
ProvNeRF: Modeling per Point Provenance in NeRFs as a Stochastic Field

Kiyohiro Nakayama¹ Mikaela Angelina Uy^{1,2} Yang You¹ Ke Li³ Leonida J. Guibas¹
¹ Stanford University ² Nvidia ³ Simon Fraser University
w4756677@stanford.edu keli@sfu.ca
{mikacuy, yangyou, guibas}@cs.stanford.edu

Abstract

Neural radiance fields (NeRFs) have gained popularity with multiple works showing promising results across various applications. However, to the best of our knowledge, existing works do not explicitly model the distribution of training camera poses, or consequently the triangulation quality, a key factor affecting reconstruction quality dating back to classical vision literature. We close this gap with ProvNeRF, an approach that models the **provenance** for each point – i.e., the locations where it is likely visible – of NeRFs as a stochastic field. We achieve this by extending implicit maximum likelihood estimation (IMLE) to functional space with an optimizable objective. We show that modeling per-point provenance during the NeRF optimization enriches the model with information on triangulation leading to improvements in novel view synthesis and uncertainty estimation under the challenging sparse, unconstrained view setting against competitive baselines¹.

1 Introduction

Neural radiance fields (NeRFs) [42], allowing for learning 3D scenes given only 2D images, have grown in popularity in recent years. It has shown promise in many different applications such as novel view synthesis [4, 6], depth estimation [14], robotics [22, 1], localization [36, 38], etc. Existing literature [11, 15, 45] show that the quality of NeRF reconstruction is correlated with the selection of training camera poses. Similar correlations are observed in the classical literature too, triangulation is highly dependent on camera poses [47, 44, 3], which greatly influences the reconstruction quality. One common and important setting in computer vision literature [2, 37, 40, 16] is the **sparse view** [18] setting in **unconstrained** [53] environments, and triangulation is even more critical, affecting the reconstruction quality as limited input views make the system more sensitive to noise.

Despite the correlation between triangulation and reconstruction quality, to the best of our knowledge, existing works do not explicitly model the former when optimizing the latter. In this work, we address this gap in the literature by modeling for each point the *locations where it is likely visible*. We dub this as the *provenances* of a point. Modeling and learning per-point provenance can help NeRF understand how the training cameras are distributed in space, which inherently links it to triangulation and reconstruction quality.

However, determining the provenances of a point x without the underlying geometry is not straightforward as many factors influence the visibility of each point in the reconstructed geometry. For example, the literature on stereo matching [44, 47] has extensively studied the influences of camera locations on 3D reconstruction. One such well-known challenge arises when selecting the baseline of a pair of cameras in a stereo system. As shown in Fig. 2, points’ visibility can suffer from different sets of errors when the length of the camera pair’s baseline changes. For NeRFs, the dependence

¹Code will be available at <https://github.com/georgeNakayama/ProvNeRF>.

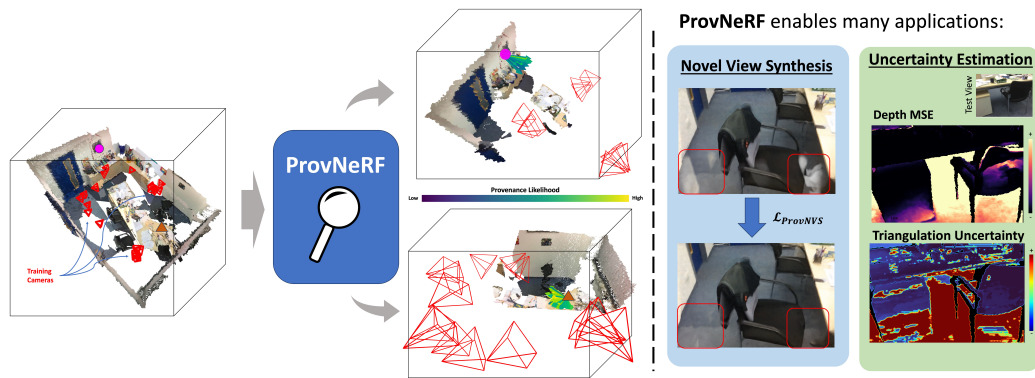


Figure 1: **(Left)** ProvNeRF models a provenance field that outputs *provenances* for each 3D point as likely samples (arrows). For 3D points (brown triangle and blue circle), the corresponding provenances (illustrated by the arrows), are locations that likely observe them. **(Right)** ProvNeRF enables better novel view synthesis and estimating the uncertainty of the capturing process because it models the locations of likely observations that is critical for NeRF’s optimization.

becomes more complex as multiple cameras’ visibility needs to be estimated. To overcome this challenge, we propose to model the provenance as the samples from a *probability distribution*, where a location y is assigned with a large likelihood if and only if x is likely to be visible from y .

To handle the potential complexity of this distribution, we represent the provenance of x as a set of location *samples*, generated from a learned probability distribution. This is distinct from the existing “attribute” prediction extensions of NeRFs [69, 30, 8] since provenance is a *distribution* for every 3D point in space. Thus, this amounts to modeling an infinite collection of distributions (per-point’s provenance) over all 3D points, which is mathematically, a *stochastic field* over $\mathbb{R}^3$. In our work, we extend implicit maximum likelihood estimation (IMLE) [33], a sample-based generative model, to model stochastic fields by adapting the objective to functional space. Furthermore, we derive an equivalent pointwise objective that can be efficiently optimized with gradient descent and use it to model the provenance field.

We dub our method **ProvNeRF** which models per-point provenance during the training stage of NeRF (Fig. 1). This enriches the model with information on triangulation quality when the model parameters are optimized. Once the provenance stochastic field is trained, we show that we can use it to improve novel view synthesis (Sec. 5.1) and estimate triangulation uncertainty in the capturing process (Sec. 5.2) under the challenging sparse, unconstrained view setting.

2 Related Works

NeRFs and their Extensions. Neural radiance fields (NeRFs) [42] have revolutionized the field of 3D reconstruction [20] and novel view synthesis [50, 32] with its powerful representation of a scene using weights of an MLP that is rendered by volume rendering [41, 58]. Follow-ups on NeRF further tackle novel view synthesis under more difficult scenarios such as unconstrained photo collections [39], unbounded [5], dynamic [35] and deformable [46] scenes, and reflective objects [59, 7]. Going beyond novel view synthesis, the NeRF representation has also shown great promise in different applications such as autonomous driving [56, 62], robotics [1, 22] and editing [67, 63]. Recent works have also extended NeRFs to model other fields in addition to color and opacity such as semantics [69, 68], normals [66], CLIP embeddings [28], image features [30] and scene flow [34]. Most of these works learn an additional function that predicts an auxiliary *deterministic* output at each point that is either a scalar or a vector, trained with extra supervision using volume rendering. All of the above works use a deterministic field to output the additional information. However, because each point’s provenance is a probabilistic distribution, we need to model a stochastic field instead of a deterministic field for provenance.

Sparse View Novel View Synthesis. NeRFs with rendering supervision alone struggle with sparse view input due to insufficient constraints in volume rendering. Several approaches have been proposed to train NeRFs under the sparse-view regime with regularization losses [43, 64], semantic

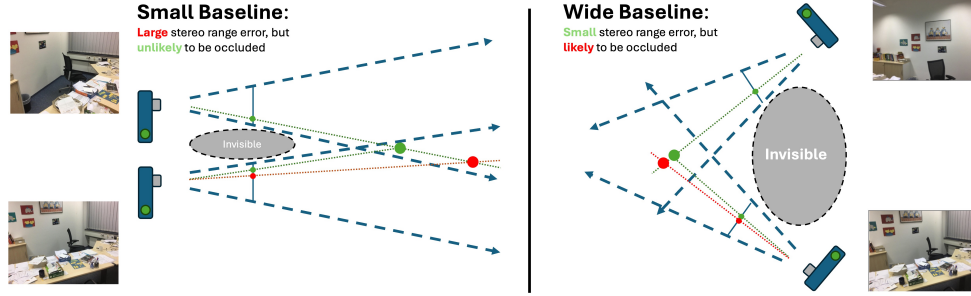


Figure 2: **Complex influence of camera baseline distance on the 3D reconstruction.** *Right:* With a wide baseline, the reconstruction is more robust against 2D measurement noises. However, it is more likely to omit hidden surfaces because the invisible region is larger than a small baseline camera pair. *Left:* With a small baseline, the 3D reconstruction is less likely to suffer from occlusions as the invisible region between cameras is small. However, the reconstruction can be noisy due to large stereo range errors (large deviation in depth with a small amount of noise in the 2D measurement).

consistency [23], and image [60] or cost volume [12, 61, 10] feature constraints. Other works also constrain the optimization using priors from data [24, 65] or depth [49, 57, 54]. Despite addressing the setting with limited number of input views, many works are not specifically designed to tackle our desired sparse, unconstrained views setting as they either focus on object-level [24, 65, 43, 23, 21], limited camera baselines [12, 61], or forward-facing scenes [60] scenes. Recent works [49, 57, 54] have looked into improving the NeRF quality on a more difficult setting of sparse, unconstrained (outward-facing) input views by incorporating depth priors. However, none of these works consider the locations and orientations of training cameras in their optimization process despite it being one of the major factors influencing the NeRF's optimization in a sparse setup.

Uncertainty Modeling in Neural Radiance Fields. The current literature separates NeRF's uncertainty into aleatoric – e.g., transient objects or changes in lighting – and epistemic – data limitation due to weak texture or limited camera views – uncertainties. Some works [39, 26] model aleatoric uncertainty by directly predicting the uncertainty values through a neural network. However, their approach requires training on large-scale data and is not suited for estimating the uncertainty of a specific scene. On the other hand, several works explore epistemic uncertainty estimation in NeRFs through variational inference [51, 52, 48], ensemble learning [55, 31], and Bayesian inference [19, 25, 45]. While these works estimate epistemic uncertainty, they still entangle different sources of uncertainty such as texture, camera poses, and model bias, resulting in unclear and inconsistent definitions of the uncertainty quantified. In our work, we specifically model the uncertainty caused by the capturing process that is useful in various downstream tasks [45, 26, 48, 31].

3 Preliminaries

3.1 Neural Radiance Fields (NeRF)

A neural radiance field (NeRF) is a coordinate-based neural network that learns a field in 3D space, where each point $\mathbf{x} \in \mathbb{R}^3$ is of certain *opacity* and *color*. Mathematically, a NeRF is parameterized by two functions representing the two fields $\mathbf{F}_{\phi, \psi} = (\sigma_{\psi}(\mathbf{x}), \mathbf{c}_{\phi}(\mathbf{x}, \mathbf{d}))$, one for opacity $\sigma_{\psi} : \mathbb{R}^3 \rightarrow \mathbb{R}_+$ and one for color $\mathbf{c}_{\phi} : \mathbb{R}^3 \times \mathbb{S}^2 \rightarrow [0, 1]^3$, where $\mathbf{d} \in \mathbb{S}^2$ is the direction from where $\mathbf{x}$ is viewed from. One of the key underpinnings of NeRFs is volume rendering allowing for end-to-end differentiable learning with only training images. Concretely, given a set of M images $I_1, I_2, \dots, I_M$ and their corresponding camera poses $P_1, P_2, \dots, P_M$, the rendered color of a pixel x is the expected color along a camera ray $\mathbf{r}_{i,x}(t) = \mathbf{o}_i + t\mathbf{d}_{i,x}$, where $\mathbf{o}_i$ is the camera origin and $\mathbf{d}_{i,x}$ is the ray direction for pixel x that can be computed from the corresponding camera pose P_i . The pixel value for 2D coordinate x is then given by the line integral:

$$\mathbf{C}_{\phi, \psi}(\mathbf{r}_{i,x}) = \int_{t_n}^{t_f} \sigma_{\psi}(\mathbf{r}_{i,x}(t)) T(\mathbf{r}_{i,x}(t)) \mathbf{c}_{\phi}(\mathbf{r}_{i,x}(t)) dt, \quad (1)$$

where t_n, t_f defines the near and far plane, and

$$T(\mathbf{r}_{i,x}(t)) = \exp \left[- \int_{t_n}^t \sigma(\mathbf{r}_{i,x}(s)) ds \right] \quad (2)$$

is the transmittance of the point $\mathbf{r}_{i,x}(t)$ along the direction $\mathbf{d}_{i,x}$.

3.2 Implicit Maximum Likelihood Estimation

One choice of probabilistic model is implicit maximum likelihood estimation (IMLE) [33] that represents a distribution as a set of samples and is designed to handle possibly multimodal distributions. As an implicit probabilistic model, IMLE learns a parameterized transformation $\mathbf{H}_\theta(\cdot)$ of a latent random variable, e.g. a Gaussian $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, where $\mathbf{H}_\theta(\cdot)$ often takes the form of a neural network that output samples $\mathbf{w}_j = \mathbf{H}_\theta(\mathbf{z}_j)$ with $\mathbf{w}_j \sim \mathbb{P}_\theta(\mathbf{w})$. Here, $\mathbb{P}_\theta$ is a probability measure obtained by transforming the standard Gaussian distribution measure via $\mathbf{H}_\theta$. Given a set of data samples $\{\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_N\}$, the IMLE objective optimizes the model parameters θ with

$$\hat{\theta} = \arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_K} \left[\sum_{i=1}^N \min_j \|\mathbf{H}_\theta(\mathbf{z}_j) - \hat{\mathbf{w}}_i\|_2^2 \right]. \quad (3)$$

It is shown that the above objective to be equivalent to maximizing the likelihood.

4 Method

In the following sections, we formally define the provenance at all points as a stochastic field (Sec. 4.1), extend IMLE to model the provenance field (Sec. 4.2), and derive an equivalent pointwise loss for gradient descent (Sec. 4.3).

Notations. We denote a stochastic field with a calligraphic font ($\mathcal{D}_\theta$) and samples from the stochastic field using the same letter but bolded ($\mathbf{D}_\theta$). Concretely $\mathbf{D}_\theta$ is a function sample, a function defined over all points $\mathbf{x} \in \mathbb{R}^3$, that is sampled from the stochastic field $\mathcal{D}_\theta$, i.e. $\mathbf{D}_\theta \sim \mathcal{D}_\theta$. $\mathbf{D}_\theta$ maps each point to one possible sample in its provenances. We also denote the distribution of the provenances at point $\mathbf{x}$ as $\mathcal{D}_\theta(\mathbf{x})$. Moreover, a provenance sample $\mathbf{D}_\theta(\mathbf{x})$ from $\mathcal{D}_\theta(\mathbf{x})$ is equivalent to evaluating the function $\mathbf{D}_\theta \sim \mathcal{D}_\theta$ at $\mathbf{x}$. Finally, we let a hat ($\hat{\cdot}$) denote the empirical samples/distributions.

4.1 Provenance as a Stochastic Field

The **provenance** of a point is defined as the *locations where it is likely visible from*, and as a point can be visible from multiple locations, it can be represented as samples from a distribution. Specifically, provenances of a point $\mathbf{x}$ can be represented as *samples* from its provenance *distribution* $\mathcal{D}_\theta(\mathbf{x})$. That means that the likelihood of sampling a location $\mathbf{y} \in \mathbb{R}^3$ from $\mathcal{D}_\theta(\mathbf{x})$ determines how likely $\mathbf{x}$ is visible from location $\mathbf{y}$. Because such distributions are defined for each 3D point $\mathbf{x} \in \mathbb{R}^3$, the collection of per-point provenances forms a stochastic field $\mathcal{D}_\theta$ indexed by coordinates $\mathbf{x} \in \mathbb{R}^3$.

Empirically, given sparse training camera views $P_1, \dots, P_M$, if $\mathbf{x}$ is inside the camera frustum Π_i for view P_i and is not occluded, an *empirical* sample from the provenances of $\mathbf{x}$ can be parameterized as a distance-direction tuple $\hat{\mathbf{D}}_i(\mathbf{x}) = (\hat{t}_{i,\mathbf{x}}, \hat{\mathbf{d}}_{i,\mathbf{x}}) \in \mathbb{R}_+ \times \mathbb{D}^3$ ². Considering all M training views, the empirical distribution of provenances at point $\mathbf{x}$ is defined by the following density function:

$$p_{emp}(t, \mathbf{d}) = \frac{1}{M} \sum_{i=1}^M \delta[(t, \mathbf{d}) = (t_{i,\mathbf{x}}, \mathbf{d}_{i,\mathbf{x}})] \quad (4)$$

$$\text{where } (t_{i,\mathbf{x}}, \mathbf{d}_{i,\mathbf{x}}) = \left(v_{i,\mathbf{x}} \|\mathbf{x} - \mathbf{o}_i\|, v_{i,\mathbf{x}} \frac{\mathbf{x} - \mathbf{o}_i}{\|\mathbf{x} - \mathbf{o}_i\|} \right) \quad (5)$$

where δ is the Dirac delta function; $v_{i,\mathbf{x}} \in [0, 1]$ determines the length of $\mathbf{d}$ to handle occlusions. We modeled it as the transmittance from the NeRF model. To recover the location that observes $\mathbf{x}$, we can write $\mathbf{y}_{i,\mathbf{x}} = \mathbf{x} - t_{i,\mathbf{x}} \mathbf{d}_{i,\mathbf{x}}$.

² $\mathbb{D}^3$ denotes a solid ball in $\mathbb{R}^3$

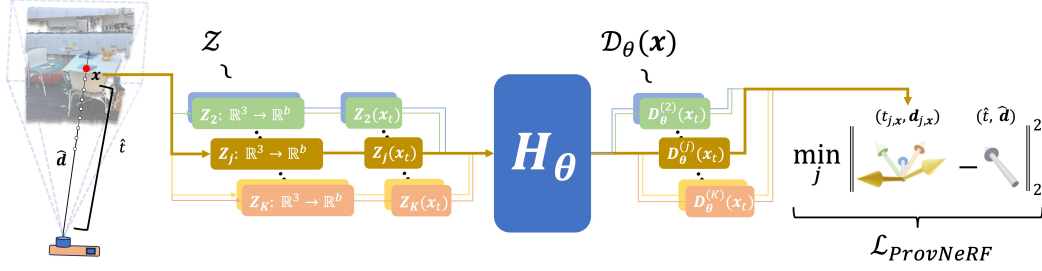


Figure 3: **Training pipeline for ProvNeRF.** For each point x seen from provenance tuple $(\hat{t}, \hat{d})$, with direction $\hat{d}$ at distance $\hat{t}$, we first sample K latent random functions $\{Z_j\}$ from distribution $\mathcal{Z}$. The learned transformation H_θ transforms each $Z_j(x)$ to a provenance sample $D_\theta^{(j)}(x)$. Finally H_θ is trained with $\mathcal{L}_{\text{ProvNeRF}}$ as defined in Eq. 9.

While the above empirical distribution of provenances is given by the training cameras, the actual distribution of provenances, i.e. for each point the locations that point is likely visible from, can have a more complex dependence on both the underlying geometry and the cameras. To capture this complexity, we model $\mathcal{D}_\theta(x)$ as a learnable network, a probabilistic model that can model potentially complex distribution, and one choice of such a model is implicit maximum likelihood estimate (IMLE) [33]. Similar to the empirical distribution, we also represent provenance samples from $\mathcal{D}_\theta(x)$ as a distance-direction tuple (t, d) as defined in Eq 5. We optimized our network with the empirical distribution $\hat{\mathcal{D}}$ as training signals.

$\mathcal{D}_\theta(x)$ defines a distribution for all point $x \in \mathbb{R}^3$. Treating $\mathbb{R}^3$ as the index set, $\mathcal{D}_\theta = \{\mathcal{D}_\theta(x)\}_{x \in \mathbb{R}^3}$ defines a stochastic field on $\mathbb{R}^3$ as a collection of distributions $\mathcal{D}_\theta(x)$ for all $x \in \mathbb{R}^3$. Because a stochastic field is composed of infinitely many random variables over $\mathbb{R}^3$, existing methods cannot be applied out of the box as they only model finite-dimensional distributions. In the following sections, we extend IMLE [33] to model this stochastic field.

4.2 ProvNeRF

ProvNeRF models provenances of a NeRF as a stochastic field by extending IMLE [33] to functional space. IMLE learns a mapping that transforms a latent distribution to the data distribution, where each data sample is either a scalar or a vector (Sec. 3.2). However, in our context, since samples from the stochastic field $\mathcal{D}_\theta$ are *functions* mapping 3D locations to provenances, we need to extend IMLE to learn a neural network mapping H_θ that transforms a pre-defined *latent stochastic field* $\mathcal{Z}$ to the provenance distribution $\mathcal{D}_\theta$ (See Fig. 3).

Let $\mathcal{Z}$ be the stochastic field where each sample $Z \sim \mathcal{Z}$ is a function $Z : \mathbb{R}^3 \rightarrow \mathbb{R}^b$. To transform $\mathcal{Z}$ to $\mathcal{D}_\theta$, fIMLE learns a deterministic mapping H_θ that maps each latent function $Z \sim \mathcal{Z}$ to a function $D_\theta \sim \mathcal{D}_\theta$ via composition: $D_\theta = H_\theta \circ Z$. H_θ here is represented as a neural network to handle complex transformations from $\mathcal{Z}$ to $\mathcal{D}_\theta$.

We define a latent function sample $Z \sim \mathcal{Z}$ to be the concatenation of a *random linear transformation* of x and x itself. Mathematically, each latent function $Z \sim \mathcal{Z}$ is a block matrix of size $(b + 4) \times 3$:

$$Z(x) = \begin{bmatrix} z \\ I \end{bmatrix} x, \text{ where } z \sim \mathcal{N}(0, \lambda^2 I), x \in \mathbb{R}^3. \quad (6)$$

Although Z can be designed to have non-linear dependence on the input location x , we experimentally show that this simple design choice works well across different downstream applications.

To train H_θ , we maximize the likelihood of the training provenances (Eq. 4) under $\mathcal{D}_\theta$ for each x using the IMLE objective [33] extended to functional space. We term this extension as functional Implicit Maximum Likelihood Estimation (fIMLE). Because a direct extension to fIMLE leads to an intractable objective, we derive an efficient pointwise loss between the training provenances and model predictions equivalent to the fIMLE objective in the following section.



Figure 4: **Visual Effect of $\mathcal{L}_{\text{ProvNVS}}$ in Eq. 10.** Compared to pre-trained SCADE model, our method can remove additional floaters in the scene (see the boxed region).

	Scannet			Tanks and Temple		
	PSNR ($\uparrow$)	SSIM ($\uparrow$)	LPIPS ($\downarrow$)	PSNR ($\uparrow$)	SSIM ($\uparrow$)	LPIPS ($\downarrow$)
NeRF [42]	19.03	0.670	0.398	17.19	0.559	0.457
DDP [49]	19.29	0.695	0.368	19.18	0.651	0.361
SCADE [57]	<u>21.54</u>	0.732	<u>0.292</u>	<u>20.13</u>	<u>0.662</u>	<u>0.358</u>
DäRF [54]	21.28	0.741	0.323	19.67	0.652	0.374
Ours	21.73	<u>0.733</u>	0.291	20.36	0.663	0.349

Table 1: **Novel View Synthesis Results.** Our method outperforms baselines in novel view synthesis on both Scannet and Tanks and Temple Datasets. This is because our novel NeRF regularizer in Eq. 10 can remove additional floaters in the scene as shown in Fig. 4. See Sec. 5.1 for details.

4.3 Functional Implicit Maximum Likelihood Estimation

We construct an IMLE objective for stochastic fields to maximize the likelihood of training provenances under $\mathcal{D}_\theta$. Similar to Eq. 3, if we have i.i.d. empirical samples $\hat{D}_1, \dots, \hat{D}_M$ from the empirical stochastic field $\hat{D}$ (defined in Sec. 4.1), and model samples $D_\theta^{(1)}, \dots, D_\theta^{(K)}$ from the parameter stochastic field $\mathcal{D}_\theta$, we define the fIMLE objective as

$$\hat{\theta} = \arg \min_{\theta} \mathbb{E}_{D_\theta^{(1)}, \dots, D_\theta^{(K)}} \left[\sum_{i=1}^n \min_j \left\| \hat{D}_i - D_\theta^{(j)} \right\|_{L^2}^2 \right]. \quad (7)$$

Unlike the original IMLE objective (Eq. 3) that can be directly optimized, the fIMLE objective in Eq. 7 requires the computation of a L^2 integral norm – a functional analogy to the L^2 vector norm – which, in general, is not analytically in closed form. Furthermore, approximations of this integral are very expensive since each point query to D_θ needs a forward pass through H_θ .

To get around this, we use the calculus of variations to reformulate Eq. 7 to minimize the pointwise difference between the empirical samples and model predictions³. This allows us to write the fIMLE objective as

$$\mathcal{L}_{\text{fIMLE}} = \mathbb{E}_{D_\theta^{(1)}, \dots, D_\theta^{(K)} \sim \mathcal{D}_\theta} \left[\sum_{i=1}^n \min_j \mathbb{E}_{\mathbf{x} \sim \mathcal{U}(\Omega)} \left\| \hat{D}_i(\mathbf{x}) - D_\theta^{(j)}(\mathbf{x}) \right\|_2^2 \right], \quad (8)$$

where $\mathcal{U}(\Omega)$ is a uniform distribution over the scene bound Ω . Eq. 8 only requires computing the pointwise difference between samples from $\hat{D}(\mathbf{x})$ and $D_\theta(\mathbf{x})$, making it efficiently optimizable with

³See the supplementary for the full derivation

gradient descent. Ultimately, **ProvNeRF** jointly updates the underlying NeRF’s parameters and $\mathcal{D}_\theta$ by minimizing

$$\mathcal{L}_{\text{ProvNeRF}} = \mathcal{L}_{\text{NeRF}} + \mathcal{L}_{\text{fIMLE}} = \mathcal{L}_{\text{NeRF}} + \mathbb{E}_{\mathcal{D}_\theta^{(1)}, \dots, \mathcal{D}_\theta^{(K)}, \mathbf{x}} \left[\min_j \mathbb{E}_{(\hat{t}, \hat{\mathbf{d}})} \left\| (\hat{t}, \hat{\mathbf{d}}) - (t_{j,\mathbf{x}}, \mathbf{d}_{j,\mathbf{x}}) \right\|_2^2 \right] \quad (9)$$

where $(t_{j,\mathbf{x}}, \mathbf{d}_{j,\mathbf{x}}) = \mathcal{D}_\theta^{(j)}(\mathbf{x})$, $(\hat{t}, \hat{\mathbf{d}})$ are i.i.d. samples from $\hat{\mathcal{D}}(\mathbf{x})$, and $\mathcal{L}_{\text{NeRF}}$ is the original objective of the NeRF model, e.g. photometric loss and depth loss. We provide implementation and architectural details in the supplementary material. See Figure 3 for the training pipeline illustration.

5 Experiments

Our ProvNeRF learns per-point provenance field $\mathcal{D}_\theta$ by optimizing $\mathcal{L}_{\text{ProvNeRF}}$ on a NeRF-based model. To validate ProvNeRF, we demonstrate that jointly optimizing the provenance distribution and NeRF representation can result in better scene reconstruction as shown in the task of novel view synthesis (Sec. 5.1). Moreover, we also show that the learned provenance distribution enables other downstream tasks such as estimating the uncertainty of the capturing field (Sec. 5.2). We provide an ablation study on fIMLE against other probabilistic methods in Sec. 5.3. Lastly, in Sec. 5.4, we show a preliminary extension of ProvNeRF to 3DGS [27].

Stochastic Provenance Field Visualization Fig. 7 visualizes the provenance stochastic field by sampling 16 provenances on a test view of the Scannet 758 scene. The directions of the samples are the *negative* of the predicted provenance directions for better illustration. Each sample is colored based on its predicted visibility. Notice that fIMLE allows ProvNeRF to predict multimodal provenance distributions at different scene locations.

5.1 Novel View Synthesis

We show modeling per-point provenance improves sparse, unconstrained novel view synthesis. As a point’s provenances are sample locations from where the point is likely visible, the region between the provenance location samples and the query point should likely be empty. We design our provenance loss for NVS with this intuition.

Concretely, starting from a given NeRF model, we first sample points $\mathbf{x}_1, \dots, \mathbf{x}_N$ for a training camera ray parameterized as $\hat{\mathbf{r}}_x(t)$. Here we denote point $\mathbf{x}_i = \hat{\mathbf{r}}_x(\hat{t}_i)$. We only take points $\mathbf{x}_i$ with transmittance greater than a selected threshold $\lambda = 0.9$. For each visible point $\mathbf{x}_i$, we sample provenances $(t_1^{(i)}, \mathbf{d}_1^{(i)}), \dots, (t_K^{(i)}, \mathbf{d}_K^{(i)})$ from $\mathcal{D}_\theta(\mathbf{x}_i)$ with $\|\mathbf{d}_1^{(i)}\|_2 \geq 0.7$. Then each distance-direction tuple $(t_j^{(i)}, \mathbf{d}_j^{(i)})$ gives a location $\mathbf{y}_j^{(i)} = \mathbf{x}_i - t_j^{(i)} \mathbf{d}_j^{(i)}$ from which $\mathbf{x}_i$ is observed. This in turn means $\mathbf{x}$ should be equally visible when rendered from ray parameterized as $\mathbf{r}_x^{(i)}(t) = \mathbf{y}_j^{(i)} + t \mathbf{d}_j^{(i)}, \forall j$. With this, we define our provenance loss for novel view synthesis as

$$\mathcal{L}_{\text{ProvNVS}} = \sum_{i=1}^N \sum_{j=1}^K \left[\alpha + T(\mathbf{r}_x^{(i)}(t_j^{(i)})) - T(\hat{\mathbf{r}}_x(\hat{t}_i)) \right]_+, \quad (10)$$

where $[\dots]_+$ denotes the hinge loss and $\alpha = 0.05$ is a constant margin. $\mathcal{L}_{\text{ProvNVS}}$ encourages the transmittance at $\mathbf{x}_i$ along training camera rays to be *at least* the visibility predicted by the sampled provenances from the provenance field with margin α . By matching transmittances between the provenance directions and the training rays, $\mathcal{L}_{\text{ProvNVS}}$ can be used together with $\mathcal{L}_{\text{ProvNeRF}}$ to optimize the NeRF representation and the provenance field, resulting in an improved scene geometry. We apply ProvNeRF to SCADe [57] for the task of novel view synthesis. See the supplement for details on the dataset, metrics, baselines, and implementation details.

Results. Table 1 shows our approach outperforms the state-of-the-art baselines in NVS on scenes from both the Scannet [13] and Tanks and Temples [29] dataset. Qualitative comparisons are shown

	PSNR ($\uparrow$)	SSIM ($\uparrow$)	LPIPS ($\downarrow$)
Deterministic Field	21.38	0.720	0.307
Frustum Check	21.56	0.728	0.297
Ours	21.73	0.733	0.291

Table 2: NVS Ablation Results on Scannet.

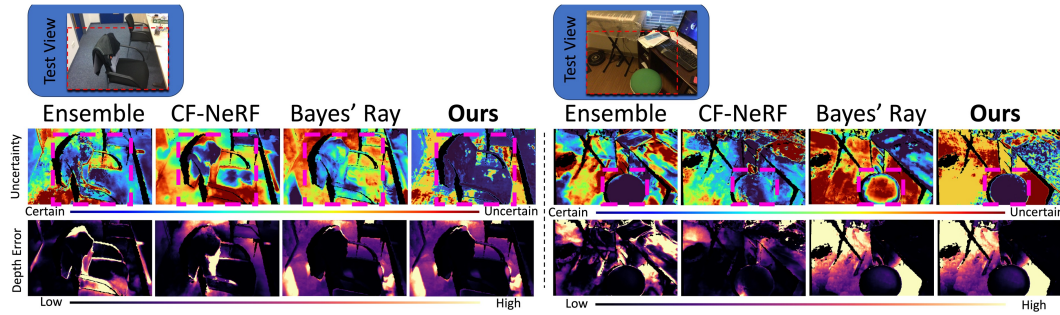


Figure 5: **Qualitative Results for Uncertainty Modeling.** We visualize our uncertainty maps obtained using the method described in Sec. 5.2. The uncertainty and depth error maps are shown with color bars specified. Uncertainty values and depth errors are normalized per test image for the result to be comparable. As shown in the boxed regions, our method predicts uncertainty regions with more correlation with the predicted depth errors.

	Scannet				Matterport			
	Avg.	#710	#758	#781	Avg.	Room 0	Room 1	Room 2
Ensemble	7.71	<u>3.01</u>	<u>2.96</u>	17.2	63.0	8.04	110	71.3
CF-NeRF [52]	660	430	571	980	507	799	488	233
Bayes' Rays [19]	<u>5.47</u>	5.11	5.23	<u>6.07</u>	<u>5.49</u>	<u>5.67</u>	<u>5.77</u>	<u>5.91</u>
Ours	-3.05	0.19	-1.93	-7.40	-11.0	-13.6	-10.2	-9.17

Table 3: **NLL Results for Triangulation Uncertainty.**

in Fig. 4. We see that compared to the baseline SCADE, whose geometry is already relatively crisp, our $\mathcal{L}_{\text{ProvNVS}}$ can further improve its NVS quality by removing additional cloud artifacts, as shown in the encircled regions. Note that this improvement does not require any additional priors and is only based on the provenance of the scene.

We also compare our performance with deterministic baselines: *Deterministic Field* regresses one provenance for each 3D location using a neural network and *Frustum Check* calculates the training provenance defined in Eq. 4 by back-projecting the sampled points to one of the training camera and use that as the regularization information. Table 2 shows that our provenance field outperforms these baselines on the novel view synthesis task because the deterministic field cannot model complex provenance distribution and the frustum check baseline lacks generalization ability as it cannot be optimized to adapt the output provenance based on the current NeRF's geometry.

5.2 Modeling Uncertainty in the Capturing Process

Provenances allow for estimating the uncertainty in triangulation, i.e., the capturing process. In classical multiview geometry [20], the angle between the rays is a good rule of thumb that determines the accuracy of reconstruction. Fig. 6 illustrates this rule as the region of uncertainty changes depending on the setup of the cameras. Formally, for a 3D point $\mathbf{x}$, we sample provenances $\{(t_j, \mathbf{d}_j)\}_{j=1}^K$ from $\mathcal{D}_\theta(\mathbf{x})$. Treating $\mathbf{d}_j$ as the principal axes and t_j as the distances from $\mathbf{x}$ to the camera origin, each provenance sample defines a pseudo camera P_j that observes $\mathbf{x}$ at pixel location $x_j = \text{Proj}_j(\mathbf{x})$. Following chapter 12.6 of [20], we define the triangulation uncertainty of $\mathbf{x}$ as the probability of $\mathbf{x}$ given its noisy 2D pseudo observations: $\mathbb{P}(\mathbf{x} | x_1, \dots, x_K) \propto \mathbb{P}(x_1, \dots, x_K | \mathbf{x}) \mathbb{P}(\mathbf{x}) = \prod_{j=1}^K \mathbb{P}(x_j | \mathbf{x}) \mathbb{P}(\mathbf{x})$.

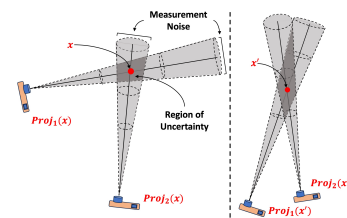


Figure 6: **Triangulation Uncertainty [20].** The figure shows that $\mathbf{x}'$ is more uncertain compared to $\mathbf{x}$ because the predicted provenances for $\mathbf{x}'$ give a narrower baseline than the baseline given by provenances of $\mathbf{x}$.

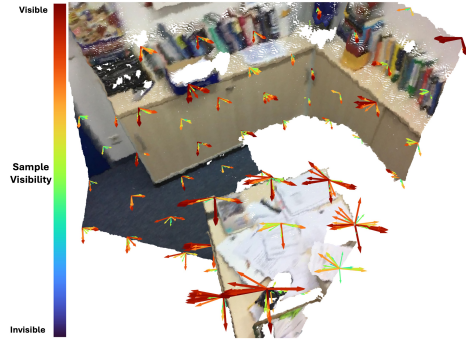


Figure 7: Visualization of Provenance Field.

	AP ($\uparrow$)	AUC ($\uparrow$)
Deterministic Field	0.163	0.168
Gaussian-based w/ $C = 2$	0.537	0.539
Gaussian-based w/ $C = 5$	0.629	0.631
VAE-based	0.323	0.325
ProvNeRF w/ Spatial Inv. $\mathcal{Z}$	0.742	0.744
Ours	0.745	0.747

Table 4: Ablation Results on Scannet.

The last two equalities are derived by assuming independence of the 2D observations and each $\mathbb{P}(x_j|\mathbf{x})$ follows a Gaussian distribution $\mathcal{N}(\text{Proj}_j(\mathbf{x}), \sigma^2)$. This assumption is equivalent to corrupting each 2D observation $\text{Proj}_j(\mathbf{x})$ by a zero-mean Gaussian noise with σ^2 variance, accounting for measurement noises in the capturing process. Assuming a uniform prior of $\mathbb{P}(\mathbf{x})$ over the scene bound, the exact likelihood can be efficiently computed with importance sampling. This quantifies a point’s triangulation quality given the sampled provenances, which becomes a measurement of the uncertainty of the capturing process. We apply our provenance field to ProvNeRF with different NeRF backbones [57, 49] and compute the likelihood. See supplementary for details on the dataset, metrics, baselines, and implementations.

Results. Tab. 3 shows the quantitative results on Scannet [13] and Matterport3D [9]. We follow [51, 55] to measure the negative log-likelihood (NLL) of the ground-truth surface under each model’s uncertainty prediction. Since our **ProvNeRF** can be applied to any pre-trained NeRF module, we use pre-trained SCADE [57] for Scannet and DDP [49] for Matterport3D, both of which are state-of-the-art approaches in each dataset. Our approach achieves the best NLL across all scenes in both datasets by a margin because we compute a more fundamentally grounded uncertainty from classical multiview geometry [20] based on triangulation errors, while both CF-NeRF and Bayes’ Rays require an approximation of the true posterior likelihood. Fig. 5 shows qualitative comparisons between baselines’ and our method’s uncertainty estimation. We expect a general correlation between uncertain regions with high-depth errors. An ideal uncertainty map should mark high-depth error regions with high uncertainty and vice versa. As shown in the boxed regions in the figure, our method’s uncertainty map shows better correlation with the depth error maps. We also quantitatively evaluate uncertainty maps using negative log-likelihood following prior works. Notice that in both examples, our method’s certain (blue) regions mostly have low-depth errors (e.g., encircled parts in Fig. 5) because our formulation only assigns a region to be certain if it is well triangulated (Fig. 6). On the other hand, baselines struggle in these regions because they either use an empirical approximation from data or a Gaussian approximation of the ground truth posterior likelihood.

5.3 Ablation Study

We validate the choice of fIMLE as our probabilistic model by measuring the average precision (AP) [17] and area under the curve (AUC) [17] of predicted provenances $(t, \mathbf{d})$ against ground truth provenances $(\hat{t}, \hat{\mathbf{d}})$ for a set of densely sampled points in the scene bound. See supplementary for metric and ablation implementation details.

Deterministic v.s. Stochastic Field. We validate the importance of modeling per-point provenance as a stochastic field rather than a deterministic field. We model $\mathcal{D}_\theta$ with a deterministic field parameterized by a neural network. Table 4 shows the importance of modeling per-point provenance as a stochastic field. Since the provenances of a point are inherently multimodal, a deterministic field that only maps each $\mathbf{x}$ to a single provenance cannot capture this multimodality.

Choice of Probabilistic Model. We validate our choice of fIMLE [33] as our probabilistic model. We first compare with explicit probabilistic models that model the provenance field as a mixture of C Gaussian processes and a VAE-based model. Table 4 shows results for the Gaussian Mixture field with $C = 2, 5$ and the VAE-based process. Although the performances for the Gaussian-based models improve as we increase C , they still suffer from expressivity because of their explicit density

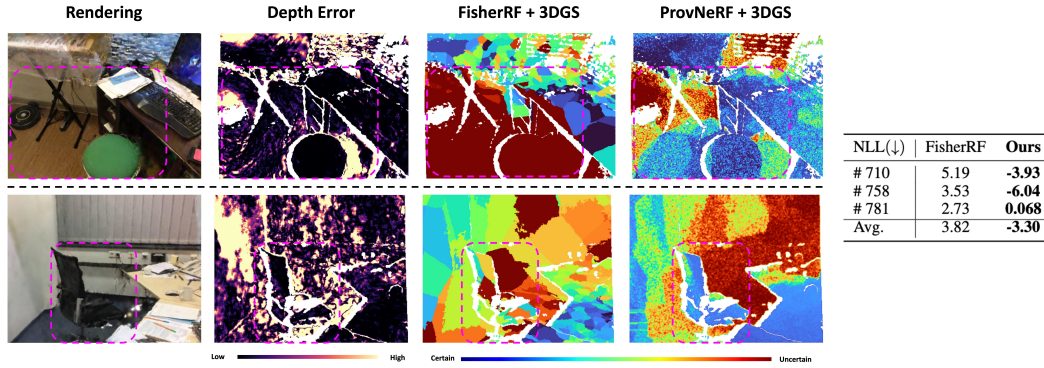


Figure 8: **Uncertainty Estimation Comparison with 3DGS.** Compared with FishRF, our method is able to estimate uncertainties that correlate more with the depth error as shown by the encircled regions. The right shows a quantitative comparison of uncertainty in negative log-likelihood.

assumption. Similarly, the VAE-based model suffers from mode-collapse while our fIMLE enables capturing a more complex distribution with a learned transformation H_θ .

Choice of Random Function $\mathcal{Z}$. Lastly, we validate our latent stochastic field $\mathcal{Z}$. We ablate our choice of $\mathcal{Z}$ with instead using a spatially *invariant* latent stochastic field $\mathcal{Z}^*$ with $\mathcal{Z}^*(x) = [\varepsilon, x] \forall x$. Here, ε is a Gaussian noise vector in $\mathbb{R}^d$. Table 4 shows the comparison between $\mathcal{D}_\theta$ obtained by transforming $\mathcal{Z}$ (**Ours**) and transforming $\mathcal{Z}^*$ (**Spatial Inv. $\mathcal{Z}$**). We see that using a spatially varying latent stochastic field further increases the expressivity of our model.

5.4 Preliminary Extension to 3D Gaussian Splatting

Because ProvNeRF is a post-hoc method that can model the provenance information for arbitrary novel view synthesis representations, we conduct a preliminary experiment that extends our provenance field modeling to 3D Gaussian Splatting [27]. Specifically, given a pre-trained Gaussian representation $\mathcal{G}$, we model a provenance distribution for each splat using IMLE with a shared 6-layer MLP for H_θ . The post-training takes around 30 minutes on a single A6000 Nvidia GPU. To show the usefulness of ProvNeRF applied to 3DGS, we use the methodology in Sec. 5.2 to estimate uncertainty maps and compare them with the predicted depth errors. Fig. 8 shows a qualitative comparison of our uncertainty map w.r.t. FishRF [25], a recent 3DGS uncertainty estimation baseline. Compared to their uncertainty map, ours shows more correlation to the depth error as highlighted by the boxed regions. Quantitatively we evaluate NLL on the three Scannet scenes shown on the right side of the same figure and show substantial improvements over FishRF. This improvement over existing literature suggests applying ProvNeRF to other representations such as 3DGS is promising. We leave further exploration of the method and applications as future works.

6 Conclusion, Limitation, & Future Works

We present ProvNeRF, a model that enhances the traditional NeRF representation by modeling provenance through an extension of IMLE for stochastic processes. ProvNeRF can be easily applied to any NeRF model to enrich its representation. We showcase the advantages of modeling per-point provenance in various downstream applications such as improving novel view synthesis and modeling the uncertainty of the capturing process.

We note that our work is not without limitations. Our ProvNeRF requires post-hoc optimization, which takes around 8 hours on SCADE, limiting its current usability for real-time or on-demand applications. However, the idea presented in our work is not specific to the model design and can be adapted to other representations. See Sec. 5.4 for preliminary adaption of ProvNeRF to 3DGS.

We also note that the hyperparameters to incorporate ProvNeRF are chosen for better performance, e.g. for the uncertainty and novel view synthesis applications, and in the future, it will be beneficial to explore a more adaptive approach in integrating provenance to different downstream applications.

Acknowledgement This work is supported by a Vannevar Bush Faculty Fellowship, ARL grant W911NF-21-2-0104, an Apple Scholars in AI/ML PhD Fellowship, and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

- [1] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.
- [2] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, 2009.
- [3] Kalin Atanassov, Vikas Ramachandra, Sergio R. Goma, and Milivoje Aleksic. 3d image processing architecture for camera phones. In J. Angelo Beraldin, Geraldine S. Cheok, Michael B. McCarthy, Ulrich Neuschaefer-Rube, Ian E. McDowall, Margaret Dolinsky, and Atila M. Baskurt, editors, *Three-Dimensional Imaging, Interaction, and Measurement*, volume 7864, page 786414. International Society for Optics and Photonics, SPIE, 2011.
- [4] 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.
- [5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.
- [6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023.
- [7] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020.
- [8] Jiazong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. In *NeurIPS*, 2023.
- [9] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *3DV*, 2017.
- [10] Di Chang, Aljaž Božič, Tong Zhang, Qingsong Yan, Yingcong Chen, Sabine Süsstrunk, and Matthias Nießner. Rc-mvsnet: Unsupervised multi-view stereo with neural rendering. In *ECCV*, 2022.
- [11] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *arXiv*, 2023.
- [12] 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 *CVPR*, 2021.
- [13] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.
- [14] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022.
- [15] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
- [16] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018.
- [17] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [18] Yasutaka Furukawa and Carlos Hernández. *Multi-View Stereo: A Tutorial*. Now Foundations and Trends, 2015.
- [19] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes’ rays: Uncertainty quantification in neural radiance fields. *arXiv*, 2023.
- [20] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

- [21] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. *arXiv preprint arXiv:2312.14937*, 2023.
- [22] Jeffrey Ichnowski*, Yahav Avigal*, Justin Kerr, and Ken Goldberg. Dex-nerf: Using a neural radiance field to grasp transparent objects. In *Conference on Robot Learning (CoRL)*, 2020.
- [23] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894, October 2021.
- [24] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *ICCV*, 2021.
- [25] Wen Jiang, Boshu Lei, and Kostas Daniilidis. Fisherrf: Active view selection and uncertainty quantification for radiance fields using fisher information, 2023.
- [26] Liren Jin, Xieyuanli Chen, Julius Rückin, and Marija Popović. Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering. In *IROS*, 2023.
- [27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.
- [28] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *ICCV*, 2023.
- [29] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.
- [30] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *NeurIPS*, 2022.
- [31] Georgios Kopanas and George Drettakis. Improving nerf quality by progressive camera placement for unrestricted navigation in complex environments. In *Vision, Modeling, and Visualization*, 2023.
- [32] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In *ICCV*, 1999.
- [33] Ke Li and Jitendra Malik. Implicit maximum likelihood estimation, 2018.
- [34] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.
- [35] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *CVPR*, 2023.
- [36] Jianlin Liu, Qiang Nie, Yong Liu, and Chengjie Wang. Nerf-loc: Visual localization with conditional neural radiance field. In *ICRA*, 2023.
- [37] Sheng Liu, Xiaohan Nie, and Raffay Hamid. Depth-guided sparse structure-from-motion for movies and tv shows. In *CVPR*, 2022.
- [38] Dominic Maggio, Marcus Abate, Jingnan Shi, Courtney Mario, and Luca Carlone. Loc-nerf: Monte carlo localization using neural radiance fields, 2022.
- [39] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021.
- [40] Hidenobu Matsuki, Raluca Scona, Jan Czarnowski, and Andrew J. Davison. Codemapping: Real-time dense mapping for sparse slam using compact scene representations. *IEEE Robotics and Automation Letters*, 2021.
- [41] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.
- [42] 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.
- [43] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022.

- [44] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993.
- [45] Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. In *ECCV*, 2022.
- [46] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021.
- [47] Satyarth Praveen. Efficient depth estimation using sparse stereo-vision with other perception techniques. In Sudhakar Radhakrishnan and Muhammad Sarfraz, editors, *Coding Theory*, chapter 7. IntechOpen, Rijeka, 2019.
- [48] Yunlong Ran, Jing Zeng, Shibo He, Jiming Chen, Lincheng Li, Yingke Chen, Gim Hee Lee, and Qi Ye. Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters*, 2022.
- [49] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, 2022.
- [50] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *CVPR*, 1997.
- [51] J. Shen, A. Ruiz, A. Agudo, and F. Moreno-Noguer. Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. In *3DV*, 2021.
- [52] Jianxiong Shen, Antonio Agudo, Francesc Moreno-Noguer, and Adria Ruiz. Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In *ECCV*, 2022.
- [53] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Trans. Graph.*, 2006.
- [54] Jiuhun Song, Seonghoon Park, Honggyu An, Seokju Cho, Min-Seop Kwak, Sungjin Cho, and Seungryong Kim. Därf: Boosting radiance fields from sparse inputs with monocular depth adaptation. In *NeurIPS*, 2023.
- [55] Niko Sünderhauf, Jad Abou-Chakra, and Dimity Miller. Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields. In *ICRA*, 2022.
- [56] Matthew Tancik, Vincent Casser, Xincheng Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, 2022.
- [57] Mikaela Angelina Uy, Ricardo Martin-Brualla, Leonidas Guibas, and Ke Li. Scade: Nerfs from space carving with ambiguity-aware depth estimates. In *CVPR*, 2023.
- [58] Mikaela Angelina Uy, George Kiyohiro Nakayama, Guandao Yang, Rahul Krishna Thomas, Leonidas Guibas, and Ke Li. Nerf revisited: Fixing quadrature instability in volume rendering. In *NeurIPS*, 2023.
- [59] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022.
- [60] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021.
- [61] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021.
- [62] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *ECCV*, 2022.
- [63] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. In *ECCV*, 2022.
- [64] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [65] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

- [66] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, 2022.
- [67] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: Geometry editing of neural radiance fields. In *CVPR*, 2022.
- [68] Xiaoshuai Zhang, Abhijit Kundu, Thomas Funkhouser, Leonidas Guibas, Hao Su, and Kyle Genova. Nerflets: Local radiance fields for efficient structure-aware 3d scene representation from 2d supervision. In *CVPR*, 2023.
- [69] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitations of our method in the supplementary

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: we provide the full derivation of our FIMLE objective in the supplementary.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the training details, hyperparameters, network architectures in the supplementary.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will provide reproducible code upon the paper's acceptance.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Guidelines:

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We address the potential societal impact on climate change in the supplementary materials

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]