Semantic Ray: Learning a Generalizable Semantic Field with Cross-Reprojection Attention

Fangfu Liu[1,3], Chubin Zhang[2], Yu Zheng[2], Yueqi Duan[1†]
[1]Department of Electronic Engineering, Tsinghua University
[2]Department of Automation, Tsinghua University
[3]Beijing National Research Center for Information Science and Technology
{liuff19,zhangcb19,zhengyu19}@mails.tsinghua.edu.cn, duanyueqi@tsinghua.edu.cn

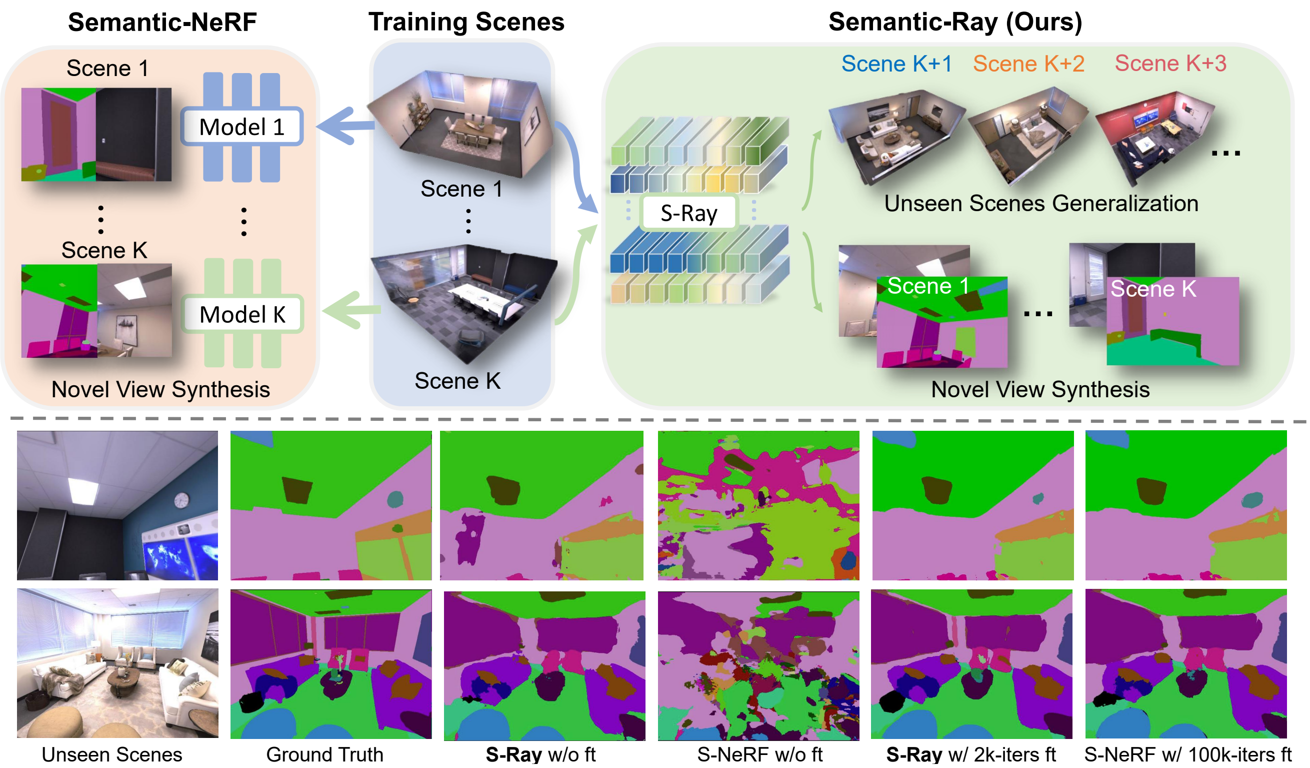

Figure 1. **Top**: Comparisons between Semantic-NeRF [61] and our method Semantic-Ray. Semantic-NeRF (S-NeRF for short) needs to train one specific model for each scene, while our Semantic-Ray (S-Ray for short) trains one unified model on multiple scenes and generalizes to unseen scenes. **Bottom**: Experimental comparisons between S-Ray and S-NeRF on generalization ability. We observe that our network S-Ray can effectively *fast generalize* across diverse unseen scenes while S-NeRF fails in a new scene. Moreover, our result can be improved by fine-tuning on more images for only 10 min ($2k$ iterations), which achieves comparable quality with the Semantic-NeRF's result for $100k$ iterations per-scene optimization.

Abstract

In this paper, we aim to learn a semantic radiance field from multiple scenes that is accurate, efficient and generalizable. While most existing NeRFs target at the tasks of neural scene rendering, image synthesis and multi-view reconstruction, there are a few attempts such as Semantic-NeRF that explore to learn high-level semantic understanding with the NeRF structure. However, Semantic-NeRF simultaneously learns color and semantic label from a single ray with multiple heads, where the single ray fails to provide rich semantic information. As a result, Semantic NeRF relies on positional encoding and needs to train one specific model for each scene. To address this, we propose Semantic Ray (S-Ray) to fully exploit semantic information along the ray direction from its multi-view reprojections. As directly

†Corresponding author.

performing dense attention over multi-view reprojected rays would suffer from heavy computational cost, we design a Cross-Reprojection Attention module with consecutive intra-view radial and cross-view sparse attentions, which decomposes contextual information along reprojected rays and cross multiple views and then collects dense connections by stacking the modules. Experiments show that our S-Ray is able to learn from multiple scenes, and it presents strong generalization ability to adapt to unseen scenes. Project page: https://liuff19.github.io/S-Ray/.

1. Introduction

Recently, Neural Radiance Field (NeRF) [32], a new novel view synthesis method with implicit representation, has taken the field of computer vision by storm [11]. NeRF and its variants [2, 32, 57, 59] adopt multi-layer perceptrons (MLPs) to learn continuous 3D representations and utilize multi-view images to render unseen views with fine-grained details. NeRF has shown state-of-the-art visual quality, produced impressive demonstrations, and inspired many subsequent works [4, 19, 20, 53, 57].

While the conventional NeRFs have achieved great success in low- and middle-level vision tasks such as neural scene rendering, image synthesis, and multi-view reconstruction [4, 12, 27, 35, 36, 42, 50], it is interesting to explore their more possibilities in high-level vision tasks and applications. Learning high-level semantic information from 3D scenes is a fundamental task of computer vision with a wide range of applications [10, 13, 15, 33]. For example, a comprehensive semantic understanding of scenes enables intelligent agents to plan context-sensitive actions in their environments. One notable attempt to learn interpretable semantic understanding with the NeRF structure is Semantic-NeRF [61], which regresses a 3D-point semantic class together with radiance and density. Semantic-NeRF shows the potential of NeRF in various high-level tasks, such as scene-labeling and novel semantic view synthesis.

However, Semantic-NeRF follows the vanilla NeRF by estimating the semantic label from a single ray with a new semantic head. While this operation is reasonable to learn low-level information including color and density, a single ray fails to provide rich semantic patterns – we can tell the color from observing a single pixel, but not its semantic label. To deal with this, Semantic-NeRF heavily relies on positional encoding to learn semantic features, which is prone to overfit the current scene and only applicable to novel *views* within the same scene [51]. As a result, Semantic-NeRF has to train one model from scratch for every scene independently or provides very limited novel scene generalization by utilizing other pretrained models to infer 2D segmentation maps as training signals for unseen scenes. This significantly limits the range of applications in real-world scenarios.

In this paper, we propose a neural semantic representation called **Semantic Ray** (S-Ray) to build a generalizable semantic field, which is able to learn from multiple scenes and directly infer semantics on novel viewpoints across novel scenes as shown in Figure 1. As each view provides specific high-level information for each ray regarding of viewpoints, occlusions, etc., we design a Cross-Reprojection Attention module in S-Ray to fully exploit semantic information from the reprojections on multiple views, so that the learned semantic features have stronger discriminative power and generalization ability. While directly performing dense attention over the sampled points on each reprojected ray of multiple views would suffer from heavy computational costs, we decompose the dense attention into intra-view radial and cross-view sparse attentions to learn comprehensive relations in an efficient manner.

More specifically, for each query point in a novel view, different from Semantic-NeRF that directly estimates its semantic label with MLP, we reproject it to multiple known views. It is worth noting that since the emitted ray from the query point is virtual, we cannot obtain the exact reprojected point on each view, but a reprojected ray that shows possible positions. Therefore, our network is required to simultaneously model the uncertainty of reprojection within each view, and comprehensively exploit semantic context from multiple views with their respective significance. To this end, our Cross-Reprojection Attention consists of an intra-view radial attention module that learns the relations among sampled points from the query ray, and a cross-view sparse attention module that distinguishes the same point in different viewpoints and scores the semantic contribution of each view. As a result, our S-Ray is aware of the scene prior with rich patterns and generalizes well to novel scenes. We evaluate our method quantitatively and qualitatively on synthetic scenes from the Replica dataset [46] and real-world scenes from the ScanNet dataset [7]. Experiments show that our S-Ray successfully learns from multiple scenes and generalizes to unseen scenes. By following Semantic-NeRF [61], we design competitive baselines based on the recent MVSNeRF [4] and NeuRay [27] architectures for generalizable semantic field learning. Our S-Ray significantly outperforms these baselines which demonstrates the effectiveness of our cross-reprojection attention module.

2. Related Work

Semantic Segmentation. Semantic segmentation is one of the high-level tasks that paves the way toward complete scene understanding, with most methods targeting a fully supervised, single-modality problem (2D [1, 5, 6, 28, 43] or 3D [3, 14, 34, 62]). Recently, machine learning methods have proven to be valuable in semantic segmentation [1, 17, 43, 49] which aims to assign a separate class label

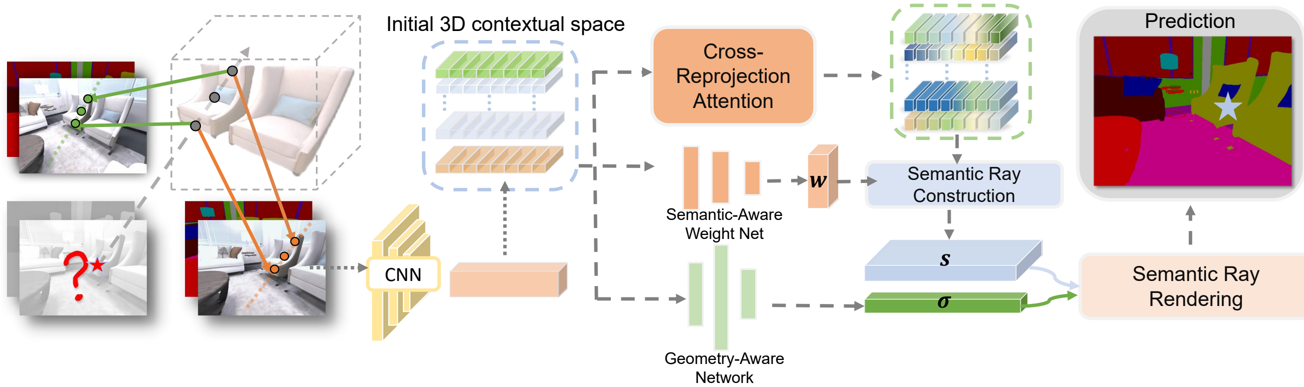

Figure 2. **Pipeline of semantic rendering with S-Ray**. Given input views and a query ray, we first reproject the ray to each input view and apply a CNN-based module to extract contextual features to build an initial 3D contextual space (Sec. 3.2). Then, we apply the Cross-Reprojection Attention module to learn dense semantic connections and build a refined contextual space (Sec. 3.3). For semantic ray rendering, we adopt the semantic-aware weight net to learn the significance of each view to construct our semantic ray from refined contextual space (Sec. 3.4). Finally, we leverage the geometry-aware net to get the density and render the semantics along the query ray.

to each pixel of an image. However, most methods suffer from severe performance degradation when the scenes observed at test time mismatch the distribution of the training data [23, 58]. To alleviate the issue, 2D-based architectures [13, 28, 43] are often trained on large collections of costly annotated data [24] while most 3D prior works [9, 16, 18, 22, 39, 40] rely on 3D sensors. Though straightforward to apply, 2D-based approaches only produce per-pixel annotations and fail to understand the 3D structure of scenes [51] and 3D sensors are too expensive to be widely available as RGB cameras. In contrast to these methods, our method reconstructs and then segments a 3D semantic representation from 2D inputs and supervision alone without ground truth 3D annotations or input geometry.

Neural Radiance Fields. Recently, implicit neural representations have advanced neural processing for 3D data and multi-view 2D images [31, 37, 45, 55]. In particular, Neural Radiance Fields (NeRF) [32] has drawn great attention, which is a fully-connected neural network that can generate novel views of complex 3D scenes, based on a partial set of 2D images. A NeRF network aims to map from 3D location and viewing direction (5D input) to opacity and color (4D input). Several following works emerge trying to address its limitations and improve its performance, including fast training [8, 48], efficient inference [12, 25, 41, 42, 56], unbounded scenes training [2, 59], better generalization [20, 44, 47, 50, 53, 57], generative modeling [30, 35, 44], editing [19, 26, 52]. As NeRF achieves very impressive results for novel view synthesis, researchers start to explore high-level tasks in NeRF such as semantic segmentation [51, 61]. However, Semantic-NeRF [61] is only applicable in the single-scene setting while NeSF [51] only conducts experiments in synthetic data with insufficient generalization and high computational costs. In contrast, taking NeRF as a powerful implicit scene representation, our method can learn a generalizable semantic representation of new, real-world scenes with high quality.

3. Method

Given a set of N input views with known camera poses, our goal is to synthesize novel semantic views from arbitrary angles across unseen scenes with strong generalizability. Our method can be divided into three stages: (1) build the 3D contextual space from source multiple views, (2) decompose the semantic information from 3D space along reprojected rays and cross multiple views with cross-reprojection attention, (3) reassemble the decomposed contextual information to collect dense semantic connections in 3D space and re-render the generalizable semantic field. Our pipeline is depicted in Figure 2. Before introducing our S-Ray in detail, we first review the volume rendering on a radiance field [29, 32].

3.1. Preliminaries

Neural Volume Rendering. Neural volume rendering aims to learn two functions: $\sigma(\mathbf{x};\theta) : \mathbb{R}^3 \mapsto \mathbb{R}$, which maps the spatial coordinate $\mathbf{x}$ to a density σ, and $\mathbf{c}(\mathbf{x},\mathbf{d};\theta) : \mathbb{R}^3 \times \mathbb{R}^3 \mapsto \mathbb{R}^3$ that maps a point with the viewing direction to the RGB color $\mathbf{c}$. The density and radiance functions are defined by the parameters θ. To learn these functions, they are evaluated through a ray emitted from a query view. The ray is parameterized by $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, $t \in [t_n, t_f]$, where $\mathbf{o}$ is the start point at the camera center, $\mathbf{d}$ is the unit direction vector of the ray, and $[t_n, t_f]$ is the near-far bound along the ray. Then, the color for the associated pixel of this ray can be computed through volume rendering [29]:

$$\hat{\mathbf{C}}(\mathbf{r};\theta) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})\mathrm{d}t, \tag{1}$$

where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\mathrm{d}s\right)$. In practice, the continuous integration can be approximated by a summation of discrete samples along the ray by the quadrature rule. For selected N random quadrature points $\{t_k\}_{k=1}^{N}$ between t_n and t_f, the approximated expected color is computed by:

$$\hat{\mathbf{C}}(\mathbf{r};\theta) = \sum_{k=1}^{N} T(t_k)\alpha(\sigma(t_k)\delta_k)\mathbf{c}(t_k), \tag{2}$$

where $T(t_k) = \exp\left(-\sum_{k'=1}^{k-1} \sigma(t_k)\delta_k\right)$, $\alpha(x) = 1 - exp(-x)$, and $\delta_k = t_{k+1} - t_k$ are intervals between sampled points.

3.2. Building 3D Contextual Space across Views

NeRF-based generalization rendering methods [50, 53, 57] construct a radiance field by reprojecting a single point of the query ray to source views and extracting the point-based feature. It is reasonable to predict color from a single point but is quite insufficient for semantics which needs more contextual information. Therefore, we build the 3D contextual space to learn rich semantic patterns by reprojecting the whole query ray across views. Given the N points $\{\mathbf{p}_i\}_{i=1,2,...,N}$ sampled from the ray emitting from the query view and known camera pose (*i.e.*, the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$). Without losing generality, we can rewrite the ray as:

$$\mathbf{r}(z) = \mathbf{p}_i + z\frac{\mathbf{p}_j - \mathbf{p}_i}{||\mathbf{p}_j - \mathbf{p}_i||}, \quad z \in \mathbb{R}. \tag{3}$$

Then, the ray warping function is defined as:

$$w(\mathbf{r}(z), \mathbf{R}, \mathbf{t}) := \mathbf{K}\pi(\mathbf{R}\cdot\mathbf{r}(z) + \mathbf{t}), \tag{4}$$

which allows reprojecting the ray onto source views to obtain the plane ray $\mathbf{r}^*(z)$, *i.e.*, $\mathbf{r}^*(z) = w(\mathbf{r}(z), \mathbf{R}, \mathbf{t})$, where $\mathbf{K}$ is the camera calibration matrix and $\pi(\mathbf{u}) := [\mathbf{u}_x/\mathbf{u}_z, \mathbf{u}_y/\mathbf{u}_z]$ is the projection function.

Let $\mathcal{F}_j(\mathbf{r}^*(z))(j = 1, 2, ..., m)$ denote the ray-based feature of the query ray reprojected on the j-th source view, $\mathcal{F}_j(\mathbf{r}^*(z_i))$ be the $\mathbf{p}_i$ point-based feature within the j-th source view. Due to the permutation invariance of the source views, we use a shared U-Net-based convolutional neural network to extract dense contextual information $\mathcal{F}$ from these views. Then, we build our 3D contextual space $\mathcal{M}$ across views as:

$$\mathcal{M} = \begin{bmatrix} \mathcal{F}_1(\mathbf{r}^*(z_1)) & \mathcal{F}_1(\mathbf{r}^*(z_2)) & \cdots & \mathcal{F}_1(\mathbf{r}^*(z_N)) \\ \mathcal{F}_2(\mathbf{r}^*(z_1)) & \mathcal{F}_2(\mathbf{r}^*(z_2)) & \cdots & \mathcal{F}_2(\mathbf{r}^*(z_N)) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{F}_m(\mathbf{r}^*(z_1)) & \mathcal{F}_m(\mathbf{r}^*(z_2)) & \cdots & \mathcal{F}_m(\mathbf{r}^*(z_N)) \end{bmatrix} \tag{5}$$

which is a 3D matrix (*i.e.*, $\mathcal{M} \in \mathbb{R}^{m\times N\times C}$) to describe a full space contextual information around the query ray with feature dimension C.

3.3. Cross-Reprojection Attention

To model full semantic-aware dependencies from above 3D contextual space $\mathcal{M}$, a straightforward approach is to perform dense attention over $\mathcal{M}$. However, it would suffer from heavy computational costs. To address the problem, we propose Cross-Reprojection Attention as shown in Figure 3, including intra-view radial attention and cross-view sparse attention to approximate dense semantic connections in 3D space with lightweight computation and memory.

Intra-view Radial Attention. First, we rewrite the contextual space as $\mathcal{M} = [\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_m]^T$, where $\mathcal{F}_i = [\mathcal{F}_i(\mathbf{r}^*(z_1)), \mathcal{F}_i(\mathbf{r}^*(z_2)), ..., \mathcal{F}_i(\mathbf{r}^*(z_N))]$ is the radial feature in the i-th view. Then, we decompose the 3D contextual space $\mathcal{M}$ along the radial direction in source views, *i.e.*, consider the $\mathcal{F}_i(i = 1, ..., m)$ which encodes the intra-view contextual information within each view. Taking the $\mathcal{F}_i \in \mathbb{R}^{N\times C}$ as input, our intra-view radial attention with H heads is formulated as

$$Q_R = \mathcal{F}_i W_q, \quad K_R = \mathcal{F}_i W_k, \quad V_R = \mathcal{F}_i W_v, \tag{6}$$

$$A^{(h)} = \sigma(\frac{Q_R^{(h)} K_R^{(h)T}}{\sqrt{d_k}})V_R^{(h)}, h = 1, ..., H, \tag{7}$$

$$f(Q_R, K_R, V_R) = \mathrm{Concat}(A^{(1)}, ..., A^{(H)})W_o, \tag{8}$$

where $\sigma(\cdot)$ denotes the softmax function, and $d_k = C/H$ is the dimension of each head. $A^{(h)}$ denotes the embedding output from the h-th attention head, $Q_R^{(h)}, K_R^{(h)}, V_R^{(h)} \in \mathbb{R}^{N\times d_k}$ denote query, key, and value of radial embeddings respectively. $W_q, W_k, W_v, W_o \in \mathbb{R}^{C\times C}$ are the projection matrices. Then, we obtain a refined $\mathcal{F}_i^{'}$ as

$$\mathcal{F}_i^{'} = f(Q_R, K_R, V_R), \tag{9}$$

which contains global semantic-aware patterns along the reprojected ray in i-th view. Similarly, we apply this intra-view radial attention module to each $\mathcal{F}_i(i = 1, ..., m)$ to refine the 3D contextual space, denoted as $\mathcal{M}^{'} = [\mathcal{F}_1^{'}, \mathcal{F}_2^{'}, ..., \mathcal{F}_m^{'}]^T$.

Cross-view Sparse Attention. After the intra-view radial attention module, we decompose $\mathcal{M}^{'}$ cross multiple views and rewrite $\mathcal{M}^{'} = [\mathcal{F}_{\mathbf{r}^*(1)}^{'}, ..., \mathcal{F}_{\mathbf{r}^*(N)}^{'}]$, where $\mathcal{F}_{\mathbf{r}^*(\mathbf{i})}^{'} = [\mathcal{F}_1^{'}(\mathbf{r}^*(z_i)), ..., \mathcal{F}_m^{'}(\mathbf{r}^*(z_i))]^T$, which encodes the global ray-based feature in each view. Aiming to exploit semantic information from multiple views with their respective significance which is sparse, we put $\mathcal{M}^{'}$ to the cross-view sparse attention module. Following the predefined notation, we compute the $\mathcal{F}_{\mathbf{r}^*(\mathbf{i})}^{''}$ from

$$\mathcal{F}_{\mathbf{r}^*(\mathbf{i})}^{''} = f(\mathcal{F}_{\mathbf{r}^*(\mathbf{i})}^{'}\widetilde{W}_q, \mathcal{F}_{\mathbf{r}^*(\mathbf{i})}^{'}\widetilde{W}_k, \mathcal{F}_{\mathbf{r}^*(\mathbf{i})}^{'}\widetilde{W}_v), \tag{10}$$

where $\widetilde{W}_q, \widetilde{W}_k, \widetilde{W}_v$ are the projection matrices in cross-view sparse attention. Therefore, we get our final 3D contextual space $\mathcal{M}^{''} = [\mathcal{F}_{\mathbf{r}^*(1)}^{''}, ..., \mathcal{F}_{\mathbf{r}^*(N)}^{''}]$, which collects dense semantic connections around the query ray.

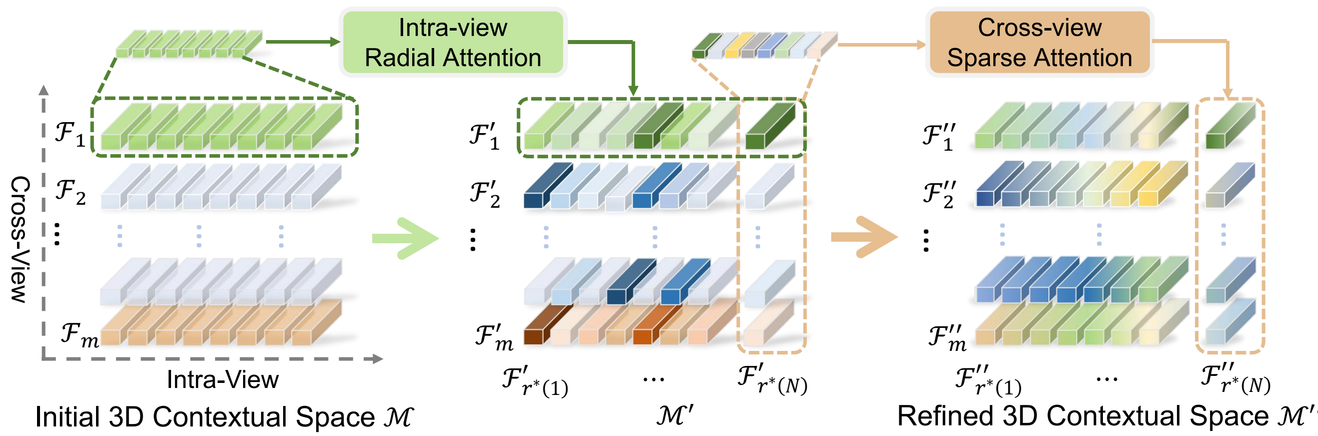

Figure 3. **Pipeline of Cross-Reprojection Attention**. Given the initial 3D contextual space $\mathcal{M}$ form Sec. 3.2, we first decompose $\mathcal{M}$ along the radial direction (*i.e.*, each intra-view). Then, we apply the *intra-view radial attention module* to each $\mathcal{F}_i(i = 1, ...m)$ to learn the ray-aligned contextual feature from each source view and build the $\mathcal{M}^{'}$. We further decompose the $\mathcal{M}^{'}$ cross multiple views and employ the *cross-view sparse attention module* to each $\mathcal{F}^{'}_{\mathbf{r}^*(i)}$, thus capturing sparse contextual patterns with their respective significance to semantics. After the two consecutive attention modules, we fuse the decomposed contextual information with the final refined 3D contextual space $\mathcal{M}^{''}$, which models dense semantic collections around the ray. (*Best viewed in color*)

3.4. Semantic Ray

Semantic Ray Construction. As done in the previous pipeline, we have built a semantic-aware space $\mathcal{M}^{''} = [\mathcal{F}^{''}_1, \mathcal{F}^{''}_2, ..., \mathcal{F}^{''}_m]^T$ which encodes refined 3D contextual patterns around the light ray emitted from the query view. To construct our final semantic ray from $\mathcal{M}^{''}$ and better learn the semantic consistency along the ray, we introduce a *Semantic-aware Weight Network* to rescore the significance of each source view. Then, we can assign distinct weights to different views and compute the final semantic ray $\mathbf{s}$ as

$$\mathbf{s} = w_1\mathcal{F}^{''}_1 + w_2\mathcal{F}^{''}_2 + ... + w_m\mathcal{F}^{''}_m, \tag{11}$$

$$\mathbf{w} \in \mathbb{C}(\tau) := \left\{ \mathbf{w} : 0 < \frac{\tau}{m} < w_i < \frac{1}{\tau m}, \sum_{i=1}^{m} w_i = 1 \right\}, \tag{12}$$

where $\mathbf{w}$ is the view reweighting vector with length m indicating the importance of source views, and τ is the small constant with $\tau > 0$. The deviation of the weight distribution from the uniform distribution is bound by the hyperparameter τ, which keeps the semantic consistency instead of bias across views.

Semantic Field Rendering. Finally, we use the rendering scheme introduced in NeRF to render semantic logits from the ray $\mathbf{r}$ with N sampled points, namely $\{z_k\}_{k=1}^{N}$. The semantic logit $\hat{\mathbf{S}}(\mathbf{r})$ is defined as

$$\hat{\mathbf{S}}(\mathbf{r}) = \sum_{k=1}^{N} T(z_k)\{1 - \exp(-\sigma(z_k)\delta_k)\}\mathbf{s}(z_k), \tag{13}$$

where $T(z_k) = \exp(-\sum_{k'=1}^{k-1} \sigma(z_k)\delta_k)$, $\delta_k = z_{k+1} - z_k$ is the distance between two adjacent quadrature points along the semantic ray and σ is predicted by a *Geometry-aware Network*.

Network Training. More specifically, we discuss the formulation of the semantic loss functions. We apply our S-Ray with a set of RGB images with known camera parameters denoted by $\mathcal{I}$. The losses are computed for the set of all rays denoted as $\mathcal{R}$, which is emitted from the query image $I \in \mathcal{I}$. The semantic loss is computed as multi-class cross-entropy loss to encourage the rendered semantic labels to be consistent with the provided labels, where $1 \leq l \leq L$ denotes the class index

$$\mathcal{L}_{sem}(I) = -\sum_{\mathbf{r}\in\mathcal{R}} \left[\sum_{l=1}^{L} p^l(\mathbf{r}) \log \hat{p}^l(\mathbf{r}) \right], \tag{14}$$

where $p^l, \hat{p}^l$ are the multi-class semantic probability at class l of the ground truth map. Unlike Semantic-NeRF [61] which needs heavy prior training for the radiance only in a single scene, we can train our S-Ray Network with semantic supervision in multiple scenes simultaneously and fast generalizes to a novel scene.

3.5. Discussion and Implementation

Discussion with GPNR [47]. The recent work GPNR [47] shares a similar motivation by aggregating features along epipolar line. It is proposed for color rendering with carefully-designed positional encoding to encode information of view direction, camera pose, and location. In contrast, we focus on learning a generalizable semantic field for semantic rendering through merely image features. In this sense, S-Ray creates a neat design space without any positional encoding engineering. While GPNR requires training

Method	Settings	Synthetic Data (Replica [46])			Real Data (ScanNet [7])		
		mIoU↑	Total Acc↑	Avg Acc↑	mIoU↑	Total Acc↑	Avg Acc↑
MVSNeRF [4] + Semantic Head	Generalization	23.41	54.25	33.70	39.82	60.01	46.01
NeuRay [27] + Semantic Head		35.90	69.35	43.97	51.03	77.61	57.12
S-Ray (Ours)		**41.59**	**70.51**	**47.19**	**57.15**	**78.24**	**62.55**
Semantic-NeRF [61]	Finetuning	75.06	94.36	70.20	**91.24**	97.54	93.89
MVSNeRF [4] + Semantic Head$_{ft}$		53.77	79.48	62.85	55.26	76.25	69.70
NeuRay [27] + Semantic Head$_{ft}$		63.73	85.54	70.05	77.48	91.56	81.04
S-Ray$_{ft}$ (Ours)		**75.96**	**96.38**	**80.81**	91.08	**98.20**	**93.97**

Table 1. **Quantitative comparison.** We show averaged results of mIoU, Total Acc, and Average Acc (higher is better) as explained in Sec. 4.1. On the top, we compare S-Ray (Ours) with NeuRay [27]+semantic head and MVSNeRF [4]+semantic head with direct network inference. On the bottom, we show our results with only 10 minutes of optimization.

24 hours on 32 TPUs, S-Ray only needs a single RTX3090-Ti GPU with similar training time.

Implementation details. Given multiple views of a scene, we construct a training pair of source and query view by first randomly selecting a target view, and sampling m nearby but sparse views as source views. We follow [27] to build our sampling strategy, which simulates various view densities during training, thus helping the network generalize across view densities. We implement our model in PyTorch [38] and train it end-to-end on a single RTX3090-Ti GPU with 24GB memory. The batch size of rays is set to 1024 and our S-Ray is trained for $250k$ steps using Adam [21] with an initial learning rate of $1e-3$ decaying to $5e-5$. S-Ray is able to generalize well to novel scenes and can also be finetuned per scene using the same objective in (14). More details of network training, architecture design, and hyperparameter settings can be found in the supplementary.

4. Experiments

We conduct extensive experiments and evaluate our method with basically two settings. 1) We directly evaluate our pretrained model on test scenes (*i.e*., unseen scenes) without any finetuning. Note that we train only *one* model called S-Ray and evaluate it on all test scenes. 2) We finetune our pretrained model for a small number of steps on each unseen scene before evaluation. While training from scratch usually requires a long optimization time, we evaluate our S-Ray by achieving comparable performance with well-trained models by much less finetuning time.

4.1. Experiment Setup

Datasets. To test the effectiveness of our method comprehensively, we conduct experiments on both synthetic data and real data. For synthetic data, we use the Replica [46], a dataset with 18 highly photo-realistic 3D indoor scene reconstructions at room and building scale. Each Scene consists of dense geometry, high-dynamic-range textures, and per-primitive semantic class. Then, we choose 12 scenes from the Replica as training datasets and the remaining unseen scenes as test datasets. For Real data, we use the ScanNet [7], which is a real-world large labeled RGB-D dataset containing 2.5M views in 1513 scenes annotated with 3D camera poses surface reconstructions and semantic segmentation. We choose 60 different scenes as training datasets and 10 unseen novel scenes as test datasets to evaluate generalizability in real data. More details about the split of datasets will be shown in the supplementary.

Metrics. To accurately measure the performance of our S-Ray, we adopt mean Intersection-over-Union (mIoU) as well as average accuracy and total accuracy to compute segmentation quality. What's more, we will discuss our rendering quality if we compute the color from the *Geometry-Aware Network* in the posterior subsection. To evaluate rendering quality, we follow NeRF [32], adopting peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM) [54], and learned perceptual image patch similarity (LPIPS) [60].

Baselines. To evaluate our fast generalizability in an unseen scene, we choose Semantic-NeRF [61] as our baseline. Since we are the first to learn a generalizable semantic field in real-world scenes, we have no baselines to compare our generalizable performance. In order to further show our strong generalizability, we add the semantic head by following Semantic-NeRF settings to the NeRF-based methods which have shown generalizability in the reconstruction task. Specifically, we compare our method against MVSNeRF [4] with semantic head and NeuRay [27] with semantic head in generalization and finetuning. Due to the space limitation, We provide a more detailed discussion and comparison with [4, 27, 47, 53, 61] in the supplementary.

4.2. Comparison with Baselines

To render each semantic map of the test view, we sample 8 source views from the training set for all evaluation datasets in generalization settings. For the per-scene

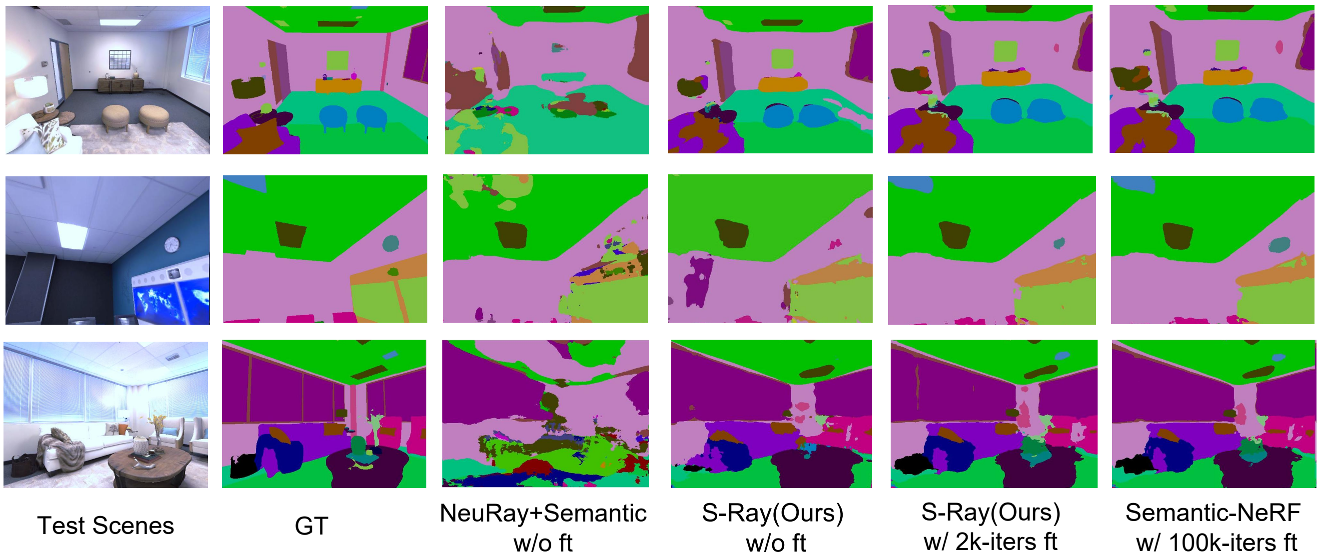

Figure 4. **Semantic rendering quality comparison**. On the left, we show direct semantic rendering results of our method and NeuRay [27]+semantic head. Limited by insufficient generalization, NeuRay+semantic head has difficulty to render semantics in unseen scenes and fails to capture contextual structure, while our method is able to learn the semantic structural prior, thus showing strong generalizability across different scenes. On the right, we show the experimental comparison between our S-Ray with $2k$ iterations finetuning ($\sim$10min) and Semantic-NeRF [61] with $100k$ iterations.

optimization, we follow [61] to train it on the new scene from scratch. To compare our results fairly, we follow the Semantic-NeRF [61] to resize the images to 640×480 for Replica [46] and 320×240 for ScanNet [7]. Results can be seen in Table 1 and in Figure 4.

Table 1 shows that our pretrained model generalizes well to unseen scenes with novel contextual information. we observe that the generalization ability of our S-Ray consistently outperforms both NeuRay [27] and MVSNeRF [4] with semantic heads. Although they are the recent methods that have strong generalization ability, we show that directly following Semantic-NeRF by adding a semantic head fails to fully capture the semantic information. Instead, our Cross-Reprojection Attention can extract relational features from multi-view reprojections of the query ray, thus achieving better accuracy and stronger generalization ability.

After finetuning for only 10 minutes, our performance is competitive and even better than Semantic-NeRF with $100k$ iters per-scene optimization. The visual results in Figure 4 clearly reflect the quantitative results of Table 1. The generalization ability of our S-Ray is obviously stronger than NeuRay [27] with semantic head. As MVSNeRF [4] shows even worse generalization performance than NeuRay with the semantic head as shown in Table 1, we do not show its visualization results due to the limited space. In general, the comparison methods directly use point-based features which benefit to per-pixel reconstruction instead of semantics. Our approach utilizes the full spatial context information around the query ray to build our semantic ray, thus leading to the best generalization ability and high segmentation across different test scenes.

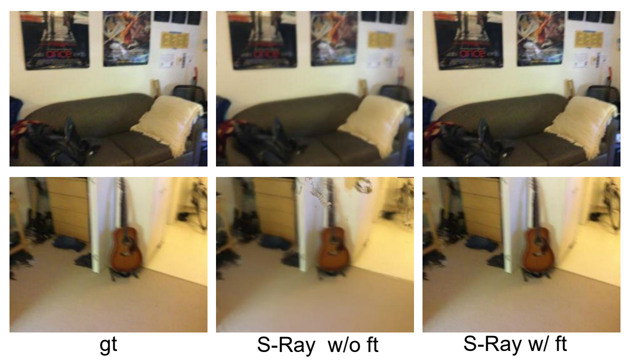

Figure 5. Qualitative results of scene rendering for generalization (w/o ft) and finetuning (w/ ft) settings in real data [7].

4.3. Ablation Studies and Analysis

Reconstruction quality of S-Ray. To evaluate the reconstruction quality of S-Ray, we follow (2) to render the radiance from geometry aware network with photometric loss same as [61]. In Table 2, we test the S-Ray for color rendering and compare it with Semantic-NeRF [61], MVSNeRF [4] and NeuRay [27]. More comparisons can be found in the supplementary. As shown in Table 2 and Figure 5, our S-Ray can also generalize well in reconstruction. This shows that our network cannot only capture the contextual semantics but also learn geometry features well.

Method	PSNR↑	SSIM↑	LIPIPS↓
Semantic-NeRF [61]	25.07	0.797	0.196
MVSNeRF [4]	23.84	0.733	0.267
NeuRay [27]	27.22	0.840	0.138
S-Ray (Ours)	26.57	0.832	0.173
S-Ray$_{ft}$ (Ours)	**29.27**	**0.865**	**0.127**

Table 2. Comparisons of scene rendering in real data [7].

Training from scratch. In order to further present the generalization ability of our S-Ray, we train our model on a scene from scratch without pretraining the model. We strictly follow the same process as Semantic-NeRF [61] to train S-Ray. Results in Table 3 (ID 9 and 10) show that training our method from scratch can also achieve similar results as finetuning the pretrained model, and it obtains even better results than Semantic-NeRF.

Evaluation of Cross-Reprojection Attention module. In Table 3, we adopt ablation studies for Cross-Reprojection module shown in Figure 3. We compare four models, the full model (ID 1), the model without Cross-Reprojection Attention (ID 2, 4), the model only with intra-view radial attention module (ID 3, 7), and the model only with cross-view attention (ID 4, 8). The results show that Cross-Reprojection Attention in S-Ray enables our network to learn more contextual information as discussed in Sec. 3.3.

ID	Description	Setting	mIoU	Total Acc
1	full S-Ray	Gen	**57.15**	**78.24**
2	w/o cross-reprojection Att	Gen	45.34	53.67
3	only intra-view Att	Gen	49.09	58.53
4	only cross-view Att	Gen	52.56	63.25
5	full S-Ray	Ft	**91.08**	**98.20**
6	w/o cross-reprojection Att	Ft	76.30	86.02
7	only intra-view Att	Ft	81.24	89.58
8	only cross-view Att	Ft	87.01	93.34
9	S-Ray	Sc	**95.31**	**98.40**
10	Semantic-NeRF [61]	Sc	94.48	95.32

Table 3. Ablation studies. mIoU and Total Acc on the real data from ScanNet [7]. "Gen" means the generalization setting, "Ft" means to finetune on the scene and "Sc" means to train from scratch.

Few-step finetuning of S-Ray. Table 4 reports mIoU and finetuning time of different models with on the ScanNet [7] dataset. We observe that by finetuning with limited time, our model is able to achieve a better performance than a well-trained Semantic-NeRF [61] with much longer training time. As Semantic-NeRF fails to present cross-scene generalization ability, it still requires full training on the unseen scene. Instead, our S-Ray is able to quickly transfer to the unseen scenes. Moreover, S-Ray outperforms other competitive baselines with similar finetuning time, which further demonstrates that our Cross-Reprojection Attention operation successfully improves the generalization ability.

Method	Train Step	Train Time	mIoU
Semantic-NeRF [61]	50k	∼2h	89.33
MVSNeRF [4] w/ s-Ft	5k	∼20min	52.02
NeuRay [27] w/ s-Ft	5k	∼32min	79.23
S-Ray-Ft **(Ours)**	5k	∼20min	**92.04**

Table 4. mIoU and training steps/time on real data [46]. "w/ s" means adding a semantic head on the baseline architectures.

From the experiments above, we have the following key observations:

1) Our Semantic Ray can exploit contextual information of scenes and presents strong generalization ability to adapt to unseen scenes. It achieves encouraging performance without finetuning on the unseen scenes, and also obtains comparable results to the well-trained Semantic-NeRF with much less time of finetuning.

2) We show the effectiveness of our Cross-Reprojection Attention module through comprehensive ablation studies. Experiments demonstrate that both intra-view and cross-view attentions are crucial for S-Ray, and we achieve the best performance by simultaneously exploiting relational information from both modules.

3) Our performance in radiance reconstruction shows the great potential of our attention strategy, which is able to learn both dense contextual connections and geometry features with low computational costs.

5. Conclusion

In this paper, we have proposed a generalizable semantic field named Semantic Ray, which is able to learn from multiple scenes and generalize to unseen scenes. Different from Semantic NeRF which relies on positional encoding thereby limited to the specific single scene, we design a Cross-Reprojection Attention module to fully exploit semantic information from multiple reprojections of the ray. In order to capture dense connections of reprojected rays in an efficient manner, we decompose the problem into consecutive intra-view radial and cross-view sparse attentions to extract informative semantics with small computational costs. Extensive experiments on both synthetic and real-world datasets demonstrate the strong generalizability of our S-Ray and effectiveness of our Cross-Reprojection Attention module. With the generalizable semantic field, we believe that S-Ray will encourage more explorations of potential NeRF-based high-level vision problems in the future.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant 62206147, and in part by Deng Feng Fund.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017. 2

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 2, 3

[3] Erzhuo Che, Jaehoon Jung, and Michael J. Olsen. Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. *Sensors*, 19(4):810, 2019. 2

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*. IEEE, 2021. 2, 6, 7, 8

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018. 2

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 6, 7, 8

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022. 3

[9] Bertrand Douillard, James Patrick Underwood, Noah Kuntz, Vsevolod Vlaskine, Alastair James Quadros, Peter Morton, and Alon Frenkel. On the segmentation of 3d LIDAR point clouds. In *ICRA*, 2011. 3

[10] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Gläser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.*, 22(3):1341–1360, 2021. 2

[11] Kyle Gao, Yina Gao, Hongjie He, Denning Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022. 2

[12] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien P. C. Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, 2021. 2, 3

[13] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017. 2, 3

[14] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(12):4338–4364, 2021. 2

[15] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging*, 32(4):582–596, 2019. 2

[16] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-Lan Tai. Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. In *ECCV (20)*, 2020. 3

[17] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019. 2

[18] Shahram Izadi, Richard A. Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Steve Hodges, Pushmeet Kohli, Jamie Shotton, Andrew J. Davison, and Andrew W. Fitzgibbon. Kinectfusion: real-time dynamic 3d surface reconstruction and interaction. In *SIGGRAPH Talks*, 2011. 3

[19] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *ICCV*, 2021. 2, 3

[20] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *CVPR*, 2022. 2, 3

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[22] Adarsh Kowdle, Christoph Rhemann, Sean Ryan Fanello, Andrea Tagliasacchi, Jonathan Taylor, Philip L. Davidson, Mingsong Dou, Kaiwen Guo, Cem Keskin, Sameh Khamis, David Kim, Danhang Tang, Vladimir Tankovich, Julien P. C. Valentin, and Shahram Izadi. The need 4 speed in real-time dense visual tracking. *ACM Trans. Graph.*, page 220, 2018. 3

[23] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *CVPR*, 2021. 3

[24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 3

[25] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 3

[26] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *ICCV*, 2021. 3

[27] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *CVPR*, pages 7814–7823. IEEE, 2022. 2, 6, 7, 8

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440. IEEE Computer Society, 2015. 2, 3

[29] Nelson L. Max. Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 1(2):99–108, 1995. 3

[30] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *ICCV*, 2021. 3

[31] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 3

[32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 3, 6

[33] Andres Milioto, Philipp Lottes, and Cyrill Stachniss. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In *ICRA*, 2018. 2

[34] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *IROS*, 2019. 2

[35] Michael Niemeyer and Andreas Geiger. GIRAFFE: representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2, 3

[36] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, pages 5589–5599, 2021. 2

[37] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 3

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[39] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 3

[40] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 3

[41] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *CVPR*, 2021. 3

[42] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, 2021. 2, 3

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 3

[44] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020. 3

[45] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 3

[46] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard A. Newcombe. The replica dataset: A digital replica of indoor spaces. *CoRR*, abs/1906.05797, 2019. 2, 6, 7, 8

[47] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *ECCV*. Springer, 2022. 3, 5, 6

[48] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 3

[49] An Tao, Yueqi Duan, Yi Wei, Jiwen Lu, and Jie Zhou. Seggroup: Seg-level supervision for 3d instance and semantic segmentation. *IEEE Trans. Image Process.*, 2022. 2

[50] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *ICCV*, 2021. 2, 3, 4

[51] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi S. M. Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes, 2021. 2, 3

[52] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *CVPR*, 2022. 3

[53] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2, 3, 4, 6

[54] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, pages 600–612, 2004. 6

[55] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020. 3

[56] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 3

[57] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 3, 4

[58] Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Generalizable model-agnostic semantic segmentation via target-specific normalization. *arXiv preprint arXiv:2003.12296*, 2020. 3

[59] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2, 3

[60] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[61] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021. 1, 2, 3, 5, 6, 7, 8

[62] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, 2021. 2