arXiv:2403.11131v2 [cs.CV] 18 Jul 2024

Omni-Recon: Harnessing Image-based Rendering for General-Purpose Neural Radiance Fields

Yonggan Fu, Huaizhi Qu, Zhifan Ye, Chaojian Li,
Kevin Zhao, and Yingyan (Celine) Lin

Georgia Institute of Technology
{yfu314,zye327,cli851,kzhao14,celine.lin}@gatech.edu

Abstract. Recent breakthroughs in Neural Radiance Fields (NeRFs) have sparked significant demand for their integration into real-world 3D applications. However, the varied functionalities required by different 3D applications often necessitate diverse NeRF models with various pipelines, leading to tedious NeRF training for each target task and cumbersome trial-and-error experiments. Drawing inspiration from the generalization capability and adaptability of emerging foundation models, our work aims to develop one general-purpose NeRF for handling diverse 3D tasks. We achieve this by proposing a framework called Omni-Recon, which is capable of (1) generalizable 3D reconstruction and zero-shot multitask scene understanding, and (2) adaptability to diverse downstream 3D applications such as real-time rendering and scene editing. Our key insight is that an image-based rendering pipeline, with accurate geometry and appearance estimation, can lift 2D image features into their 3D counterparts, thus extending widely explored 2D tasks to the 3D world in a generalizable manner. Specifically, our Omni-Recon features a general-purpose NeRF model using image-based rendering with two decoupled branches: one complex transformer-based branch that progressively fuses geometry and appearance features for accurate geometry estimation, and one lightweight branch for predicting blending weights of source views. This design achieves state-of-the-art (SOTA) generalizable 3D surface reconstruction quality with blending weights reusable across diverse tasks for zero-shot multitask scene understanding. In addition, it can enable real-time rendering after baking the complex geometry branch into meshes, swift adaptation to achieve SOTA generalizable 3D understanding performance, and seamless integration with 2D diffusion models for text-guided 3D editing. Our code is available at: https://github.com/GATECH-EIC/Omni-Recon.

1 Introduction

Neural Radiance Fields (NeRFs) [41] have garnered significant attention and hold promise to become crucial enablers for emerging 3D applications. However, deploying NeRFs in real-world scenarios still requires substantial effort, as different 3D applications often demand distinct NeRF models with varied

pipelines, leading to tedious NeRF training for each target task and cumbersome trial-and-error experiments. For instance, while both cross-scene generalization and real-time rendering are desirable in the real world, generalizable NeRFs for instant reconstruction [10, 38, 77] and real-time NeRFs using mesh-based rasterization [12, 65, 85] typically feature diverse pipelines, making it hard to simultaneously satisfy the two requirements. Additionally, understanding new scene properties [36, 90, 93] necessitates training new NeRF models, which is not scalable given the increasing number of scene properties of interest.

In parallel, the emergence of foundation models has driven transformative progress in real-world artificial intelligence applications. Inspired by this exciting trend, there has been a growing interest in integrating NeRFs into this realm by developing general-purpose NeRFs for handling various 3D tasks. Specifically, similar to the generalization capability and adaptability of foundation models, we expect general-purpose NeRFs to possess three essential capabilities, including (1) generalizable 3D reconstruction for instant surface reconstruction and novel view synthesis, (2) zero-shot multitask scene understanding of various scene properties, e.g., semantics and edges, and (3) adaptability to diverse downstream tasks such as real-time rendering and scene editing. Achieving these capabilities is promising to enable users to effortlessly start from these general-purpose NeRFs when facing a new 3D application, significantly reducing the need for training and experimenting with various NeRF pipelines.

Despite the promise, existing NeRF pipelines have not yet realized the mentioned capabilities. Firstly, per-scene optimized NeRFs [2, 3, 22, 25, 27, 42, 85] rely on costly per-scene training, thus precluding generalization capabilities. Secondly, although generalizable NeRF designs [10, 38, 54, 68, 75, 77, 82, 88] can achieve cross-scene generalization, their huge computational complexity limits their suitability for application scenarios where real-time rendering is crucial. Some generalizable surface reconstruction methods [34, 55] can support mesh extraction, but an expensive volumetric rendering process, independent of the extracted mesh, is still necessary for rendering novel views, leaving the demand for real-time rendering unmet. Furthermore, their potential for zero-shot scene understanding and scene editing remains unexplored.

Given the significant potential and associated challenges, our work aims to advance the integration of NeRFs into the realm of foundation models by developing general-purpose NeRFs. To achieve this goal, we propose a framework called Omni-Recon, which is an image-based rendering pipeline that realizes all the mentioned capabilities. The key insight leveraged by our Omni-Recon is that an image-based rendering pipeline, when equipped with accurate geometry and appearance estimation, can lift 2D image features into their 3D counterparts, thus extending widely explored 2D tasks to the 3D world in a generalizable manner. Specifically, our contributions are summarized as follows:

- We introduce Omni-Recon, a general-purpose NeRF pipeline, which possesses both decent generalization capabilities, including generalizable 3D reconstruction and zero-shot multitask scene understanding, and adaptability to diverse downstream 3D applications.

- Our Omni-Recon features a general-purpose NeRF model backbone using image-based rendering with two decoupled branches: one complex transformer-based geometry branch that progressively fuses geometry and appearance features to accurately predict a signed distance function (SDF), and one lightweight appearance branch for predicting blending weights of source views from the target scene. The advantages of this design include: (1) ensuring accurate geometry estimation for high-quality 3D reconstruction, and (2) enabling real-time rendering after baking the complex geometry branch into meshes and using the lightweight appearance branch as a shader.
- We intriguingly find that a properly trained image-based rendering pipeline can directly lift 2D monocular priors to 3D novel views by reusing the blending weights predicted by the appearance branch and leverage this to enable generalizable, zero-shot, and multitask scene understanding.
- We demonstrate the adaptability of Omni-Recon by extending it to diverse downstream tasks, including: (1) a Nvdiffrast-based [31] real-time rendering pipeline using the extracted mesh, (2) parameter-efficient tuning (PET) of Omni-Recon on downstream generalizable scene understanding tasks, and (3) a new text-guided scene editing scheme by iteratively editing and reconstructing the source images from a 3D scene using 2D diffusion models.
- Extensive experiments demonstrate that our Omni-Recon can achieve state-of-the-art (SOTA) generalizable 3D surface reconstruction and scene understanding performance. Additionally, it can support real-time rendering of a new scene after rapid adaptation, along with easy-to-use 3D scene editing.

Our Omni-Recon underscores the potentially wide usage of image-based rendering pipelines in diverse real-world 3D applications, which may have been overlooked by previous works. We believe that the insights we provide could ignite future innovations in more advanced generalizable rendering pipelines.

2 Related Work

Neural Radiance Fields. NeRFs [42], which represent the target scene as a continuous volume with density and view-dependent color, have garnered increasing attention thanks to their SOTA rendering quality when compared to prior approaches using explicit neural representations [24,39,60,66]. Subsequent works further improve NeRFs from various angles, such as enhancing their rendering quality [2,3,11,72], training NeRFs with sparse views [18,28,46,69,80,81], extending NeRFs to large-scale scenes [64, 70, 78, 79, 91], or applying NeRFs to other tasks, such as generative modeling [7, 47, 58, 74], dynamic scenes [6, 33, 49, 50], or scene understanding [21, 30, 73, 93].

Generalizable NeRFs. Vanilla NeRFs rely on per-scene optimization, which limits their ability to perform instant reconstruction. To address this limitation, generalizable NeRF designs [10,13,29,38,54,68,75,77,82,88] achieve cross-scene generalization by conditioning vanilla NeRFs on the source views from the target new scene. Specifically, [68,75,77,88] take the extracted scene features from the source views as inputs to reconstruct the radiance field via a one-shot forward

pass. Subsequent works further improve cross-scene generalization capabilities by enhancing scene geometry estimation [10, 29, 38] and by introducing attention mechanisms and mixture-of-experts into NeRF backbone design [15, 36, 75]. Despite the promise of generalizable NeRFs, they still fall short in terms of rendering efficiency due to their huge computational complexity, which entails costly inference for each sampled point along the ray. While some generalizable surface reconstruction methods [34, 55] can support mesh extraction, a costly volumetric rendering process, independent of the extracted mesh, is still necessary for rendering novel views. This poses a hindrance to their deployment in 3D applications that require real-time rendering. Furthermore, their potential for zero-shot scene understanding and scene editing remains unexplored.

Efficient scene representations. Various scene representations have been proposed to accelerate NeRFs' training and inference process. To speed up NeRF training, [8, 9, 20, 43, 62] integrate explicit neural representations with volumetric rendering. In parallel, to speed up NeRF inference/rendering, [35, 52, 53] attempt to reduce the complexity of the multi-layer perceptron (MLP) model in NeRF. In addition, [22, 25, 27, 37, 53, 59, 87] explore the free space in a target 3D scene to improve the sampling efficiency and thus reduce unnecessary MLP inference. Motivated by the insight of deferred rendering [67], [12, 25, 65, 85] pre-compute scene features offline and thus require less computation at run-time. In particular, [12, 65, 85] represents NeRF as a set of textured polygons, where the textures can be pre-computed and thus the NeRF pipeline can be efficiently executed in modern graphics hardware. However, these efficient scene representations still necessitate costly per-scene optimization, lacking generalization capabilities.

3 Omni-Recon: The General-Purpose NeRF Backbone

To develop general-purpose NeRFs for diverse 3D tasks, in this section, we design a general-purpose NeRF backbone and demonstrate its SOTA generalizable 3D reconstruction capability. We will leverage this backbone to enable zero-shot multitask scene understanding in Sec. 4 and adapt it to diverse downstream 3D tasks, including real-time rendering, PET for 3D scene understanding, and text-guided scene editing in Sec. 5.

3.1 Backbone Design Overview

Design principles. Our design is driven by the following principles: (1) Similar to foundation models, the ability to instantly reconstruct a new scene with generalizable capabilities is essential. Therefore, we leverage image-based rendering [10, 38, 77], which lifts 2D image features into their 3D counterparts, as our backbone design and uncover their decent capabilities in diverse 3D applications, which may have been overlooked by previous works. (2) Considering a wide range of 3D applications that require real-time rendering, it is highly desirable that our backbone design can be seamlessly converted into efficient 3D representations, particularly meshes, to enable real-time rendering.

Our backbone design. As shown in Fig. 1, the backbone model in Omni-Recon leverages image-based rendering with two decoupled branches: one complex transformer-based geometry branch that progressively fuses geometry and

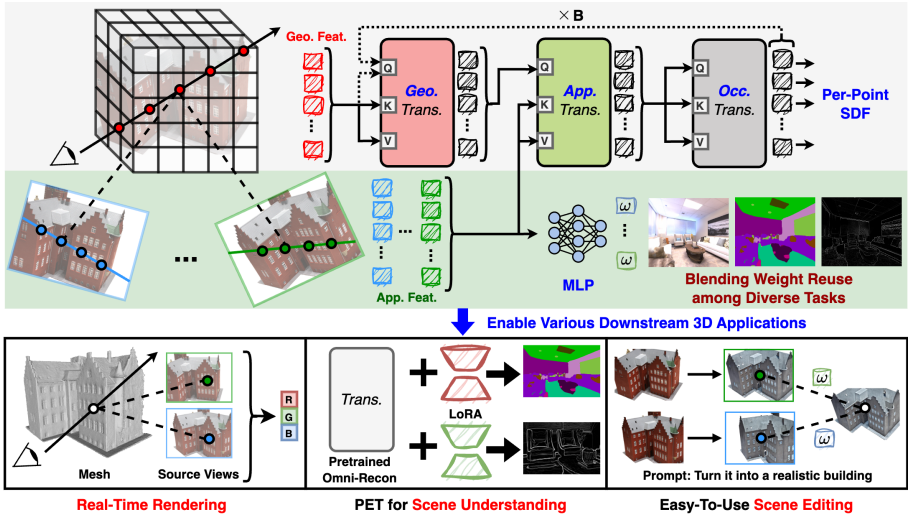

Fig. 1: An overview of our Omni-Recon framework.

appearance features to accurately predict an SDF, and one lightweight appearance branch for estimating blending weights of source views. Specifically, the progressive feature fusion in the geometry branch empowers accurate geometry estimation for high-quality 3D reconstruction, and the decoupling between geometry and appearance enables real-time rendering after baking the complex geometry branch into meshes, with the lightweight appearance branch serving as a shader. Additionally, the nature of image-based rendering, when equipped with accurate geometry and appearance estimation, can enable a wide range of downstream 3D applications, as shown in Sec. 4 and Sec. 5.

3.2 The Overall Image-Based Rendering Pipeline

Following generalizable NeRFs [10,38,77,88], our NeRF backbone is conditioned on source views from the target scene. This is achieved by estimating the geometry and appearance based on the extracted features of the source views from the target scene, which is elaborated as follows.

Acquire geometry features. Given N source views $\{I_i\}_{i=1}^N \in \mathbb{R}^{H\times W\times 3}$ from the target scene, we utilize a CNN encoder [34, 55] to extract source features $\{\mathbf{F}_i\}_{i=1}^N \in \mathbb{R}^{H\times W\times C}$ from the source views. Next, we construct a 3D feature volume $V \in \mathbb{R}^{M\times M\times M\times C}$, which serves similar roles as cost volumes for multiview stereo [10, 83], to aggregate geometry information from different source views, following [44, 55, 63]. Specifically, we divide the bounding volume of the scene into M^3 voxels and project each voxel center onto N source views, where the mean and variance of N source features are computed and concatenated as the corresponding voxel feature. We then leverage a 3D U-Net [14] to further enhance the feature volume to acquire the final geometry feature volume V.

Acquire appearance features. For each sampled 3D point $\mathbf{p}(t) = \mathbf{o}+t\mathbf{d}$ along the ray emitted from the target pixel on the novel view with camera origin $\mathbf{o}$

and direction $\mathbf{d}$, we project it on the source views to acquire the projected colors $\{\mathbf{c}_i\}_{i=1}^N$ and appearance features $\{\mathbf{f}_i\}_{i=1}^N$ using bilinear interpolation.

Geometry and appearance estimation. For more accurate reconstruction of the scene geometry and mitigation of the intrinsic color-density ambiguity [76], we estimate the per-point SDF and model the density as a function of SDF, following [40,55,76]. For appearance, we estimate the per-point radiance through color blending, i.e., predicting the blending weights of the projected colors on the source views to compose the per-point color, following [38,77,88]. The geometry and appearance are modeled using separate branches and can be formulated as:

$$s = \mathbf{M}_{sdf}(\{\mathbf{f}_i\}_{i=1}^N, V), \quad \{\omega_i\}_{i=1}^N = \mathbf{M}_{color}(\{\mathbf{f}_i\}_{i=1}^N, \mathbf{d}), \tag{1}$$

where s is the SDF, $\{\omega_i\}_{i=1}^N$ are the blending weights, and $\mathbf{M}_{sdf}$ and $\mathbf{M}_{color}$ are the geometry and appearance branches, respectively. The per-point radiance can be derived as the weighted sum $\hat{\mathbf{c}} = \sum_{i=1}^N \omega_i \mathbf{c}_i$ across N source views.

For SDF-based volumetric rendering, we adopt the formulation in NeuS [76]. Specifically, the color is accumulated along the K sampled points along each ray:

$$\hat{\mathbf{C}} = \sum_{j=1}^{K} T_j \alpha_j \hat{\mathbf{c}}_j, \quad \alpha_j = 1 - \exp(-\int_{t_j}^{t_{j+1}} \rho(t)dt), \tag{2}$$

where $T_j = \prod_{n=1}^{j-1}(1-\alpha_n)$ is the accumulated transmittance and $\rho(t)$ is the density determined by the estimated SDF, following the definition in NeuS [76].

Complexities of the two branches. When predicting the radiance of a sampled 3D point, the colors of valid projections on the source views are already close to the radiance of this 3D point. In light of this, we employ a lightweight $\mathbf{M}_{color}$, consisting of three MLP layers, for efficient blending weight prediction. In contrast, the geometry branch should be allocated with sufficient complexity for accurately estimating the occlusion effects among different sampled points along the same ray as well as that among the sampled points and the source views. We introduce the design of the geometry branch in the next subsection.

3.3 The Proposed Transformer-based Geometry Branch

Overview. Our geometry branch leverages transformer modules [71] to progressively fuse geometry and appearance features, thus properly handling the two occlusion effects mentioned above. Specifically, for the K sampled 3D points along the ray, we acquire their geometry features $\{\mathbf{v}_k\}_{k=1}^K$ from the geometry feature volume V, which are sequentially processed through B blocks (B=2 in our design), each consisting of a geometry transformer, an appearance transformer, and an occlusion transformer, to produce the per-point SDF.

The geometry transformer. The geometry transformer fuses the geometry features into its inputs by using $\{\mathbf{v}_k\}_{k=1}^K$ as the key and value in a cross-attention scheme [71] and performs attention across the sampled points along the same ray to model their occlusion effects. This process can be formulated as follows:

$$\mathbf{M}_{sdf}^{geo}(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K) = CrossAttention\,(\mathbf{q}=\mathbf{x}, \mathbf{k}=\mathbf{v}=\{\mathbf{v}_k\}_{k=1}^K), \tag{3}$$

where $\mathbf{x}$ is the input to the geometry transformer, which is exactly the geometry features themselves for the geometry transformer in the first block, and $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$ are the query, key, and value of the attention scheme [71], respectively.

The appearance transformer. The appearance transformer integrates the appearance features $\{\mathbf{f}_i\}_{i=1}^N$ into the outputs of the geometry transformer, considering the potential occlusions among the sampled points and the source views that may cause invalid projections. This integration is achieved using subtraction attention [75, 92], which is more effective for geometric relationship reasoning [75, 92], with $\{\mathbf{f}_i\}_{i=1}^N$ as the key and value. Specifically, subtraction attention computes the attention scores between the input query features and each source view in the key features, which are then used to aggregate different source views in the value features into new output features with the same dimension as the input query features. This process can be formulated as follows:

$$\mathbf{M}_{sdf}^{appr}(\mathbf{x}, \{\mathbf{f}_i\}_{i=1}^N) = SubAttention\left(\mathbf{q}=\mathbf{x}, \mathbf{k}=\mathbf{v}=\{\mathbf{f}_i\}_{i=1}^N\right), \tag{4}$$

where $\mathbf{x}$ is the output of $\mathbf{M}_{sdf}^{geo}$. The detailed formulation of subtraction attention is provided in the supplementary materials.

The occlusion transformer. We further integrate an occlusion transformer that performs self-attention across sampled points along the same ray to model their occlusion effects explicitly:

$$\mathbf{M}_{sdf}^{occ}(\mathbf{x}) = SelfAttention\left(\mathbf{q}=\mathbf{k}=\mathbf{v}=\mathbf{x}\right), \tag{5}$$

where $\mathbf{x}$ is the output of $\mathbf{M}_{sdf}^{appr}$. We apply another MLP on top of the output features of the last M_{sdf}^{occ} module to predict the per-point SDF s in Eq. 1.

3.4 Training Objectives

Our NeRF backbone is trained using two objectives: (1) a photometric loss $\mathcal{L}_{rgb}$ between the rendered images and the ground-truth images, and (2) a depth loss $\mathcal{L}_{depth}$ between the rendered depth $\hat{\mathbf{D}} = \sum_{j=1}^K T_j \alpha_j t$ and the ground-truth depth. The overall objective can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{color} + \beta \mathcal{L}_{depth}, \tag{6}$$

where $\mathcal{L}_{rgb} = \frac{1}{R}\sum_{\mathbf{r}=1}^R \left\|\hat{\mathbf{C}}_\mathbf{r} - \mathbf{C}_\mathbf{r}\right\|_2^2$ with R denoting the number of rendered pixels, $\mathcal{L}_{depth} = \frac{1}{R}\sum_{\mathbf{r}=1}^R \left\|\hat{\mathbf{D}}_\mathbf{r} - \mathbf{D}_\mathbf{r}\right\|_2^2$, and β is set as 1.

3.5 Evaluation: Generalizable 3D Reconstruction

Setup. We assess the backbone of Omni-Recon by evaluating its generalizable 3D reconstruction capability. Datasets: Following [10, 34, 40, 55], we adopt the DTU dataset [1], which comprises 124 different scenes and 7 distinct lighting conditions. For testing, 15 scenes are utilized, while the remaining scenes are allocated for training, following the dataset split in [34, 40, 55]. Training settings: During training, we employ $N = 4$ source views with a resolution of 640×512. The ray number per batch and batch size are set to 1024 and 2, respectively,

Table 1: Benchmark the quantitative performance in sparse view mesh reconstruction in terms of Chamfer distance (the lower, the better) on 15 testing scenes on DTU. The best scores are in **bold**, and the second-best are underlined.

Method	Mean	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122
COLMAP [57]	1.52	**0.90**	2.89	1.63	1.08	2.18	1.94	1.61	1.30	2.34	1.28	1.10	1.42	0.76	1.17	1.14
MVSNet [83]	1.22	1.05	2.52	1.71	1.04	1.45	**1.52**	0.88	**1.29**	1.38	1.05	**0.91**	**0.66**	0.61	1.08	1.16
IDR [86]	3.39	4.01	6.40	3.52	1.91	3.96	2.36	4.85	1.62	6.37	5.97	1.23	4.73	0.91	1.72	1.26
VolSDF [84]	3.41	4.03	4.21	6.12	0.91	8.24	1.73	2.74	1.82	5.14	3.09	2.08	4.81	0.60	3.51	2.18
UNISURF [48]	4.39	5.08	7.18	3.96	5.30	4.61	2.24	3.94	3.14	5.63	3.40	5.09	6.38	2.98	4.05	2.81
NeuS [76]	4.00	4.57	4.49	3.97	4.32	4.63	1.95	4.68	3.83	4.15	2.50	1.52	6.47	1.26	5.57	6.11
PixelNeRF [88]	6.18	5.13	8.07	5.85	4.40	7.11	4.64	5.68	6.76	9.05	6.11	3.95	5.92	6.26	6.89	6.93
IBRNet [77]	2.32	2.29	3.70	2.66	1.83	3.02	2.83	1.77	2.28	2.73	1.96	1.87	2.13	1.58	2.05	2.09
MVSNeRF [10]	2.09	1.96	3.27	2.54	1.93	2.57	2.71	1.82	1.72	2.29	1.75	1.72	1.47	1.29	2.09	2.26
SparseNeuS [40]	1.96	2.17	3.29	2.74	1.67	2.69	2.42	1.58	1.86	1.94	1.35	1.50	1.45	0.98	1.86	1.87
VolRecon [55]	1.38	1.20	2.59	1.56	1.08	1.43	1.92	1.11	1.48	1.42	1.05	1.19	1.38	0.74	1.23	1.27
ReTR [34]	1.17	1.05	2.31	**1.44**	0.98	1.18	**1.52**	0.88	1.35	1.30	0.87	1.07	0.77	0.59	**1.05**	1.12
Omni-Recon (Ours)	**1.13**	0.91	**2.13**	1.52	**0.93**	**1.09**	1.70	**0.84**	**1.29**	**1.20**	**0.83**	1.04	0.81	**0.55**	**1.05**	**1.05**

following [34,55]. Mesh extraction: We utilize TSDF fusion [45,63] to reconstruct scene meshes from predicted source view depths, following [34,55]. More detailed model/training settings can be found in the supplementary materials.

Baselines. We primarily benchmark against SOTA generalizable neural implicit reconstruction methods [16, 34, 55], along with generalizable neural rendering methods [10, 77, 88], per-scene optimization methods [48, 76, 84, 86], and multi-view stereo methods [57, 83]. We directly report their official results in [34, 55].

Sparse view reconstruction. We benchmark the performance of sparse view reconstruction using three views from each testing scene in DTU, following the setting in [16, 34, 55]. As shown in Tab. 1, we can observe that (1) Our Omni-Recon can achieve new SOTA reconstruction performance, with the lowest Chamfer distance averaged over all scenes. Additionally, as visualized in Fig. 2 (Row 1), our reconstructed mesh can exhibit smooth surfaces and correctly maintain the structure, e.g., the hole in the mouth. (2) Our method achieves the best reconstruction performance in 10 out of 15 scenes, indicating its general effectiveness across diverse scenes.

Generalization to more diverse scenes. To demonstrate the generalization capability on more diverse scenes with larger domain shifts compared to the training scenes, we further benchmark the mesh reconstruction quality on the NeRF-Synthetic dataset [41] and NSVF-Synthetic [37] under a full-view reconstruction setting, i.e., the depth of each source view from each scene's training set is estimated and fused into a mesh using TSDF fusion [45, 63]. As visualized in Fig. 2 (Rows 2-4), we observe that our Omni-Recon can deliver notably higher-quality meshes with better-maintained structures, fewer holes, and smoother surfaces compared to the strongest baselines ReTR [34] and VolRecon [55]. The decent generalization capability of our method is attributed to the proper decoupling and fusion of geometry and appearance features in our backbone. More visualizations and ablation studies, such as the contributions of each backbone component, are provided in the supplementary materials.

Rendering quality. We further benchmark the performance of generalizable novel view rendering on the test scenes from DTU. As shown in Tab. 2, our

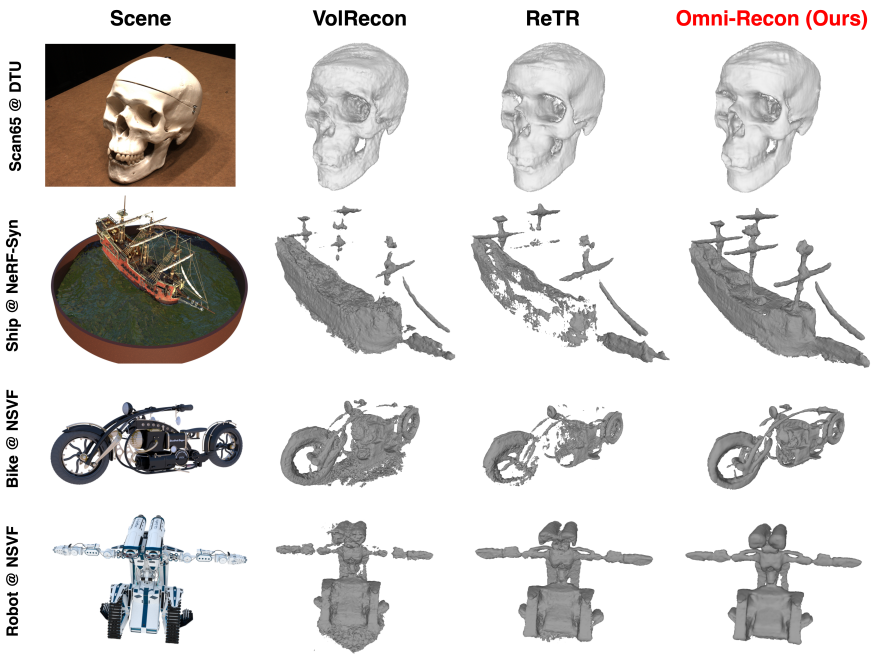

Fig. 2: Visualize the reconstructed mesh of our Omni-Recon and the two strongest baselines [34, 55]. Row 1: Scan65 from the test scenes of DTU; Rows 2-4: Scenes from NeRF-Synthetic [41] and NSVF-Synthetic [37], which present relatively challenging cases due to domain shifts w.r.t. the training scenes from DTU.

Table 2: Benchmark the quality of novel view rendering in terms of PSNR on DTU.

Method	Mean.	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122
VolRecon [55]	24.58	22.33	20.59	21.53	23.72	24.2	23.65	24.47	22.77	23.54	22.62	26.89	27.44	25.76	30.14	29.19
ReTR [34]	25.59	24.32	21.84	23.4	24.56	26.31	24.5	24.63	24.3	24.58	23.85	27.84	27.97	26.76	30.03	28.96
Omni-Recon	**26.32**	**24.77**	**22.33**	**23.92**	**25.56**	**26.37**	**24.75**	**25.19**	**24.94**	**24.92**	**25.06**	**28.39**	**28.63**	**28.01**	**31.49**	**30.5**

method achieves the highest PSNR, e.g., a +0.73 and +1.74 PSNR improvement over ReTR and VolRecon, respectively. As will be demonstrated in Sec. 5.4, this PSNR can be notably boosted with super-fast Nvdiffrast-based finetuning [31].

4 Omni-Recon: Enable Zero-shot Scene Understanding

To enable the understanding of various 3D scene properties, one approach is to jointly train the NeRF backbone on all target tasks. However, this approach is not scalable due to the diversity of scene properties, such as semantic segmentation, edge detection, and keypoint prediction, making it costly to cover new properties of interest. Therefore, it is highly desirable to enable zero-shot scene understanding for new tasks that were not seen during training.

4.1 Our Strategy: The Predict-then-Blend Strategy

Our hypothesis. We hypothesize that with accurate geometry and appearance estimation, the blending weights initially learned for radiance can be repurposed for other scene understanding tasks. In essence, just as blending source colors

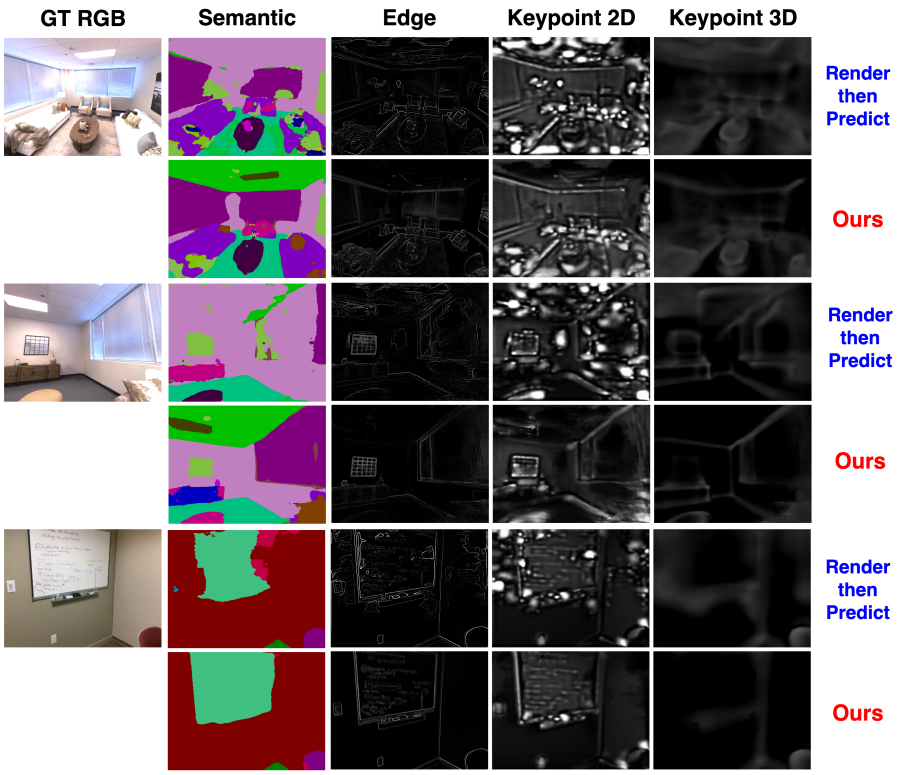

Fig. 3: Benchmark our predict-then-blend strategy with the render-then-predict baseline across different scene understanding tasks on Replica [61] and ScanNet [17].

yields radiance for each sampled 3D point, blending the properties of source views using the same weights can yield other properties. This hypothesis stems from the observation that scene appearance is closely related to other scene properties and thus regions with similar appearance are likely to share similar scene properties, such as semantics and edges. As such, the blending weights learned for appearance can also indicate the weighting for other scene properties.

Our method. Motivated by this hypothesis, our Omni-Recon employs a predict-then-blend strategy to enable zero-shot multitask scene understanding by lifting 2D monocular priors to 3D novel views. Specifically, considering the prevalence of 2D monocular vision models, our method first utilizes pretrained 2D models to generate predictions (e.g., logits in semantic segmentation) on each source view, denoted as $\{\mathbf{P}_i\}_{i=1}^N$. Then, the property of each sampled point can be obtained by blending their projected predictions on source views $\{\mathbf{p}_i\}_{i=1}^N$, reusing the RGB blending weights, i.e., $\hat{\mathbf{p}} = \sum_{i=1}^N \omega_i \mathbf{p}_i$. The final pixel-wise prediction can be derived using volumetric rendering similar to RGB reconstruction in Eq. 2.

4.2 Evaluation: Zero-shot Multitask Scene Understanding

Setup. To evaluate the zero-shot scene understanding capabilities of our model, we adopt two indoor 3D scene datasets: Replica [61] and ScanNet [17], following

the dataset splits in [93] and [36], respectively. Specifically, we focus on four scene understanding tasks: semantic segmentation, edge detection, and two keypoint detection tasks defined in [89]. For these tasks, the 2D monocular priors are provided by [51], [5], and [56], respectively. We employ mIoU for the semantic segmentation task and ℓ_1 error for the other three tasks as default metrics.

Baseline. An intuitive way to achieve zero-shot 3D scene understanding is to leverage existing 2D monocular models to predict the scene properties on top of the rendered images, a process referred to as a render-then-predict pipeline. We benchmark this strategy against our proposed predict-then-blend strategy.

Zero-shot scene understanding benchmark. As shown in Tab. 3 and Fig. 3, we observe that (1) Our predict-then-blend strategy achieves decent scene understanding performance on unseen scenes and unseen tasks, which echoes our hypothesis in Sec. 4.1 that 2D monocular priors can be elevated to 3D novel views through blending weight reuse, thanks to the high correlation between scene appearance and other properties; (2) Compared to the render-then-predict scheme, our method achieves notably higher quantitative accuracy, e.g., a 19.79% higher mIoU on ScanNet, and consistently better visual effects across all tasks. We attribute this improvement to two aspects: (1) Our predict-then-blend strategy can avoid feeding noisy inputs caused by rendering errors to the 2D monocular models, thus reducing reliance on the robustness of monocular models and ensuring more accurate monocular priors; (2) The multiview information gathered by our strategy can contribute to a more accurate understanding of objects with limited observations under a monocular setting. More experiments on other vision models like CLIP-LSeg [32] are in the supplementary materials.

Table 3: Benchmark the two zero-shot scene understanding strategies across tasks.

Dataset	Method	↑ Semantic	↓ Edge	↓ Key Point	↓ Key Point 3D
Replica [61]	Render-then -Predict	15.64	0.0456	0.1101	0.0470
	Predict-then -Blend (Ours)	**32.11**	**0.0412**	**0.0774**	**0.0176**
ScanNet [17]	Render-then -Predict	41.32	0.0471	0.0568	0.0412
	Predict-then -Blend (Ours)	**61.11**	**0.0434**	**0.0424**	**0.0197**

5 Omni-Recon: Adaptability to Downstream 3D Tasks

In addition to the generalization capabilities, we further demonstrate the adaptability of our Omni-Recon by extending it to three different types of downstream 3D applications. These demonstrations highlight the potentially broad utility of image-based rendering pipelines, which may have been overlooked by previous works, and can be generalized to a wider range of real-world 3D applications.

5.1 Application 1: A New Real-time Rendering Pipeline

To rapidly enable real-time rendering using Omni-Recon's pretrained backbone, we build a real-time rendering pipeline based on Nvdiffrast [31]. This involves baking the complex geometry branch into meshes using TSDF fusion [45, 63] as discussed in Sec. 3.5, and using the lightweight appearance branch as a shader.

Our real-time rendering pipeline. As illustrated in Fig. 1 (bottom left), for any novel view of a new scene, we first transform its camera pose to clip space as required by Nvdiffrast [31]. Then, we perform a rasterization process to obtain the intersection point of each camera ray with the extracted mesh. Next, these

intersection points are input into the shader to predict the color by blending their projected colors on the source views as described in Sec. 3.2. Thanks to the blending scheme, where valid projections can provide strong appearance clues, the rendering quality is often satisfactory even without finetuning.

Joint mesh and shader finetuning. To further enhance the rendering quality, we can jointly finetune the mesh and the shader in a gradient-based manner using differentiable rendering enabled by Nvdiffrast [31]. Specifically, we finetune the location of each vertex in the mesh and the parameters of the lightweight shader. Additionally, we periodically prune redundant mesh faces that cannot intersect with any camera rays in one training epoch. More details are provided in the supplementary materials.

5.2 Application 2: Parameter-Efficient Tuning for 3D Understanding

We explore the possibility of finetuning Omni-Recon for various downstream 3D tasks, utilizing a series of 3D scene understanding tasks as demonstrations. Specifically, we conduct PET [19] on top of Omni-Recon's pretrained backbone by incorporating low-rank adapters [26] into the linear layers within the transformer modules introduced in Sec. 3.3, and only tuning these adapters while keeping the pretrained backbone frozen. It is worth noting that when evaluating the performance of the tuned model, we assess it on new scenes that have never been encountered during both training and tuning, representing a generalizable scene understanding scenario [36].

5.3 Application 3: Text-guided 3D Scene Editing

We target text-guided 3D scene editing, which involves editing a 3D scene based on textual instructions. To achieve this, our Omni-Recon capitalizes on the nature of image-based rendering, allowing edits made to 2D source views to be propagated to 3D novel views. Specifically, leveraging the availability of powerful 2D diffusion models, we use publicly available ones to edit the 2D source views. However, doing so naively may result in inconsistent source views, as the same textual instruction may lead to different visualizations, thus degrading reconstruction quality. Inspired by [23], which iteratively edits the training images and retrains a new NeRF to achieve scene editing, we address the inconsistent editing issue by proposing an iterative editing and reconstruction pipeline.

The iterative editing and reconstruction pipeline. To ensure consistent 3D editing, we iterate between source view editing and reconstruction as follows: First, we edit each source view using the diffusion model InstructPix2Pix [4], which is guided by both textual instructions and the original image to maintain image fidelity. While the edited source views can faithfully follow the textual instructions, they may exhibit inconsistencies in the edited region. Next, we reconstruct each source view by conditioning on their nearby source views, as described in Sec. 3.2. This process improves the 3D consistency among source views at the expense of reduced image fidelity and less adherence to instructions. We then iterate through the editing and reconstruction steps to progressively enhance image fidelity, 3D consistency, and adherence to instructions. The resulting source views are finally used to reconstruct the entire scene.

Table 4: Benchmark the rendering speed and rendering quality of our real-time rendering pipeline and the baselines [34, 55] on DTU. The best PSNR is in **bold**, and the shortest finetuning time that our pipelines surpass the strongest baseline is underlined.

Method	FPS	Mean	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122
VolRecon [55]	0.029	24.58	22.33	20.59	21.53	23.72	24.2	23.65	24.47	22.77	23.54	22.62	26.89	27.44	25.76	30.14	29.19
ReTR [34]	0.024	25.59	24.32	21.84	23.4	24.56	26.31	24.5	24.63	24.3	24.58	23.85	27.84	27.97	26.76	30.03	28.96
Ours w/o ft.	71.3 (40.82)	22.96	20.12	19.71	22.27	22.78	24.55	21.77	21.51	26.72	22.33	24.49	22.52	22.93	24.68	23.84	24.12
Ours (ft. 10s)		25.68	21.42	21.68	24.06	24.12	28.19	24.10	23.95	31.65	24.41	28.15	25.63	25.85	26.15	26.82	26.89
Ours (ft. 20s)		27.21	22.63	22.92	25.12	25.42	30.03	25.95	26.16	33.19	26.22	30.36	27.44	27.04	26.93	29.15	29.65
Ours (ft. 30s)		27.78	23.2	23.26	25.54	25.7	30.59	26.83	26.96	33.66	26.47	30.73	28.14	27.70	27.1	30.17	30.65
Ours (ft. 1min)		28.34	24.69	24.11	25.76	26.05	30.93	27.66	27.49	33.68	27.07	30.97	28.51	28.54	27.35	31.17	31.54
Ours (ft. 3min)		28.95	25.2	24.32	**25.94**	26.16	32.15	28.99	**27.88**	34.94	**27.35**	31.62	28.93	28.97	27.56	31.70	32.49
Ours (ft. 5min)		**29.02**	**25.34**	**24.36**	25.63	**26.21**	**32.16**	**29.33**	27.81	**34.94**	27.32	**31.74**	**29.04**	**29.05**	**27.69**	**31.74**	**32.89**

5.4 Evaluation: Effectiveness on Different Downstream 3D Tasks

Real-time rendering. We evaluate our real-time rendering pipeline in Sec. 5.1 in terms of both rendering quality and rendering speed, measured on an NVIDIA A5000 GPU, using the DTU dataset. Specifically, we finetune the scene meshes, extracted by TSDF fusion as mentioned in Sec. 3.5, and the shader in no more than 5 minutes and record the rendering quality at different time steps.

Observations and analysis: As shown in Tab. 4, we can observe that (1) Without any finetuning, the rendering pipeline already exhibits certain rendering capabilities, despite the model backbone being trained using volumetric rendering with more than one sampled point; (2) Our rendering pipeline can be swiftly finetuned to significantly boost the PSNR. For instance, with just a 10-second finetuning, our rendering pipeline can match the average rendering quality of the strongest baseline, ReTR [34]; With 1-minute/2-minute finetuning, it can surpass ReTR by a +2.75 and +3.36 PSNR improvement on average, respectively; (3) Our rendering pipeline enables real-time rendering with 71.3 FPS, i.e., a >2458× speed-up over the baselines. Here, the assumption is that the feature extraction of each source view from a target scene can be done in a one-time effort and does not account for throughput. If we still consider this feature extraction latency for each rendering, the resulting 40.82 FPS still meets the real-time requirement.

These experiments indicate that *given a new scene*, our Omni-Recon is promising to enable both instant and accurate reconstruction and real-time rendering.

PET for 3D Understanding. To evaluate the adaptability of our Omni-Recon, we apply PET to it for various scene understanding tasks on ScanNet [17]. Specifically, we finetune a LoRA adaptor [26] for each task on a set of training scenes and evaluate on non-overlapping test scenes, representing a generalizable 3D understanding setting, following the dataset split described in [36]. We benchmark against the SOTA generalizable semantic segmentation work, SRay [36].

Table 5: Evaluate the effectiveness of PET on Omni-Recon and benchmark with SRay [36].

Method	↑ Sem. mIoU	↑ Sem. Total Acc	↑ Sem. Avg Acc	↓ Edge	↓ Key Point	↓ Key Point 3D
SRay [36]	57.15	78.24	62.55	-	-	-
Ours (Zero-shot)	61.11	80.80	69.17	0.0434	0.0424	0.0197
Ours (PET + Zero-shot)	**62.35**	**81.84**	**69.89**	**0.0342**	**0.0207**	**0.0132**

Observations and analysis: As shown in Tab. 5, we can observe that (1) After PET, our model outperforms SRay in terms of generalizable semantic segmentation with a +5.20% mIoU and a +7.34% accuracy improvement; (2) Despite

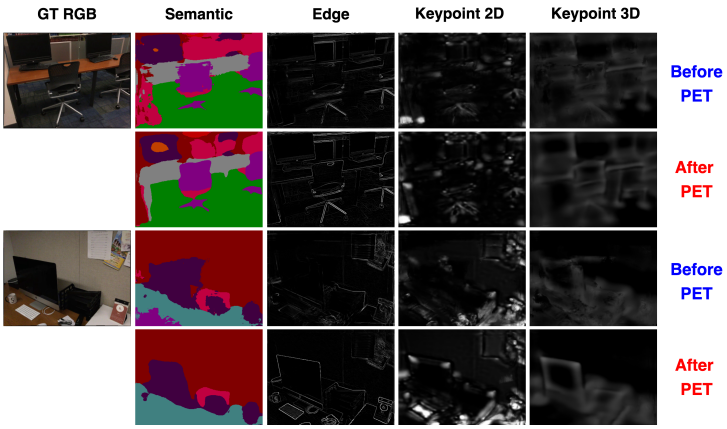

Fig. 4: Visualize the scene understanding performance on hard cases before/after PET.

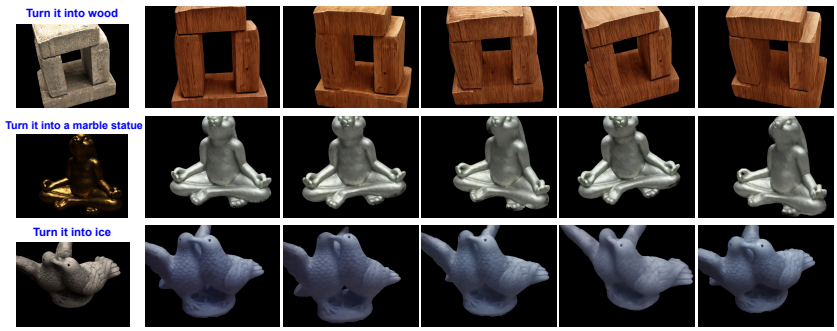

Fig. 5: Three text-guided scene editing examples using our pipeline in Sec. 5.3.

having limited trainable parameters, PET can enhance the scene understanding performance across tasks, especially on challenging cases visualized in Fig. 4, indicating the adaptability of Omni-Recon to diverse downstream tasks.

Text-guided scene editing. We visualize the edited examples in Fig. 5, showcasing the ability to ensure both instruction-following and 3D consistency.

6 Conclusion

Motivated by the significant demand for supporting various functionalities required by different 3D applications through a unified NeRF model, our work aims to develop a general-purpose NeRF capable of handling a broad spectrum of 3D tasks. We propose a framework called Omni-Recon, which features a general-purpose NeRF model using image-based rendering with two separate branches, and demonstrate its integration with various 3D tasks. Specifically, our Omni-Recon can achieve SOTA generalizable 3D reconstruction quality, enable zero-shot multitask scene understanding, achieve SOTA scene understanding performance and real-time rendering after rapid adaptation, and support scene editing. Omni-Recon underscores the potential of image-based rendering pipelines in diverse real-world 3D applications, which may have been underestimated by prior studies, thus sparking advancements in future rendering pipelines.

Acknowledgement

The work is supported by the National Science Foundation (NSF) through the SCH program (Award number: 1838873) and CoCoSys, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

References

1. Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B.: Large-scale data for multiple-view stereopsis. International Journal of Computer Vision **120**, 153–168 (2016)
2. 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)
3. 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)
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023)
5. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence (6), 679–698 (1986)
6. Cao, A., Johnson, J.: Hexplane: A fast representation for dynamic scenes. arXiv preprint arXiv:2301.09632 (2023)
7. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. CVPR (2021)
8. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. arXiv preprint arXiv:2203.09517 pp. 333–350 (2022)
9. 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)
10. 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. pp. 14124–14133 (2021)
11. Chen, T., Wang, P., Fan, Z., Wang, Z.: Aug-nerf: Training stronger neural radiance fields with triple-level physically-grounded augmentations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15191–15202 (2022)
12. Chen, Z., Funkhouser, T., Hedman, P., Tagliasacchi, A.: Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. arXiv preprint arXiv:2208.00277 (2022)
13. Chibane, J., Bansal, A., Lazova, V., Pons-Moll, G.: Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7911–7920 (2021)

14. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19. pp. 424–432. Springer (2016)
15. Cong, W., Liang, H., Wang, P., Fan, Z., Chen, T., Varma, M., Wang, Y., Wang, Z.: Enhancing nerf akin to enhancing llms: Generalizable nerf transformer with mixture-of-view-experts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3193–3204 (2023)
16. Croce, F., Hein, M.: Sparse and imperceivable adversarial attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4724–4732 (2019)
17. 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)
18. 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)
19. Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.M., Chen, W., et al.: Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence **5**(3), 220–235 (2023)
20. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5501–5510 (2022)
21. 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. arXiv preprint arXiv:2203.15224 (2022)
22. 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. pp. 14346–14355 (2021)
23. Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. arXiv preprint arXiv:2303.12789 (2023)
24. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics (2018)
25. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.: Baking neural radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5875–5884 (2021)
26. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
27. Hu, T., Liu, S., Chen, Y., Shen, T., Jia, J.: Efficientnerf efficient neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12902–12911 (2022)
28. 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. pp. 5885–5894 (2021)
29. 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. pp. 18365–18375 (2022)

30. Kundu, A., Genova, K., Yin, X., Fathi, A., Pantofaru, C., Guibas, L.J., Tagliasacchi, A., Dellaert, F., Funkhouser, T.: Panoptic neural fields: A semantic object-aware neural scene representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12871–12881 (2022)
31. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. ACM Transactions on Graphics (TOG) **39**(6), 1–14 (2020)
32. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=RriDjddCLN
33. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. CVPR (2021)
34. Liang, Y., He, H., Chen, Y.: Retr: Modeling rendering via transformer for generalizable neural surface reconstruction. Advances in Neural Information Processing Systems **36** (2024)
35. Lindell, D.B., Martel, J.N., Wetzstein, G.: Autoint: Automatic integration for fast neural rendering. CVPR (2021)
36. Liu, F., Zhang, C., Zheng, Y., Duan, Y.: Semantic ray: Learning a generalizable semantic field with cross-reprojection attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17386–17396 (2023)
37. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Advances in Neural Information Processing Systems **33**, 15651–15663 (2020)
38. 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. pp. 7824–7833 (2022)
39. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. SIGGRAPH (2019)
40. Long, X., Lin, C., Wang, P., Komura, T., Wang, W.: Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In: European Conference on Computer Vision. pp. 210–227. Springer (2022)
41. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)
42. 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)
43. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. arXiv preprint arXiv:2201.05989 **41**(4), 1–15 (2022)
44. Murez, Z., Van As, T., Bartolozzi, J., Sinha, A., Badrinarayanan, V., Rabinovich, A.: Atlas: End-to-end 3d scene reconstruction from posed images. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16. pp. 414–431. Springer (2020)
45. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE international symposium on mixed and augmented reality. pp. 127–136. Ieee (2011)

46. 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)
47. 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)
48. Oechsle, M., Peng, S., Geiger, A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5589–5599 (2021)
49. Ost, J., Mannan, F., Thuerey, N., Knodt, J., Heide, F.: Neural scene graphs for dynamic scenes. CVPR (2021)
50. 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. pp. 10318–10327 (2021)
51. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. ArXiv preprint (2021)
52. Rebain, D., Jiang, W., Yazdani, S., Li, K., Yi, K.M., Tagliasacchi, A.: DeRF: Decomposed radiance fields. CVPR (2021)
53. 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)
54. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10901–10911 (2021)
55. Ren, Y., Zhang, T., Pollefeys, M., Süsstrunk, S., Wang, F.: Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16685–16695 (2023)
56. Sax, A., Emi, B., Zamir, A.R., Guibas, L.J., Savarese, S., Malik, J.: Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies. (2018)
57. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. pp. 501–518. Springer (2016)
58. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: GRAF: Generative radiance fields for 3D-aware image synthesis. NeurIPS (2020)
59. Schwarz, K., Sauer, A., Niemeyer, M., Liao, Y., Geiger, A.: Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. arXiv preprint arXiv:2206.07695 (2022)
60. Seitz, S.M., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. IJCV (1999)
61. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
62. 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. pp. 5459–5469 (2022)

63. 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)
64. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8248–8258 (2022)
65. Tang, J., Zhou, H., Chen, X., Hu, T., Ding, E., Wang, J., Zeng, G.: Delicate textured mesh recovery from nerf via adaptive surface refinement. arXiv preprint arXiv:2303.02091 (2023)
66. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. ACM Transactions on Graphics (2019)
67. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. Acm Transactions on Graphics (TOG) **38**(4), 1–12 (2019)
68. Trevithick, A., Yang, B.: Grf: Learning a general radiance field for 3d representation and rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15182–15192 (2021)
69. Truong, P., Rakotosaona, M.J., Manhardt, F., Tombari, F.: Sparf: Neural radiance fields from sparse and noisy poses. arXiv preprint arXiv:2211.11738 (2022)
70. Turki, H., Ramanan, D., Satyanarayanan, M.: Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12922–12931 (2022)
71. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30**, 5998–6008 (2017)
72. 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)
73. Vora, S., Radwan, N., Greff, K., Meyer, H., Genova, K., Sajjadi, M.S., Pot, E., Tagliasacchi, A., Duckworth, D.: Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. arXiv preprint arXiv:2111.13260 (2021)
74. 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. pp. 3835–3844 (2022)
75. Wang, P., Chen, X., Chen, T., Venugopalan, S., Wang, Z., et al.: Is attention all nerf needs? arXiv preprint arXiv:2207.13298 (2022)
76. 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)
77. 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. pp. 4690–4699 (2021)
78. Xiangli, Y., Xu, L., Pan, X., Zhao, N., Rao, A., Theobalt, C., Dai, B., Lin, D.: Citynerf: Building nerf at city scale. arXiv preprint arXiv:2112.05504 (2021)
79. Xiangli, Y., Xu, L., Pan, X., Zhao, N., Rao, A., Theobalt, C., Dai, B., Lin, D.: Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII. pp. 106–122. Springer (2022)

80. Xu, D., Jiang, Y., Wang, P., Fan, Z., Shi, H., Wang, Z.: Sinnerf: Training neural radiance fields on complex scenes from a single image. arXiv preprint arXiv:2204.00928 (2022)
81. Xu, D., Jiang, Y., Wang, P., Fan, Z., Wang, Y., Wang, Z.: Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360 {\deg} views. arXiv preprint arXiv:2211.16431 (2022)
82. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5438–5448 (2022)
83. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018)
84. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. Advances in Neural Information Processing Systems **34**, 4805–4815 (2021)
85. 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)
86. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. Advances in Neural Information Processing Systems **33**, 2492–2502 (2020)
87. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: PlenOctrees for real-time rendering of neural radiance fields. In: ICCV (2021)
88. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4578–4587 (2021)
89. Zamir, A.R., Sax, A., , Shen, W.B., Guibas, L., Malik, J., Savarese, S.: Taskonomy: Disentangling task transfer learning. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2018)
90. Zhang, M., Zheng, S., Bao, Z., Hebert, M., Wang, Y.X.: Beyond rgb: Scene-property synthesis with neural radiance fields. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 795–805 (2023)
91. Zhang, X., Bi, S., Sunkavalli, K., Su, H., Xu, Z.: Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5449–5458 (2022)
92. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16259–16268 (2021)
93. Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.J.: In-place scene labelling and understanding with implicit scene representation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15838–15847 (2021)