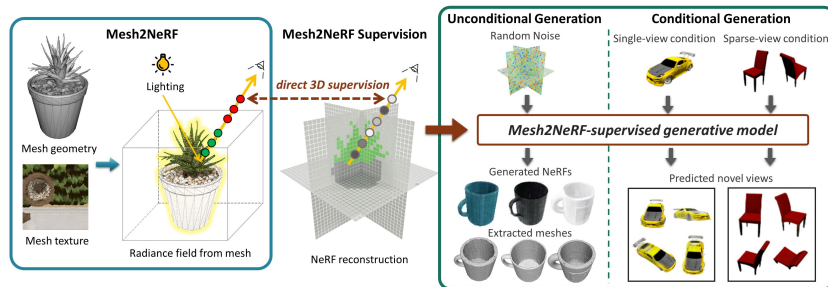# Mesh2NeRF: Direct Mesh Supervision for Neural Radiance Field Representation and Generation

Yujin Chen[1], Yinyu Nie[1], Benjamin Ummenhofer[2], Reiner Birkl[2],
Michael Paulitsch[2], Matthias Müller[2], and Matthias Nießner[1]

[1] Technical University of Munich
[2] Intel Labs

Fig. 1: We propose Mesh2NeRF, a novel method for extracting ground truth radiance fields directly from 3D textured meshes by incorporating mesh geometry, texture, and environment lighting information. Mesh2NeRF serves as direct 3D supervision for neural radiance fields, offering a comprehensive approach to leveraging mesh data for improving novel view synthesis performance. Mesh2NeRF can function as supervision for generative models during training on mesh collections, advancing various 3D generation tasks, including unconditional and conditional generation.

Abstract. We present Mesh2NeRF, an approach to derive ground-truth radiance fields from textured meshes for 3D generation tasks. Many 3D generative approaches represent 3D scenes as radiance fields for training. Their ground-truth radiance fields are usually fitted from multi-view renderings from a large-scale synthetic 3D dataset, which often results in artifacts due to occlusions or under-fitting issues. In Mesh2NeRF, we propose an analytic solution to directly obtain ground-truth radiance fields from 3D meshes, characterizing the density field with an occupancy function featuring a defined surface thickness, and determining view-dependent color through a reflection function considering both the mesh and environment lighting. Mesh2NeRF extracts accurate radiance fields which provides direct supervision for training generative NeRFs and single scene representation. We validate the effectiveness of Mesh2NeRF across various tasks, achieving a noteworthy 3.12dB improvement in PSNR for view synthesis in single scene representation on the ABO dataset, a 0.69 PSNR enhancement in the single-view conditional generation of ShapeNet Cars, and notably improved mesh extraction from NeRF in the unconditional generation of Objaverse Mugs.

Keywords: Radiance Field Supervision · NeRF Generation · Mesh Prior

1   Introduction

3D virtual content generation has garnered increased attention within the domains of computer vision and graphics. This surge is fueled by advancements in large-scale datasets [11, 19, 21–23] and generative models [9, 10, 45, 50, 51], fostering significant growth in the 3D asset creation industry. Classical 3D representations, including point clouds, meshes, voxel grids, SDF, etc., have seamlessly integrated into generative models, yielding promising results in generating 3D shapes [30, 34, 46, 53, 64]. Radiance fields have emerged as a powerful learning representation for 3D reconstruction and generation, as evidenced by their effectiveness in various studies [5, 15, 28, 31, 38, 39, 55, 62].

Radiance fields possess significant potential as high-quality 3D generative representations, but ground truth radiance fields are required as training samples for generative models. However, obtaining GT radiance fields is extremely challenging. To overcome the lack of direct 3D radiance field supervision, some recent approaches utilize 2D generative models [8, 43, 50, 66] to generate multiview posed images [27, 56]. Subsequently, these 2D generations are fitted into NeRFs or 4D grids (RGB + opacity) to obtain textured 3D outputs [32, 33]. Nevertheless, these methods rely on the quality and consistency of upstream 2D multi-view generation, posing challenges in ensuring high-quality 3D outputs.

Some methods directly train a 3D generative model on native 3D data [18, 35, 41, 52], leveraging inherent 3D supervision for natural view consistency. However, these methods commonly encounter challenges in jointly recovering geometry and texture due to the diversity of shape and appearance distributions. Most approaches focus on generating geometry [16, 42] and then proceed through additional stages to obtain appearance [14]. Recent prevalent approaches involve using 2D supervision, which supervises differentiable renderings from 3D generations using the GT mesh renderings from 3D meshes. For example, [15, 39] learn NeRF generation from mesh collections like ShapeNet. However, they require rendering dense multi-views from each mesh, learning respective shape NeRF representation for each training sample from the rendered images, and training the generative diffusion model to model the distribution of NeRF representations for all samples. Despite achieving reasonable results, we argue that the rendering process is redundant and also introduces information loss. In addition, relying on 2D supervision from multi-view renderings also introduce inaccurate reconnstructions, where neural volume rendering is employed to integrate ray colors and calculate rendering loss from pixel colors. We contend that this loss function offers weak supervision, as each pixel color is tasked with supervising the learning of all points' density and color in a ray. Particularly when faced with few or imbalanced view observations, this methodology struggles to synthesize an accurate radiance field.

In this work, we aim to synthesize textured shapes while inheriting the advantages of 3D mesh data by denoising the radiance field with guidance from ray color and density values extracted from 3D meshes. Our method employs individual supervision for each ray point, overseeing both density and color. To facilitate this, we introduce a module named **Mesh2NeRF**, designed to extract

theoretically GT sampled ray points from meshes. By utilizing this module to supervise a generative model, such as a modern triplane-based NeRF Diffusion model [15], our approach exhibits superior performance in both unconditional and conditional 3D textured shape generation tasks. In summary, our contributions are:

1. We propose Mesh2NeRF, a theoretical derivation to directly create a radiance field from mesh data. Employing Mesh2NeRF to convert a mesh to a radiance field eliminates the need for rendering multi-view images, avoiding typical imperfections in multi-view reconstruction.
2. We show how Mesh2NeRF can be employed as direct supervision in training generative models with 3D data, especially in applications such as conditional and unconditional NeRF generation.

2 Related Work

NeRF as 3D representation. Introduced by [38] in 2020, NeRF has become a prominent method for representing scenes. Advances in works like cone tracing with positional encoding [6] and proposal multi-layer perceptron (MLP) [7] have enhanced neural representation and rendering across various scenarios [7, 37, 37, 44, 60]. While NeRF has made substantial progress and unlocked numerous applications, there remain significant differences between its representation and the mesh representation widely utilized in traditional computer graphics. Bridging the gap between these two representations is a challenge, and existing methods have explored incorporating meshes into the rendering process to expedite the rendering [58, 65]. NeRF2Mesh [58] focuses on reconstructing surface meshes from multi-view images for applications such as scene editing and model composition. We posit that leveraging existing mesh collections [11, 19, 22, 23], often featuring high-quality data crafted by artists [1, 2], can enhance NeRF/3D applications. To achieve this, we propose Mesh2NeRF to directly obtain a radiance field from a textured 3D mesh. By using Mesh2NeRF as direct mesh supervision in NeRF tasks, we aim to harness mesh data to enhance applications utilizing NeRF as the 3D representation.

NeRF supervision. The original NeRF is trained using a pixel loss, where rays sampled from the NeRF are integrated to colors supervised by ground-truth pixel colors. However, this approach requires substantial multi-view coverage, as each pixel color is utilized to supervise all 3D points sampled along a ray emitted from the pixel. To mitigate ambiguity, various methods introduce depth priors as additional supervision to inform networks about scene surfaces [24, 49, 63]. In addition to addressing ambiguity, the inference of scene surfaces is often employed for ray termination to prevent further point sampling [48]. In contrast, our approach, Mesh2NeRF, accepts textured meshes as input and employs an analytical solution to generate radiance fields. This derived radiance field proves effective in various NeRF tasks, encompassing the fitting of single scenes and the training of NeRF generative models.

NeRF generation with diffusion models. Recently, NeRF representations have gained popularity in shape generation, especially within diffusion-based methods [20]. In the realm of diffusion models for 3D shape generation, approaches can be broadly categorized into two groups: those extending 2D diffusion priors to the 3D domain [4,29,36,59,61], and those directly applying diffusion processes to 3D data [17, 18, 39, 52]. In essence, 2D-lifted diffusion methods inherit the richness of large 2D diffusion models but may grapple with issues of view consistency, while 3D native data diffusion methods exhibit complementary characteristics. Several diffusion models learn priors from synthetic objects either by learning diffusion weights on pre-calculated NeRF representations [39] or by jointly optimizing diffusion weights with each object's NeRF [15]. However, these approaches heavily rely on multi-view observations rendered from meshes. In contrast, our Mesh2NeRF directly derives analytically correct colors and density values for ray points from meshes. These serve as robust 3D supervision in NeRF generation diffusion models.

3   Method

We provide a brief overview of volume rendering and NeRF in Section 3.1. Following that, we delve into how Mesh2NeRF directly derives radiance field representation from textured meshes in Section 3.2 and elaborate on the application of Mesh2NeRF as direct supervision in NeRFs in Section 3.3. In Section 3.4, we illustrate the application of Mesh2NeRF in NeRF generation diffusion models.

### 3.1   Volume Rendering and NeRFs Revisited

NeRFs model a 3D scene using a neural network to represent the radiance field. This field takes a 3D position $\mathbf{x} = (x, y, z)$ and viewing direction $\mathbf{d} = (\theta, \phi)$, providing the volume density $\sigma(\mathbf{x}) \in \mathbb{R}_0^+$ and RGB color $\mathbf{c}(\mathbf{x}, \mathbf{d})$ at that point. To render the color of a pixel in a target camera view, the 3D position $\mathbf{o}$ of the camera center and the camera viewing direction $\mathbf{d}$ are used to construct the ray $\mathbf{y} = \mathbf{o} + t\mathbf{d}$, with the ray parameter $t \in \mathbb{R}_0^+$. The ray is sampled at N points $\mathbf{y}_i$ with parameter values t_i, where the volume density and color are $\sigma_i = \sigma(\mathbf{y}_i)$ and $\mathbf{c}_i = \mathbf{c}(\mathbf{y}_i, \mathbf{d})$. The pixel color is then approximated by the formula from the original NeRF paper [38, 57]:

$$\hat{\mathbf{C}}(\mathbf{y}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i \tag{1}$$

Here,

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \tag{2}$$

is the accumulated transmittance along the ray $\mathbf{y}$ to the sample i and $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples.
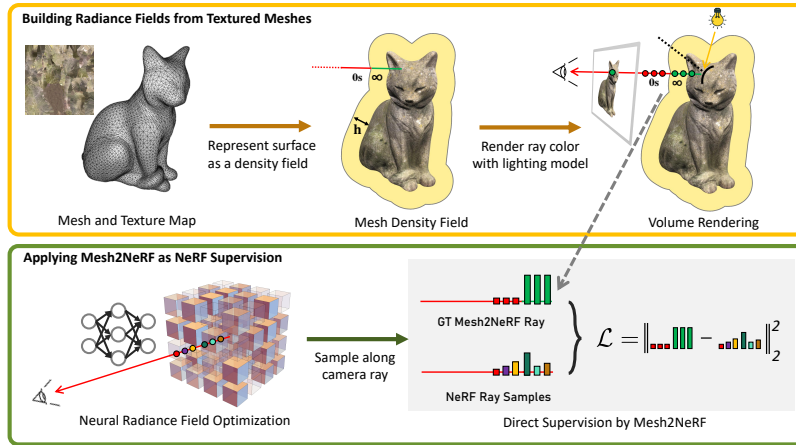
Fig. 2: Our method, illustrated above, constructs a GT radiance field from a textured mesh. Using a surface-based occupancy function with a distance threshold, we model the scene's density field. View-dependent color is then modeled, considering view direction, surface geometry, and light direction. Integrating samples along the camera ray enables accurate volume rendering from our defined radiance field. The bottom part showcases Mesh2NeRF as direct 3D supervision for NeRF tasks, where the density and color values of each ray sample supervise NeRF ray samples during optimization.

During NeRF training, the MLP's weights, encoding scene characteristics, are optimized. The training process aims to optimize MLP weights to match the rendered and GT colors of the rays. With comprehensive scene coverage in the input images, this straightforward process yields MLP weights that precisely represent the 3D volumetric density and appearance of the scene. For novel view synthesis, a corresponding image can be rendered from a given camera pose.

3.2 Radiance Field from Textured Mesh

In the process of constructing a radiance field from a given textured mesh in conjunction with environmental lighting, our objective is to approximate the ideal density field and color field, collectively representing a 3D mesh as a continuous function in $\mathbb{R}^3$. In the ideal scenario, the color along each ray is constant and equal to the color of the hitting point on the mesh. The density, on the other hand, resembles a Dirac delta function, implying that only the mesh surface positions have infinite density values, while the density is zero elsewhere. However, such a density distribution is impractical for training a neural network or for discrete volume rendering with a finite number of samples. We therefore adapt our ideal density and color fields to make them suitable for network training and discrete volume rendering.

Density field of mesh surfaces. Discrete volume rendering is based on Eq. 1, which assumes the density σ and color $\mathbf{c}$ are piecewise constant, as demonstrated

in [57]. We find that the density can be modeled by the top hat functions with a parameter n:

$$\Delta_n(t) = \begin{cases} \frac{n}{2}, & \text{if } |t| < \frac{1}{n} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

These functions provide a density $\sigma(t)$ which is $n\Delta_n(t)$ at every step of the limit process $\lim_{n\to\infty}$ of a Dirac delta function, and are thus piece-wise constant.

The height of the density $\sigma(t)$ near the mesh is $n^2/2$ for the top hat functions, which grows indefinitely during the limit process. However, very large numbers are not adequate for a representation with a neural network. Therefore, we use the alpha values

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i) \tag{4}$$

and reformulate Eqs. 1 and 2 to

$$\hat{C}(\mathbf{y}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i \tag{5}$$

with the accumulated transmittance:

$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{6}$$

The alpha value α_i denotes how much light is contributed by sample i. We refer to the original NeRF paper [57] for more details. The alpha value changes from 0 to 1 when touching the surface and back to 0 after passing through it. Hence, for large n, the density $n\Delta_n(t)$ can approximately be represented by:

$$\alpha = \begin{cases} 1, & \text{if } d < h \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where α is a function of distance to mesh surface d; h is the half of the surface thickness.

Our key insight is that we can use occupancy to represent the alpha value α. From Eq. 5, we can get the conclusion that $\hat{C}(\mathbf{y}) = \alpha_{i_m} \mathbf{c}_{i_m}$, where i_m is the first sampled point intersecting with the surface, $\alpha_{i_m} = 1$, and $\mathbf{c}_{i_m}$ is the color of intersection point from the texture mesh.

Modeling the color field. In Mesh2NeRF, we represent the color of sampled points along each ray as a constant using a BRDF model, e.g., we use the Phong model in our experiments while any BRDF can be applied. For any sampled point on the ray with direction $\mathbf{v}$, its color is defined as the color $\mathbf{c_i}$ of the first hitting point of the ray and mesh surface. In Fig. 3, we qualitatively compare volume rendering outcomes between our defined Mesh2NeRF fields (obtained through Mesh2NeRF analytic solution) with ground truth renderings. The latter corresponds to the mesh rendering conducted under identical lighting and shading conditions. Our method generates accurate, high-quality view-dependent renderings, showcasing the precision of our discretely defined density and color distribution along the ray.
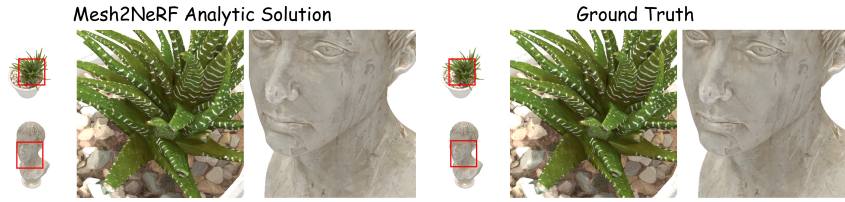
Fig. 3: Volume renderings from the defined radiance fields of Mesh2NeRF analytic solution. Our rendering results are very close to the ground truth, indicating that our defined radiance fields can serve as an effective ground truth representation for NeRFs.

3.3   Mesh2NeRF as Supervision in NeRFs

Our approach enables the generation of density and color attributes for each sampled point along the rays, allowing direct supervision of these values in the outputs of NeRF MLP. For predicted density $\hat{\sigma}$ and color $\hat{\mathbf{c}}_i$ along the ray, alpha $\hat{\alpha}_i$ can be computed using Eq. 4. The overall optimization loss term is composed of alpha and color terms:

$$\mathcal{L}_{alpha} = \sum_{i=1}^{N} |\hat{\alpha}_i - \alpha_i|^2 \tag{8}$$

$$\mathcal{L}_{color} = \sum_{i=1}^{N} \|\hat{\mathbf{c}}_i - \mathbf{c}_i\|_2^2 \tag{9}$$

Optionally, the overall optimization loss can include a term, $\mathcal{L}_{integral}$, accounting for the difference in the ray color integral:

$$\mathcal{L}_{integral} = \left\| \sum_{i=1}^{N} \hat{\alpha}_i \hat{\mathbf{c}}_i \prod_{j=1}^{i-1} (1-\hat{\alpha}_j) - \sum_{i=1}^{N} \alpha_i \mathbf{c}_i \prod_{j=1}^{i-1} (1-\alpha_j) \right\|_2^2 \tag{10}$$

The final optimization loss term is given by:

$$\mathcal{L} = \mathcal{L}_{alpha} + w_{color}\mathcal{L}_{color} + w_{integral}\mathcal{L}_{integral} \tag{11}$$

where w_{color} and $w_{integral}$ are weighting terms.

Space sampling for neural radiance field optimization. Mesh2NeRF offers robust supervision for a neural radiance field generated from a mesh. As depicted in the lower part of Fig. 2, points are sampled along a ray and fed to an MLP to predict density and color (like NeRFs). We emulate a virtual camera in unit spherical to define ray origins and directions similar to NeRFs but use ray casting to compute intersections with the geometry. For efficient sampling of the fields, we implement queries on top of the bounding volume hierarchy acceleration structures of the Embree library [3]. This information optimizes the sampling

of points inside the volume. For rays intersecting the surface, we implement stratified sampling of points along the ray in both the empty scene space and within a narrow band at a distance h to the surface. If a ray does not intersect the mesh, we randomly sample points in the domain along the ray. Mesh2NeRF directly supervises the MLPs at each sampled point using the density and color fields generated from the mesh through the loss terms $\mathcal{L}_{alpha}$ and $\mathcal{L}_{color}$.

3.4   Mesh2NeRF in NeRF Generation Tasks

To facilitate 3D generation tasks, we advocate for the incorporation of Mesh2NeRF into the NeRF generation framework. We build on our generative framework based on the single-stage diffusion NeRF model [15], which combines a triplane NeRF auto-decoder with a triplane latent diffusion model. The pivotal innovation lies in replacing the rendering loss, which relies on multi-view observations rendered from meshes, with our direct supervision.

SSDNeRF revisited. In the context of multiple scene observations, generalizable multi-scene NeRFs can be trained by optimizing per-scene codes and shared parameters, as proposed in [13]. This optimization is achieved by minimizing the rendering loss $\mathcal{L}_{rend}$ associated with each observation image. Consequently, the model is trained as an auto-decoder [47], where each scene code is interpreted as a latent code. To enhance the expressiveness of latent representations, a latent diffusion model (LDM) is employed to learn the prior distribution in the latent space. The LDM introduces Gaussian perturbations into the code at the diffusion time step. Subsequently, a denoising network with trainable weights is utilized to remove noise during the diffusion stage, predicting a denoised code. The LDM is trained using a simplified L2 denoising loss $\mathcal{L}_{diff}$. In the SSDNeRF framework, the training objective aims to minimize the variational upper bound on the negative log-likelihood (NLL) of observed data:

$$\mathcal{L}_{ssdnerf} = w_{rend}\mathcal{L}_{rend} + w_{diff}\mathcal{L}_{diff} \tag{12}$$

Here, the scene codes, prior parameters, and NeRF decoder parameters are jointly optimized, with w_{rend} and w_{diff} serving as weighting factors for the rendering and diffusion losses, respectively.

   With trained diffusion prior, a variety of solvers (such as DDIM [54]) can be used to recursively denoise a random Gaussian noise, until reaching the denoised state in the unconditional sampling process. When the generation is conditioned, e.g., from single-view or sparse-view observations, the sampling process is guided by the gradients of rendering loss from known observation. For more details about the image-guided sampling and fine-tuning, we refer to SSDNeRF [15].

Supervision with Mesh2NeRF. When mesh data is directly used, one can replace the rendering loss $\mathcal{L}_{rend}$ to Mesh2NeRF loss as Eq. 11 during the training process. In the conditional generation stage, since the surface distribution is not available from observation conditions, we use the rendering loss $\mathcal{L}_{rend}$ to compare standard volume rendering with the observation pixel to guide the generation.
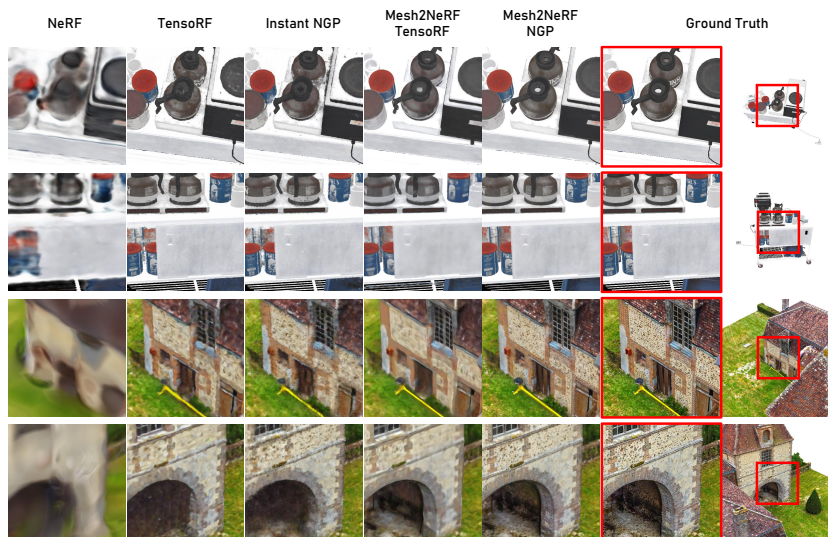
Fig. 4: Comparison of single scene fitting for scenes from Poly Haven and Sketchfab. For each visualized scene, we present two renderings of test views by each method. Our results showcase higher accuracy and a superior ability to capture finer details in renderings when compared to the baseline methods (Mesh2NeRF TensoRF vs. TensoRF and Mesh2NeRF NGP vs. Instant NGP).

4   Results

We illustrate that our direct supervision of radiance fields from meshes significantly improves performance in NeRF optimization and generation tasks. We showcase results of a single scene fitting using Mesh2NeRF, comparing it with NeRF pipelines in Section 4.1. Importantly, we then demonstrate the utility of Mesh2NeRF in diverse 3D generation tasks in Section 4.2 and Section 4.3.

4.1   Single Scene Fitting

This section compares Mesh2NeRF with traditional NeRF methods in representing a single scene via a neural network. Mesh2NeRF directly supervises the neural network from mesh data, while traditional NeRFs use rendering technologies and then fuses information from renderings to the radiance field.

We evaluate our method on scenes from Amazon Berkeley Objects (ABO) [19], Poly Haven Models [1] and Sketchfab Scenes [2]. The ABO dataset contains twelve realistic 3D models of household objects. Each object is rendered at 512×512 pixels from viewpoints sampled on a sphere, covering the object's surface area (90 views as input and 72 for testing). We also show results on six hyper-real models from the Poly Haven library, which consists of quality content designed by artists. Each scene is rendered at 512×512 with 90 input and 120

Table 1: View synthesis results for fitting a single scene on ABO and Poly Haven datasets. Our comparison involves NeRF methods and Mesh2NeRF, utilizing identical network architectures while differing in supervision style.

| Method	ABO Dataset			Poly Haven Dataset		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF	25.09	0.882	0.137	21.24	0.689	0.422
TensoRF	31.33	0.944	0.032	23.32	0.748	0.260
Instant NGP	30.16	0.928	0.039	24.39	0.779	0.185
Mesh2NeRF NeRF	32.40	0.942	0.044	22.75	0.710	0.296
Mesh2NeRF TensoRF	32.00	0.957	0.024	23.97	0.761	0.220
Mesh2NeRF NGP	**33.28**	**0.969**	**0.018**	**25.30**	**0.825**	**0.129**

test views. We use the *Entrée du château des Bois Francs* scene from Sketchfab to evaluate large scene-scale data. The used Sketchfab scene is also rendered at 512×512 pixels with 90 input views and 120 test views. For Mesh2NeRF, we use the textured mesh from both sets by setting a fixed point light. We evaluate our approach with three different encodings and parametric data structures for storing trainable feature embeddings, *i.e.*, the network part of NeRFF [38], TensoRF [12], and Instant NGP [40] with our Mesh2NeRF direct supervision using the corresponding mesh. We denote these three methods by Mesh2NeRF NeRF, Mesh2NeRF TensoRF, and Mesh2NeRF NGP, respectively.

In Table 1, we report mean PSNR/SSIM (higher is better) and LPIPS (lower is better) for objects in both the ABO and Poly Haven datasets. Our method consistently outperforms prior work on both datasets. For the ABO dataset, all three variants of our method surpass baseline optimization across all evaluation metrics. Mesh2NeRF NGP, employing Instant NGP style multi-resolution hash table encodings, achieves the best performance. Compared to the raw Instant NGP version, Mesh2NeRF NGP exhibits notable improvements (+3.12 PSNR, +0.041 SSIM, and −0.021 LPIPS). Similarly, on the Poly Haven dataset, Mesh2NeRF NGP delivers the best results.

We present qualitative results in Fig. 4, illustrating the superiority of our method over NeRF baselines in representing single scenes on challenging *Coffee Cart* from Poly Haven and *Entrée du château des Bois Francs* from Sketchfab. Mesh2NeRF TensoRF supasses the original TensoRF, indicating that Mesh2NeRF supervision contributes to the neural network's ability to learn accurate scene representation. Mesh2NeRF NGP also outperforms Instant NGP, exhibiting fewer artifacts and providing more precise renderings. For additional implementation details, results and analysis, please refer to the supplemental material.

4.2   Conditional Generation

This section presents experimental results of NeRF generation conditioned on images of unseen objects from the ShapeNet Cars and Chairs [11], as well as the KITTI Cars [25]. These datasets pose unique challenges, with the ShapeNet Cars set featuring distinct textures, the ShapeNet Chair set exhibiting diverse
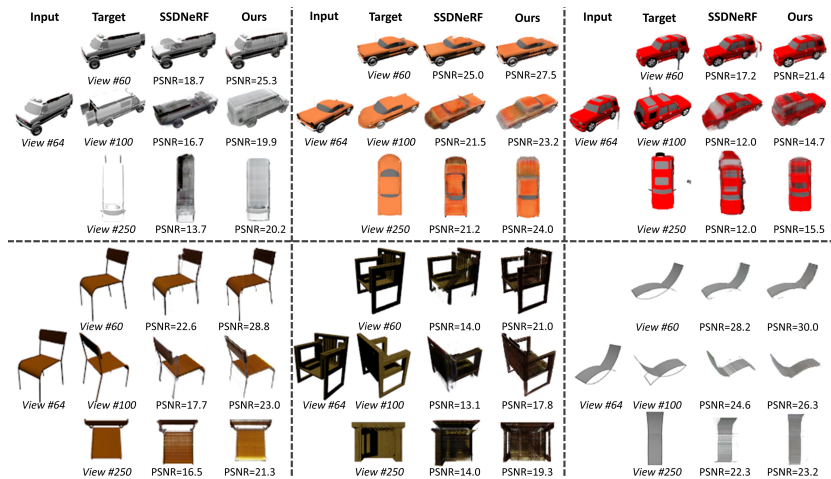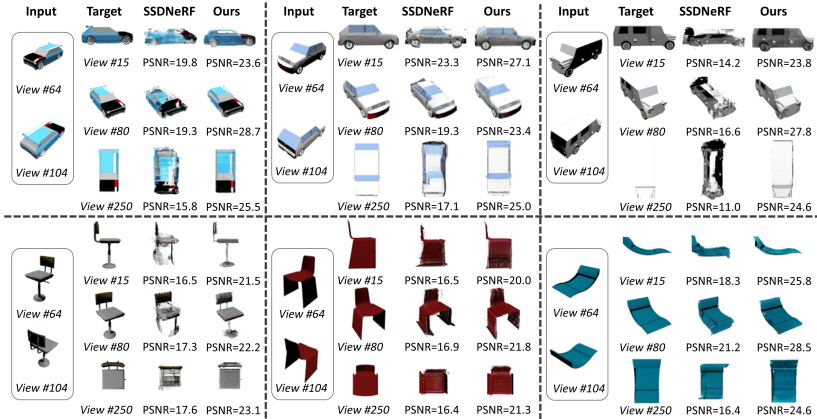
Fig. 5: Qualitative comparison of NeRF generation conditioned on single-view for unseen objects in ShapeNet Cars and Chairs between SSDNeRF and our method. Our approach enables more accurate novel views.

shapes, and KITTI Cars representing real-world inference data with a substantial domain gap when compared to the training data of the generative model.

Implementation information. We adopt SSDNeRF [15] as the framework for sparse-view NeRF reconstruction tasks since it achieves state-of-the-art NeRF generation performance. In its single-stage training, SSDNeRF jointly learns triplane features of individual scenes, a shared NeRF decoder, and a triplane diffusion prior. The optimization involves a pixel loss from known views. For *Ours*, we replace the pixel loss with Mesh2NeRF supervision (Eq. 11) from the train set meshes. We generate training data for ShapeNet Cars and Chairs using the same lighting and mesh processes as in our Mesh2NeRF setup. SSDNeRF is trained using the official implementation; our model is trained with the same viewpoints as in the SSDNeRF experiments, without additional ray sampling, ensuring a fair comparison. Our evaluation primarily centers on assessing the quality of novel view synthesis based on previously unseen images. PSNR, SSIM, and LPIPS are evaluated to measure the image quality.

Comparative evaluation. In Table 2, a comprehensive comparison between our method and the state-of-the-art SSDNeRF is provided. Our method demonstrates overall superiority in conditional NeRF reconstruction tasks when provided with sparse-view conditions (ranging from one to four views) for both the Car and Chair categories. Fig. 5 visually illustrates the impact of Mesh2NeRF supervision on the accuracy of NeRFs in the context of single-view condition generation. Notably, our method outperforms SSDNeRF, especially when dealing with objects that possess unconventional forms (e.g., the first car with a roof of the same color as the background, and the third red car exhibiting input noise). SSDNeRF struggles in such scenarios, yielding unreasonable geometries.

Fig. 6: Qualitatively comparison of NeRF generation conditioned on two-view for unseen objects in ShapeNet Cars and Chairs. Three novel views are displayed for each two-view input. Our approach enables more accurate novel views.

Table 2: Conditional NeRF generation results on ShapeNet Cars and Chairs.

	Cars 1-view			Cars 2-view			Cars 3-view			Cars 4-view		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
SSDNeRF	21.09	0.881	0.104	24.67	0.926	**0.071**	25.71	0.934	0.069	**26.54**	0.939	0.067
Ours	**21.78**	**0.893**	**0.101**	**24.98**	**0.932**	0.072	**25.89**	**0.942**	**0.066**	26.51	**0.945**	**0.062**
	Chairs 1-view			Chairs 2-view			Chairs 3-view			Chairs 4-view		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
SSDNeRF	19.05	0.853	0.133	19.65	0.859	0.136	20.42	0.863	0.149	21.84	0.884	0.134
Ours	**19.62**	**0.859**	**0.128**	**22.22**	**0.888**	**0.112**	**23.03**	**0.900**	**0.116**	**22.68**	**0.907**	**0.109**

Qualitative comparisons in the setting of 2-view conditions are presented in both Fig. 6. In these scenarios, our approach consistently outshines SSDNeRF, producing superior results across various instances.

Single-view NeRF generation from real images. We also conduct a comparison with SSDNeRF using real KITTI car data for conditional generation based on a single-view input. This task poses a challenge because the generative model is trained on synthetic ShapeNet cars, resulting in a significant domain gap. The input images are obtained from the KITTI 3D object detection dataset [25] using annotated 3D bounding boxes. To align them with the ShapeNet Cars dataset, we utilize provided bounding box dimensions and poses. Segmentation masks remove the background [26]. Images are cropped and resized to 128×128. The processed image conditions our generative model, guiding the generation of a radiance field for rendering novel views around the cars. In Fig. 7, we show qualitative examples of novel view synthesis. Both SSDNeRF and our model successfully reconstruct the car's shape in most cases. However,

Fig. 7: Qualitative comparison of novel view synthesis on in-the-wild car images from the KITTI dataset. Four novel views are displayed for each single-view input. Our approach enables more reasonable novel views.

SSDNeRF fails in the last row sample due to color similarity between the car and the background. Our model excels in reconstructing the radiance field of the car in this challenging scenario. Regarding the generated car NeRFs, our approach produces more accurate textures with improved global consistency and better correspondence to the input image. In contrast, SSDNeRF tends to generate textures only in the vicinity of the observation area, leaving other regions textureless or with incorrect colors. Our results exhibit greater realism and coherence across the entire scene, showcasing the superior generalization capability of our model, especially in the face of substantial domain gaps.

4.3 Unconditional Radiance Field Synthesis

We conduct evaluation for unconditional generation using the Objaverse Mugs collection [23], which consists of 153 mug models. The Mugs dataset presents a challenge in generating shape geometry and realistic textures due to the limited training samples. The baseline SSDNeRF is trained on rendered images from 50 views of each mesh, and Mesh2NeRF is trained on meshes with the same viewpoints. Both models underwent training for 100,000 iterations.

As shown in Fig. 8, both SSDNeRF and Mesh2NeRF generate reasonable NeRFs in the rendered images. However, when extracting meshes from NeRFs of SSDNeRF, inaccuracies are evident; the geometries of its unconditional generation do not faithfully represent the real geometry of the training samples or real data. Notably, traditional NeRF supervision struggles to enable the generative model to capture precise geometric details, such as whether a cup is sealed. In contrast, Mesh2NeRF synthesizes diverse and realistic geometries corresponding to the rendered images. This underscores the robustness of Mesh2NeRF supervision, proving instrumental in generating physically plausible 3D content.

SSDNeRF Unconditional Generated Mugs   Ours Unconditional Generated Mugs

Fig. 8: Qualitative comparison between unconditional generative models training on Objaverse Mugs. We present two renderings and extracted mesh from each generated NeRF. Our approach exhibits a notable advantage in reasonable geometry, particularly for shapes that suffer from self-occlusions when trained only with image supervision.

4.4   Limitations

While Mesh2NeRF effectively employs direct NeRF supervision for NeRF representation and generation tasks, several limitations persist. Similar to NeRF, which bakes lighting information into appearance, we model the result of the Phong lighting model in the generated radiance field for compatibility with existing approaches. Another limitation stems from the ray sampling employed by baselines due to the dependence on rendered images. Hence, future work should delve into more efficient sampling techniques that leverage ground truth geometry and appearance, bypassing the need for sampling schemes relying on virtual cameras. Furthermore, acknowledging the diversity of mesh data, the application of Mesh2NeRF to generative models can span across various categories of mesh data, contributing to the development of a more universal and robust prior.

5   Conclusion

We have introduced Mesh2NeRF, a new approach for directly converting textured mesh into radiance fields. By leveraging surface-based density and view-dependent color modeling, our method ensures faithful representation of complex scenes. Employing Mesh2NeRF as direct 3D supervision for NeRF optimization yields superior performance, enhancing accuracy and detail in view synthesis across diverse hyper-real scenes. Importantly, the versatility of our method extends to various generalizable NeRF tasks, encompassing both conditional and unconditional NeRF generation. When combined with available mesh data, Mesh2NeRF mitigates traditional NeRF supervision shortcomings, leading to enhanced overall performance on synthetic datasets and real-world applications. Mesh2NeRF is applicable to various NeRF representations, such as frequency encoding (NeRF), hash encoding (NGP), tensorial (TensoRF), and triplane (SSDNeRF), consistently improving respective tasks. We believe that this work contributes valuable insights to the integration of mesh and NeRF representations, paving the way for new possibilities in 3D content generation.

References

1. Poly haven models. `https://polyhaven.com/models`
2. Skatchfab 3d models. `https://sketchfab.com/3d-models`
3. Áfra, A.T., Wald, I., Benthin, C., Woop, S.: Embree ray tracing kernels: overview and new features. In: SIGGRAPH Talks. pp. 52:1–52:2. ACM (2016)
4. Anciukevičius, T., Xu, Z., Fisher, M., Henderson, P., Bilen, H., Mitra, N.J., Guerrero, P.: Renderdiffusion: Image diffusion for 3D reconstruction, inpainting and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12608–12618 (2023)
5. Azinović, D., Martin-Brualla, R., Goldman, D.B., Nießner, M., Thies, J.: Neural rgb-d surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6290–6301 (2022)
6. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021)
7. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)
8. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al.: Improving image generation with better captions. Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf **2**(3), 8 (2023)
9. 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. pp. 16123–16133 (2022)
10. 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. pp. 5799–5809 (2021)
11. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
12. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
13. Chen, A., Xu, Z., Wei, X., Tang, S., Su, H., Geiger, A.: Factor fields: A unified framework for neural fields and beyond. arXiv preprint arXiv:2302.01226 (2023)
14. 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)
15. Chen, H., Gu, J., Chen, A., Tian, W., Tu, Z., Liu, L., Su, H.: Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In: Proceedings of the

IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2416–2425 (October 2023)

16. Chen, Z., Tagliasacchi, A., Zhang, H.: Bsp-net: Generating compact meshes via binary space partitioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 45–54 (2020)

17. Cheng, Y.C., Lee, H.Y., Tulyakov, S., Schwing, A.G., Gui, L.Y.: Sdfusion: Multi-modal 3D shape completion, reconstruction, and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4456–4465 (2023)

18. Chou, G., Bahat, Y., Heide, F.: Diffusion-sdf: Conditional generative modeling of signed distance functions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2262–2272 (2023)

19. Collins, J., Goel, S., Deng, K., Luthra, A., Xu, L., Gundogdu, E., Zhang, X., Yago Vicente, T.F., Dideriksen, T., Arora, H., Guillaumin, M., Malik, J.: Abo: Dataset and benchmarks for real-world 3D object understanding. CVPR (2022)

20. 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)

21. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)

22. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. Advances in Neural Information Processing Systems **36** (2024)

23. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153 (2023)

24. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised NeRF: Fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12882–12891 (2022)

25. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)

26. Heylen, J., De Wolf, M., Dawagne, B., Proesmans, M., Van Gool, L., Abbeloos, W., Abdelkawy, H., Reino, D.O.: Monocinis: Camera independent monocular 3d object detection using instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 923–934 (2021)

27. Höllein, L., Božič, A., Müller, N., Novotny, D., Tseng, H.Y., Richardt, C., Zollhöfer, M., Nießner, M.: Viewdiff: 3d-consistent image generation with text-to-image models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)

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

29. Karnewar, A., Vedaldi, A., Novotny, D., Mitra, N.J.: Holodiffusion: Training a 3D diffusion model using 2d images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18423–18433 (2023)

30. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3907–3916 (2018)

31. Lin, K.E., Lin, Y.C., Lai, W.S., Lin, T.Y., Shih, Y.C., Ramamoorthi, R.: Vision transformer for NeRF-based view synthesis from a single input image. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 806–815 (2023)

32. Liu, M., Shi, R., Chen, L., Zhang, Z., Xu, C., Wei, X., Chen, H., Zeng, C., Gu, J., Su, H.: One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. arXiv preprint arXiv:2311.07885 (2023)

33. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023)

34. Liu, S., Li, T., Chen, W., Li, H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7708–7717 (2019)

35. 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. pp. 2837–2845 (2021)

36. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-NeRF for shape-guided generation of 3D shapes and textures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12663–12673 (2023)

37. Mildenhall, B., Hedman, P., Martin-Brualla, R., Srinivasan, P.P., Barron, J.T.: NeRF in the dark: High dynamic range view synthesis from noisy raw images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16190–16199 (2022)

38. 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), 99–106 (2021)

39. 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. pp. 4328–4338 (2023)

40. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)

41. Nam, G., Khlifi, M., Rodriguez, A., Tono, A., Zhou, L., Guerrero, P.: 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. arXiv preprint arXiv:2212.00842 (2022)

42. Nash, C., Ganin, Y., Eslami, S.A., Battaglia, P.: Polygen: An autoregressive generative model of 3d meshes. In: International conference on machine learning. pp. 7220–7229. PMLR (2020)

43. 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)

44. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5480–5490 (2022)

45. 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. pp. 11453–11464 (2021)

46. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3504–3515 (2020)

47. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 165–174 (2019)

48. Reiser, C., Peng, S., Liao, Y., Geiger, A.: KiloNeRF: Speeding up neural radiance fields with thousands of tiny mlps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14335–14345 (2021)

49. Roessle, B., Barron, J.T., Mildenhall, B., Srinivasan, P.P., Nießner, M.: Dense depth priors for neural radiance fields from sparse input views. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12892–12901 (2022)

50. 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. pp. 10684–10695 (2022)

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

52. Shue, J.R., Chan, E.R., Po, R., Ankner, Z., Wu, J., Wetzstein, G.: 3d neural field generation using triplane diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20875–20886 (2023)

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

54. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)

55. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15598–15607 (2021)

56. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. arXiv preprint arXiv:2306.07881 (2023)

57. Tagliasacchi, A., Mildenhall, B.: Volume rendering digest (for NeRF). arXiv preprint arXiv:2209.02417 (2022)

58. Tang, J., Zhou, H., Chen, X., Hu, T., Ding, E., Wang, J., Zeng, G.: Delicate textured mesh recovery from NeRF via adaptive surface refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17739–17749 (October 2023)

59. Tang, J., Wang, T., Zhang, B., Zhang, T., Yi, R., Ma, L., Chen, D.: Make-It-3D: High-fidelity 3D creation from a single image with diffusion prior. arXiv preprint arXiv:2303.14184 (2023)

60. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5481–5490. IEEE (2022)

61. 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. pp. 12619–12629 (2023)

62. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021)

63. Wei, Y., Liu, S., Rao, Y., Zhao, W., Lu, J., Zhou, J.: NerfingMVS: Guided optimization of neural radiance fields for indoor multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5610–5619 (2021)

64. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. Advances in neural information processing systems **29** (2016)

65. Yariv, L., Hedman, P., Reiser, C., Verbin, D., Srinivasan, P.P., Szeliski, R., Barron, J.T., Mildenhall, B.: Bakedsdf: Meshing neural sdfs for real-time view synthesis. arXiv preprint arXiv:2302.14859 (2023)

66. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)