of the generative head. A simple latent diffusion model to generate vectors in CLIP latent.*

trained on different shape-appearance pairs, it can bridge the high-level semantic correspondence between the reference conditions and the generation results. This is natural because the pretrained CLIP encodes multimodal data with high-level semantic meanings. In other words, we do not need the paired groundtruth 3D models to train the model but rely on the semantic consistency introduced by the CLIP. In the end, the newly produced sample will be input into our pretrained neural point rendering system (declared in Section 3) to synthesize novel images.

5.3. Generation process.

Since the above-mentioned two-stage diffusion model generates the 3D neural field according to the exemplar prompts, it can also be extended to an unconditional generative model by adding a lightweight auxiliary head module to generate the "condition" for the last two models. In this section, we introduce the generative module as a latent diffusion model to produce CLIP latent (Figure 4). This generative head is trained on the dataset consisting of the precomputed CLIP latent vectors belonging to the specific category. Furthermore, the generated clip latent vector can be considered as the conditions of the other disentangled generative models. For reconstruction, the two models receive the same condition to generate corresponding plausible 3D neural models. Besides, in the unconditional generation settings, the σ in DDIM sampling process is set to 0 to make the conditionally generated sample repeatable and largely agrees to the reference vector. By contrast, if we release the σ constraints, the details of the generated samples become various, and the diversity increases. To clarify the robustness to the σ selection in diffusion model (Equation 8), we test different σ. First we define $\sigma = \eta\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\beta_t$. Then we linearly interpolate η between 0 and 1 with 10 values. It is observed that all metrics in Table 1 fluctuate no more than 0.1. For example if one expects to conduct single view reconstruction by our framework, it is recommended to fix the variance to 0 for better consistency, whereas non-zero variance benefits for generating similar objects with slightly different details.

6. Experiments.

The proposed approach is evaluated in different settings and the results are reported in this section.

Datasets. We evaluate our method on the Photoshape Chairs dataset [PRFS18] and the Amazon Berkeley Objects (ABO) Tables dataset [CGD*22]. We follow the dataset configuration of DiffRF for the Photoshape Chairs dataset. Approximately 15,000 chairs are rendered by Blender Cycles from 200 given viewpoints. For the ABO Tables dataset, we not only contain the samples that are provided 91 renderings with 2-3 different environment maps described in DiffRF but also incorporate the renderings of the ABO 3D models dataset. In detail, we select table examples in the ABO 3D models dataset, and render them with 100 viewpoints from an Archimedean spiral, The final Table dataset contains 2418 tables. Besides, we first pretrain our neural point field on DTU [JDV*14] and Photoshape Chair datasets to obtain a generic NeRF model. Then we run the model on the above two multiview datasets to obtain their neural representations for training our generative model. It is noted that each point cloud scaffold of all objects is downsampled to 8192 points before producing neural representations.

Table 1: *Illustrate the performance for each model. To keep the evaluation metrics in the same order of magnitude, we magnified the FID by a factor of 1000.*

	PhotoShape Chairs		ABO Tables	
Methods	FID ↓	KID ↓	FID ↓	KID ↓
DiffRF	17.71	8.37	27.10	10.10
$Ours_{wo/stage}$	17.97	8.49	26.99	9.84
$Ours_{wo/clip}$	56.46	14.69	47.81	14.15
$Ours_{wo/recon}$	39.82	12.71	33.61	11.28
$Ours_{wo/rend}$	18.68	8.99	28.57	12.00
$Ours_{wo/sil}$	17.26	8.31	27.07	10.19
$Ours_{wo/cliploss}$	18.95	8.79	26.16	10.08
$Ours_{full}$	17.45	8.36	26.08	10.05

Implementation details. The backbone Set Transformer contains 16 layers each with 8 attention heads for both neural point generative models in the manipulation module. The hidden feature vector is 512 dimensions and the post Layer Normalization layers are replaced with Adaptive Group Normalization layers which have 8 groups for incorporating CLIP and timestamp conditions. On the other hand, the latent diffusion model to generate CLIP latent vectors of specific categories is a simple 1D UNet with 1024 hidden features.

6.1. Unconditional Generation Results.

We compare our method with the state-of-the-art diffusion-based NeRF synthesis method DiffRF on the two datasets. Both works require pre-processing of the image collections to create a radiance

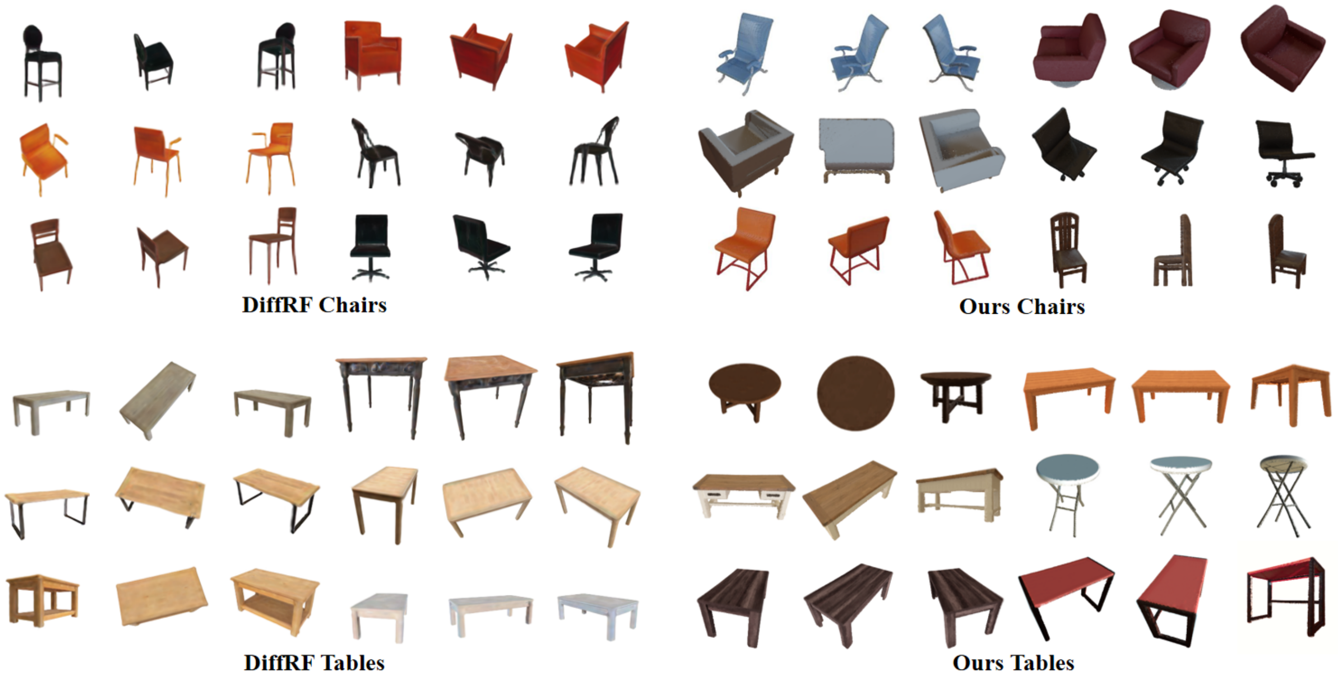

Figure 5: *Qualitative comparison between DiffRF and our method on the PhotoShape Chairs and ABO Tables datasets.*

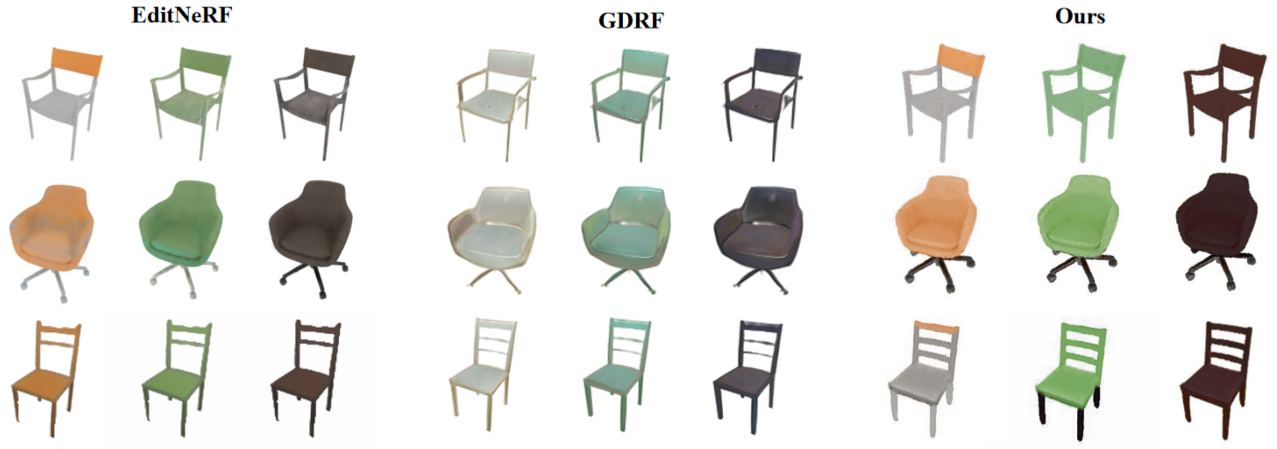

Figure 6: *Qualitative comparison of the shape and texture transfer results of EditNeRF, GDRF, and our method on the PhotoShape Chairs dataset.*

field for each example, but our pipeline needs significantly less time due to our generic radiance field representation. We evaluate the quality of generated radiance field by assessing the quality of images that are obtained by rendering the radiance field. We compute Fr´echet Inception Distance [HRU*17] (FID) and Kernel Inception Distance [BSAG18] (KID) over the 256 × 256 resolution images. Additionally, we also evaluate degraded versions of our method, including 1. a one-stage model to simultaneously generate shape and appearance; 2. the staged diffusion model without CLIP condition. 3. the CLIP module is replaced with an AutoEncoder and trained from scratch 4. without using rendering loss 5. without using silhouette loss 6. without using CLIP loss. Table 1 illustrates the performance for each model. Clearly, the full model delivers the best performance. The staged model without CLIP fails to model the data distribution that highlights the significance of CLIP condition in our method. The unified model achieves comparable results, but it sacrifices the ability to manipulate objects. This also indicates the effectiveness of our neural point field. In addition, the rendering and silhouette losses also improve the performance. Even if the CLIP loss only brings limited improvement in performance, we observed that it benefits the stability of training. We additionally evaluate the Coverage Score (COV) and Minimum Matching Distance (MMD) between our full model and DiffRF. The Coverage Score measures the diversity, while MMD assesses the quality. Fol-

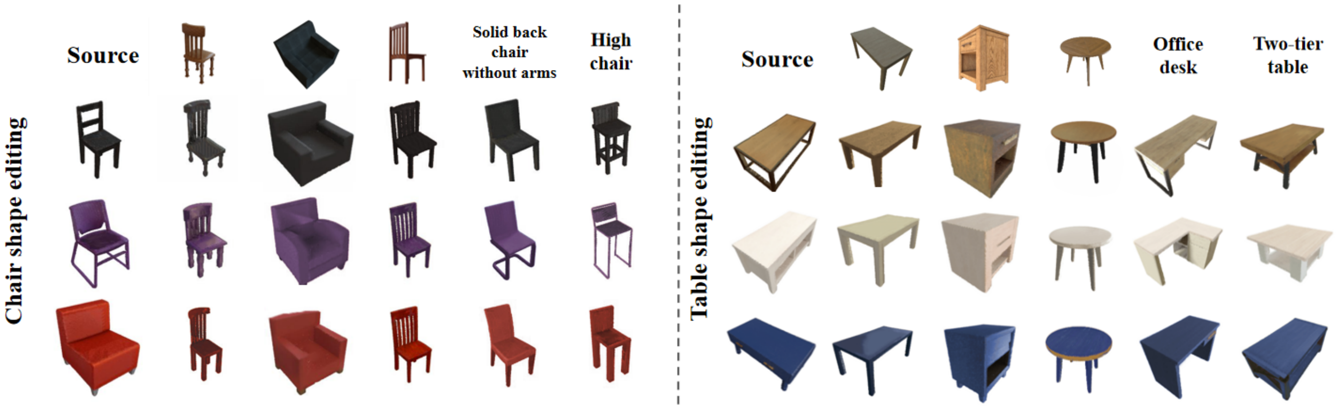

Figure 7: *Example of **shape editing** on the PhotoShape Chairs and ABO Tables datasets with detached manipulated shapes and textures.*

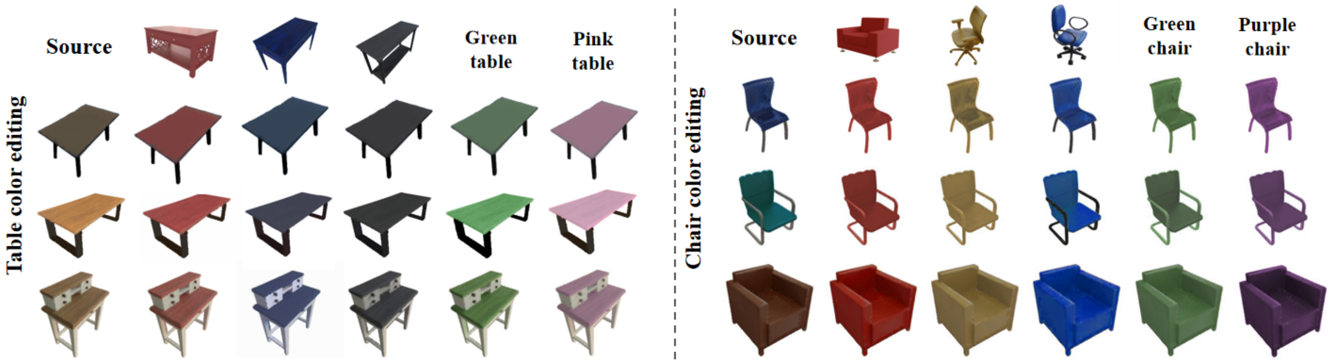

Figure 8: *Example of **color editing** on the PhotoShape Chairs and ABO Tables datasets with detached manipulated shapes and textures.*

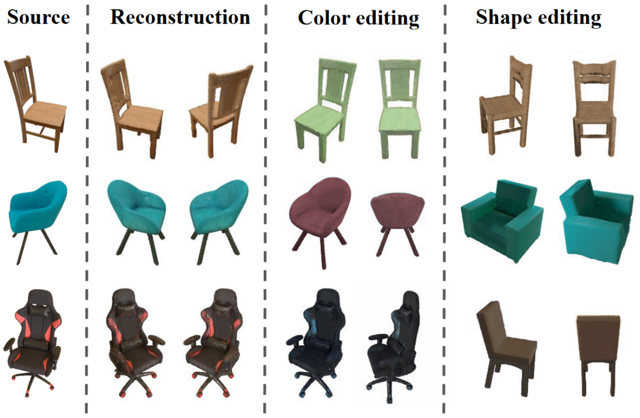

Figure 9: *Real image reconstruction and disentangled manipulation for shape and appearance in our method.*

lowing DiffRF, we use Chamfer Distance in the two metrics. The results are compared in Table 2. Figure 5 qualitatively indicates the comparison between DiffRF and ours. While DiffRF delivers good results for both shape and appearance, it shows limited diversity of its generation. As our generative processes are based on the sampling of CLIP condition vector, this way has more robust guidance than sampling from normal noises.

Table 2: *Illustrate the COV and MMD metrics between the baseline and our full model.*

	PhotoShape Chairs		ABO Tables	
Methods	COV ↑	MMD ↓	COV ↑	MMD ↓
DiffRF	59.20	4.42	61.60	7.64
$Ours_{full}$	59.56	4.01	66.61	7.52

6.2. Disentangled Manipulation Results.

The proposed method is capable of manipulating objects under the guidance of images/texts in a disentangled manner. This can be simply achieved by replacing different prompts in the disentangled generation pipeline (Figure 3). In contrast, GAN-based methods have to project the reference images to their original sampling distributions or train additional networks to predict the inverse processes. Due to the discrepancy between the original normal distribution and the output distribution of GANs, the inverse GAN process has always been complex and intractable. Our diffusion-based model considers this as a forward process thereby bypassing the inversion. In this section, we compare the editability of our model with the EditNerf and GDRF, which are optimization-based and generative-based models respectively. Figure 6 includes the shape and texture transfer of the above three methods. The diagonal elements are their original. It is clear that our method gives more

Table 3: *Performance comparisons between CLIP and an Autoencoder in the appearance module.*

	PSNR ↑	SSIM ↑	LPIPS ↓
CLIP	24.71	0.93	0.078
AE	19.52	0.76	0.15

correct texture transfer with regard to semantic correspondence. See the first and third example, EditNeRF wrongly recognize the color of the chair seat and chair back, it tends to mix the two color and transfer them to the target object. In contrast, our method semantically distinguishes the texture of different parts of the chair and successfully transfers it to the corresponding parts of the target chair. This is because the CLIP latent encodes high-level semantic information Compared with the GDRF, our method provides more realistic geometry.

We further show more examples of disentangled manipulation shape and texture on the two datasets in Figure 7 and Figure 8. The results show a clear separation of shape and appearance on account of our explicitly disentangled representation. Besides, it is noted from the third experiment in Table 1 that CLIP shows significance in the reconstruction. We modified this experiment to analyze the effect of CLIP on appearance independently, only replacing the CLIP encoder in the appearance generation module with an Autoencoder. We test three reconstruction metrics in Table 3 and the results illustrate the great assistance of CLIP in appearance reconstruction.

6.3. Real Image Reconstruction and Manipulation.

We test our model on real-world images for reconstruction and manipulation. Since our model is trained with white background for all examples, the colorful backgrounds of real-world images are removed before input to CLIP. In addition, we fixed the variance in DDIM as zero before reconstructing the neural point field for better consistency. Figure 9 illustrates that our method can faithfully reconstruct the entire 3D object as the neural point field merely based on a single realistic image and synthesize novel views from arbitrary viewpoints. Besides, we can further manipulate the shape and texture of the real object by simply replacing the CLIP conditions. Different from GAN-based methods, our goal is to directly edit the 3D representations rather than the images, but the edited images can also be obtained indirectly by accessing the 3D scenes based on associated camera poses.

6.4. Efficiency.

Commonly used volumetric radiance field in previous works often has a resolution of $32^3 = 32768$, our method contains 8192 neural points. Thus our utilization rate of each neural element is higher than the counterparts. Smaller resolutions also lead to faster convergence. We trained our model on four NVIDIA 3090 GPUs with a convergence time of 22 hours, which is shorter than the 38 hours of GDRF on the same device. The generation process takes only 56 seconds, which is faster than DiffRF's 1 minute and 47 seconds.

6.5. Other categories.

To clarify the generalizability of the proposed approach, we further retrained our model and DiffRF on two complex categories from ShapeNet and Objaverse datasets, "car" and "airplane". The quantitative comparisons are listed in Table 4. It is noted that the superiority of our model does not fall into specific categories.

Table 4: *Illustrate the performance of the other two categories to explain the generalization of the proposed approach. The FID is magnified by a factor of 1000.*

	Car		Airplane	
Methods	FID ↓	KID ↓	FID ↓	KID ↓
DiffRF	26.20	11.03	21.14	9.85
$Ours_{full}$	23.20	10.77	19.36	9.05

7. Conclusion.

We propose a purely diffusion-based three-stage framework for unifying generation and disentangled editing tasks that directly involve 3D neural representations. To the best of our knowledge, our method is the first attempt to separately manipulate neural radiance field by diffusion models, and the first to use neural point field as the 3D representation for NeRF-related generative tasks. The presented generic neural point field both improves the efficiency and reduces the difficulties of 3D datasets conversion. Our model considers manipulation tasks as a conditional generation process. We leverage the high-level semantic correspondence naturally encoded in CLIP latent to edit the shape and texture. The proposed method is evaluated on two datasets, achieves better performance than SOTA methods for both generation and editing tasks, and releases the potential of diffusion model in the new regions.

8. Limitation.

While the proposed method shows promising results on both generation and manipulation tasks, several limitations remain. At present, this method only supports category-specific objects due to the lack of large 3D datasets. Knowledge distillation or collecting plenty of 3D data might help handle this challenge. Additionally, the method only deals with 3D objects without colorful backgrounds. In the future, one possible solution is to model the background via a regular neural grid and train another neural block to handle backgrounds independently.

References

[AHS*23] ACHLIOPTAS P., HUANG I., SUNG M., TULYAKOV S., GUIBAS L.: Shapetalk: A language dataset and framework for 3d shape edits and deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 12685–12694. 1

[AJH*21] AUSTIN J., JOHNSON D. D., HO J., TARLOW D., VAN DEN BERG R.: Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems 34* (2021), 17981–17993. 3

[ALF22] AVRAHAMI O., LISCHINSKI D., FRIED O.: Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 18208–18218. 1, 2, 3

[BKC*22] BREMPONG E. A., KORNBLITH S., CHEN T., PARMAR N., MINDERER M., NOROUZI M.: Denoising pretraining for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 4175–4186. 3

[BNX*23] Bao F., Nie S., Xue K., Li C., Pu S., Wang Y., Yue G., Cao Y., Su H., Zhu J.: One transformer fits all distributions in multi-modal diffusion at scale. *arXiv preprint arXiv:2303.06555* (2023). 2, 3

[BRV*21] Baranchuk D., Rubachev I., Voynov A., Khrulkov V., Babenko A.: Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126* (2021). 3

[BSAG18] Bińkowski M., Sutherland D. J., Arbel M., Gretton A.: Demystifying mmd gans. *arXiv preprint arXiv:1801.01401* (2018). 8

[CCS*19] Chen K., Choy C. B., Savva M., Chang A. X., Funkhouser T., Savarese S.: Text2shape: Generating shapes from natural language by learning joint embeddings. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14* (2019), Springer, pp. 100–116. 3

[CGD*22] Collins J., Goel S., Deng K., Luthra A., Xu L., Gundogdu E., Zhang X., Vicente T. F. Y., Dideriksen T., Arora H., et al.: Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 21126–21136. 7

[CHIS23] Croitoru F.-A., Hondru V., Ionescu R. T., Shah M.: Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023). 3

[CJ23] Cao A., Johnson J.: Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 130–141. 2

[CLC*22] Chan E. R., Lin C. Z., Chan M. A., Nagano K., Pan B., De Mello S., Gallo O., Guibas L. J., Tremblay J., Khamis S., et al.: Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 16123–16133. 1, 3

[CLH*22] Chen D., Liu Y., Huang L., Wang B., Pan P.: Geoaug: Data augmentation for few-shot nerf with geometry constraints. In *European Conference on Computer Vision* (2022), Springer, pp. 322–337. 2

[CMK*21] Chan E. R., Monteiro M., Kellnhofer P., Wu J., Wetzstein G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 5799–5809. 1, 3

[CSL*23] Chen D. Z., Siddiqui Y., Lee H.-Y., Tulyakov S., Niessner M.: Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396* (2023). 2, 3

[CSY22] Chung H., Sim B., Ye J. C.: Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 12413–12422. 3

[CXZ*21] Chen A., Xu Z., Zhao F., Zhang X., Xiang F., Yu J., Su H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14124–14133. 2

[CY22] Chung H., Ye J. C.: Score-based diffusion models for accelerated mri. *Medical image analysis 80* (2022), 102479. 3

[DBS*21] DeVries T., Bautista M. A., Srivastava N., Taylor G. W., Susskind J. M.: Unconstrained scene generation with locally conditioned radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14304–14313. 1

[DMH21] Daniels M., Maunu T., Hand P.: Score-based generative neural networks for large-scale optimal transport. *Advances in neural information processing systems 34* (2021), 12955–12965. 1, 3

[FZC*22] Fu X., Zhang S., Chen T., Lu Y., Zhu L., Zhou X., Geiger A., Liao Y.: Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *2022 International Conference on 3D Vision (3DV)* (2022), IEEE, pp. 1–11. 1

[GCB*22] Gu S., Chen D., Bao J., Wen F., Zhang B., Chen D., Yuan L., Guo B.: Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 10696–10706. 1, 2, 3

[GKJ*21] Garbin S. J., Kowalski M., Johnson M., Shotton J., Valentin J.: Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14346–14355. 2

[GLF*22] Gong S., Li M., Feng J., Wu Z., Kong L.: Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933* (2022). 3

[GLWT21] Gu J., Liu L., Wang P., Theobalt C.: Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985* (2021). 1, 3

[GMJS22] Graikos A., Malkin N., Jojic N., Samaras D.: Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems 35* (2022), 14715–14728. 3

[GPAM*20] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.: Generative adversarial networks. *Communications of the ACM 63*, 11 (2020), 139–144. 3

[GSW*22] Gao J., Shen T., Wang Z., Chen W., Yin K., Li D., Litany O., Gojcic Z., Fidler S.: Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems 35* (2022), 31841–31854. 3

[HAZ*22a] Huang I., Achlioptas P., Zhang T., Tulyakov S., Sung M., Guibas L.: Ladis: Language disentanglement for 3d shape editing. *arXiv preprint arXiv:2212.05011* (2022). 1

[HAZ*22b] Huang I., Achlioptas P., Zhang T., Tulyakov S., Sung M., Guibas L.: Ladis: Language disentanglement for 3d shape editing. *arXiv preprint arXiv:2212.05011* (2022). 3

[HCO*23] Höllein L., Cao A., Owens A., Johnson J., Niessner M.: Text2room: Extracting textured 3d meshes from 2d text-to-image models. *arXiv preprint arXiv:2303.11989* (2023). 2, 3

[HCS*22] Ho J., Chan W., Saharia C., Whang J., Gao R., Gritsenko A., Kingma D. P., Poole B., Norouzi M., Fleet D. J., et al.: Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303* (2022). 3

[HJA20] Ho J., Jain A., Abbeel P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems 33* (2020), 6840–6851. 3, 4

[HKT22] Han X., Kumar S., Tsvetkov Y.: Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. *arXiv preprint arXiv:2210.17432* (2022). 3

[HMR19] Henzler P., Mitra N. J., Ritschel T.: Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 9984–9993. 1, 3

[HNM*22] Harvey W., Naderiparizi S., Masrani V., Weilbach C., Wood F.: Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems 35* (2022), 27953–27965. 3

[HRU*17] Heusel M., Ramsauer H., Unterthiner T., Nessler B., Hochreiter S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems 30* (2017). 8

[JA21] Jang W., Agapito L.: Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12949–12958. 2

[JDV*14] Jensen R., Dahl A., Vogiatzis G., Tola E., Aanæs H.: Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 406–413. 7

[JLF22] Johari M. M., Lepoittevin Y., Fleuret F.: Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 18365–18375. 2

[JSJ*21] Jo K., Shim G., Jung S., Yang S., Choo J.: Cg-nerf: Conditional generative neural radiance fields. *arXiv preprint arXiv:2112.03517* (2021). 1

[JTA21] Jain A., Tancik M., Abbeel P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5885–5894. 2

[KAAL22] Karras T., Aittala M., Aila T., Laine S.: Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems 35* (2022), 26565–26577. 3

[KGY*22] Kundu A., Genova K., Yin X., Fathi A., Pantofaru C., Guibas L. J., Tagliasacchi A., Dellaert F., Funkhouser T.:. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 12871–12881. 1

[KSH22] Kim M., Seo S., Han B.: Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 12912–12921. 1

[KZL*23] Kawar B., Zada S., Lang O., Tov O., Chang H., Dekel T., Mosseri I., Irani M.: Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 6007–6017. 1, 3

[LDR*22] Lugmayr A., Danelljan M., Romero A., Yu F., Timofte R., Van Gool L.: Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 11461–11471. 1, 3

[LGT*23] Lin C.-H., Gao J., Tang L., Takikawa T., Zeng X., Huang X., Kreis K., Fidler S., Liu M.-Y., Lin T.-Y.: Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 300–309. 2, 3

[LH21] Luo S., Hu W.: Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 2837–2845. 3

[LLK*19] Lee J., Lee Y., Kim J., Kosiorek A., Choi S., Teh Y. W.: Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning* (2019), PMLR, pp. 3744–3753. 5

[LLW*23] Li S., Li H., Wang Y., Liao Y., Yu L.: Steernerf: Accelerating nerf rendering via smooth viewpoint trajectory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 20701–20711. 2

[LLZ*22] Li C., Li S., Zhao Y., Zhu W., Lin Y.: Rt-nerf: Real-time on-device neural radiance fields towards immersive ar/vr rendering. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design* (2022), pp. 1–9. 2

[LNSW21] Li Z., Niklaus S., Snavely N., Wang O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 6498–6508. 2

[LPL*22] Liu Y., Peng S., Liu L., Wang Q., Wang P., Theobalt C., Zhou X., Wang W.: Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 7824–7833. 2

[LTG*22] Li X., Thickstun J., Gulrajani I., Liang P. S., Hashimoto T. B.: Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems 35* (2022), 4328–4343. 3

[LWYL22] Liu X., Wu L., Ye M., Liu Q.: Let us build bridges: Understanding and extending diffusion generative models. *arXiv preprint arXiv:2208.14699* (2022). 3

[LZZ*21] Liu S., Zhang X., Zhang Z., Zhang R., Zhu J.-Y., Russell B.: Editing conditional radiance fields. 5773–5783. 2

[MHS*21] Meng C., He Y., Song Y., Song J., Wu J., Zhu J.-Y., Ermon S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073* (2021). 1, 3

[MSP*23] Müller N., Siddiqui Y., Porzi L., Bulo S. R., Kontschieder P., Niessner M.: Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 4328–4338. 3

[MST*21] Mildenhall B., Srinivasan P. P., Tancik M., Barron J. T., Ramamoorthi R., Ng R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. 1, 2, 3

[NDR*21] Nichol A., Dhariwal P., Ramesh A., Shyam P., Mishkin P., McGrew B., Sutskever I., Chen M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021). 1, 2, 3

[NG21] Niemeyer M., Geiger A.: Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 11453–11464. 2

[NJD*22] Nichol A., Jun H., Dhariwal P., Mishkin P., Chen M.: Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751* (2022). 5

[NSL*23] Ni H., Shi C., Li K., Huang S. X., Min M. R.: Conditional image-to-video generation with latent flow diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 18444–18455. 2

[PCPMMN21] Pumarola A., Corona E., Pons-Moll G., Moreno-Noguer F.: D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 10318–10327. 2

[PGZ*22a] Peng C., Guo P., Zhou S. K., Patel V. M., Chellappa R.: Towards performant and reliable undersampled mr reconstruction via diffusion model sampling. 623–633. 3

[PGZ*22b] Peng C., Guo P., Zhou S. K., Patel V. M., Chellappa R.: Towards performant and reliable undersampled mr reconstruction via diffusion model sampling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2022), Springer, pp. 623–633. 3

[PJBM22] Poole B., Jain A., Barron J. T., Mildenhall B.: Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022). 2, 3

[PRFS18] Park K., Rematas K., Farhadi A., Seitz S. M.: Photoshape: Photorealistic materials for large-scale shape collections. *arXiv preprint arXiv:1809.09761* (2018). 7

[PVG*21] Popov V., Vovk I., Gogoryan V., Sadekova T., Kudinov M.: Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning* (2021), PMLR, pp. 8599–8608. 2, 3

[QCZ*23] Qi C., Cun X., Zhang Y., Lei C., Wang X., Shan Y., Chen Q.: Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv preprint arXiv:2303.09535* (2023). 3

[RBL*22] Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B.: High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 10684–10695. 1, 3

[RKP*23] Raj A., Kaza S., Poole B., Niemeyer M., Ruiz N., Mildenhall B., Zada S., Aberman K., Rubinstein M., Barron J., et al.: Dreambooth3d: Subject-driven text-to-3d generation. *arXiv preprint arXiv:2303.13508* (2023). 2, 3

[RLJ*23] RUIZ N., LI Y., JAMPANI V., PRITCH Y., RUBINSTEIN M., ABERMAN K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 22500–22510. 1, 2, 3

[SDWMG15] SOHL-DICKSTEIN J., WEISS E., MAHESWARANATHAN N., GANGULI S.: Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning* (2015), PMLR, pp. 2256–2265. 3

[SE19] SONG Y., ERMON S.: Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems 32* (2019). 3

[SHC*22] SAHARIA C., HO J., CHAN W., SALIMANS T., FLEET D. J., NOROUZI M.: Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*, 4 (2022), 4713–4726. 1, 3

[SLNG20] SCHWARZ K., LIAO Y., NIEMEYER M., GEIGER A.: Graf: Generative radiance fields for 3d-aware image synthesis. vol. 33, pp. 20154–20166. 2

[SME20] SONG J., MENG C., ERMON S.: Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020). 4

[SPH*22] SINGER U., POLYAK A., HAYES T., YIN X., AN J., ZHANG S., HU Q., YANG H., ASHUAL O., GAFNI O., ET AL.: Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792* (2022). 2, 3

[SSC22] SUN C., SUN M., CHEN H.-T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5459–5469. 2

[SSDK*20] SONG Y., SOHL-DICKSTEIN J., KINGMA D. P., KUMAR A., ERMON S., POOLE B.: Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020). 3

[TRG*22] TEVET G., RAAB S., GORDON B., SHAFIR Y., COHEN-OR D., BERMANO A. H.: Human motion diffusion model. *arXiv preprint arXiv:2209.14916* (2022). 2, 3

[TTG*21] TRETSCHK E., TEWARI A., GOLYANIK V., ZOLLHÖFER M., LASSNER C., THEOBALT C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12959–12970. 2

[TZFR23] TURKI H., ZHANG J. Y., FERRONI F., RAMANAN D.: Suds: Scalable urban dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 12375–12385. 2

[VWG*22] VAHDAT A., WILLIAMS F., GOJCIC Z., LITANY O., FIDLER S., KREIS K., ET AL.: Lion: Latent point diffusion models for 3d shape generation. vol. 35, pp. 10021–10039. 2, 3

[WCH*22] WANG C., CHAI M., HE M., CHEN D., LIAO J.: Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 3835–3844. 1, 2, 6

[WDL*23] WANG H., DU X., LI J., YEH R. A., SHAKHNAROVICH G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 12619–12629. 2, 3

[WDY*22] WANG Z., DENG Y., YANG J., YU J., TONG X.: Generative deformable radiance fields for disentangled image synthesis of topology-varying objects. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 431–442. 2

[WLC*23] WANG P., LIU Y., CHEN Z., LIU L., LIU Z., KOMURA T., THEOBALT C., WANG W.: F2-nerf: Fast neural radiance field training with free camera trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 4150–4159. 2

[WWG*21] WANG Q., WANG Z., GENOVA K., SRINIVASAN P. P., ZHOU H., BARRON J. T., MARTIN-BRUALLA R., SNAVELY N., FUNKHOUSER T.: Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 4690–4699. 2

[XWC*23] XU J., WANG X., CHENG W., CAO Y.-P., SHAN Y., QIE X., GAO S.: Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 20908–20918. 2, 3

[XZL*23] XIE S., ZHANG Z., LIN Z., HINZ T., ZHANG K.: Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 22428–22437. 1, 3

[YBZ*22] YANG B., BAO C., ZENG J., BAO H., ZHANG Y., CUI Z., ZHANG G.: Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *European Conference on Computer Vision* (2022), Springer, pp. 597–614. 2

[YGZ*23] YANG B., GU S., ZHANG B., ZHANG T., CHEN X., SUN X., CHEN D., WEN F.: Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 18381–18391. 1, 3

[YPW23] YANG J., PAVONE M., WANG Y.: Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 8254–8263. 2

[YSL*22] YUAN Y.-J., SUN Y.-T., LAI Y.-K., MA Y., JIA R., GAO L.: Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 18353–18364. 2

[ZCP*22] ZHANG M., CAI Z., PAN L., HONG F., GUO X., YANG L., LIU Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001* (2022). 3

[ZDW21] ZHOU L., DU Y., WU J.: 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5826–5835. 3

[ZHG*23] ZHANG Z., HAN L., GHOSH A., METAXAS D. N., REN J.: Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 6027–6037. 1, 3

[ZLW*23] ZHU Y., LI Z., WANG T., HE M., YAO C.: Conditional text image generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 14235–14245. 2

[ZXNT21] ZHOU P., XIE L., NI B., TIAN Q.: Cips-3d: A 3d-aware generator of gans based on conditionally-independent pixel synthesis. *arXiv preprint arXiv:2110.09788* (2021). 1, 3