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

Kiyohiro Nakayama[1]   **Mikaela Angelina Uy**[1,2]   **Yang You**[1]   **Ke Li**[3]   **Leonida J. Guibas**[1]
[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 baselines[1].

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

---

[1]Code will be available at `https://github.com/georgeNakayama/ProvNeRF`.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).
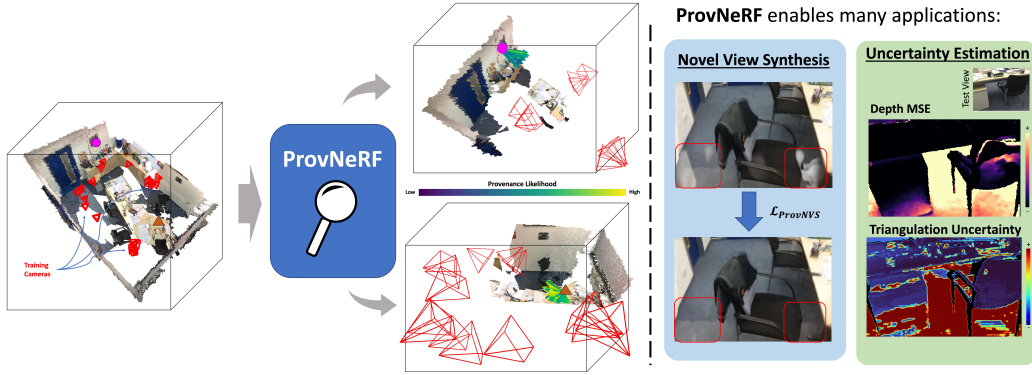
Figure 1: (**Left**) **ProvNeRF** models a provenance field that outputs *provenances* for each 3D point as likely samples (arrows). For 3D points (brown triangle and blue circle), the corresponding provenances (illustrated by the arrows), are locations that likely observe them. (**Right**) **ProvNeRF** 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 [72, 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, 59]. 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 [60, 7]. Going beyond novel view synthesis, the NeRF representation has also shown great promise in different applications such as autonomous driving [56, 64], robotics [1, 22] and editing [69, 65]. Recent works have also extended NeRFs to model other fields in addition to color and opacity such as semantics [72, 71], normals [68], 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, 66], semantic

2
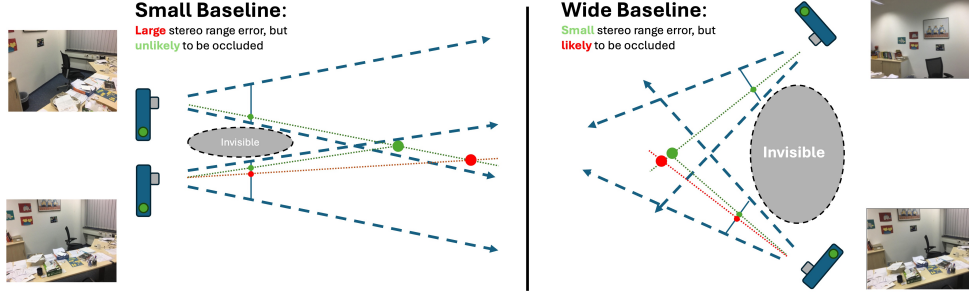
Figure 2: **Complex influence of camera baseline distance on the 3D reconstruction.** *Right:* With a wide baseline, the reconstruction is more robust against 2D measurement noises. However, it is more likely to omit hidden surfaces because the invisible region is larger than a small baseline camera 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 [61] or cost volume [12, 63, 10] feature constraints. Other works also constrain the optimization using priors from data [24, 67] or depth [49, 58, 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, 67, 43, 23, 21], limited camera baselines [12, 63], or forward-facing scenes [61] scenes. Recent works [49, 58, 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 $x \in \mathbb{R}^3$ is of certain *opacity* and *color*. Mathematically, a NeRF is parameterized by two functions representing the two fields $\boldsymbol{F}_{\phi,\psi} = (\boldsymbol{\sigma}_\psi(\boldsymbol{x}), \boldsymbol{c}_\phi(\boldsymbol{x}, \boldsymbol{d}))$, one for opacity $\boldsymbol{\sigma}_\psi : \mathbb{R}^3 \to \mathbb{R}_+$ and one for color $\boldsymbol{c}_\phi : \mathbb{R}^3 \times \mathbb{S}^2 \to [0, 1]^3$, where $\boldsymbol{d} \in \mathbb{S}^2$ is the direction from where $\boldsymbol{x}$ 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 $I_1, I_2, ..., I_M$ and their corresponding camera poses $P_1, P_2, ..., P_M$, the rendered color of a pixel x is the expected color along a camera ray $\boldsymbol{r}_{i,x}(t) = \boldsymbol{o}_i + t\boldsymbol{d}_{i,x}$, where $\boldsymbol{o}_i$ is the camera origin and $\boldsymbol{d}_{i,x}$ 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:

$$\boldsymbol{C}_{\phi,\psi}\left(\boldsymbol{r}_{i,x}\right) = \int_{t_n}^{t_f} \boldsymbol{\sigma}_\psi\left(\boldsymbol{r}_{i,x}(t)\right) T\left(\boldsymbol{r}_{i,x}(t)\right) \boldsymbol{c}_\phi\left(\boldsymbol{r}_{i,x}(t)\right) \, dt, \tag{1}$$

3

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

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

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

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 $\boldsymbol{H}_\theta(\cdot)$ of a latent random variable, e.g. a Gaussian $\boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I})$, where $\boldsymbol{H}_\theta(\cdot)$ often takes the form of a neural network that output samples $\boldsymbol{w}_j = \boldsymbol{H}_\theta(\boldsymbol{z}_j)$ with $\boldsymbol{w}_j \sim \mathbb{P}_\theta(\boldsymbol{w})$. Here, $\mathbb{P}_\theta$ is a probability measure obtained by transforming the standard Gaussian distribution measure via $\boldsymbol{H}_\theta$. Given a set of data samples $\{\hat{\boldsymbol{w}}_1, ..., \hat{\boldsymbol{w}}_N\}$, the IMLE objective optimizes the model parameters θ with

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

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 ($\boldsymbol{D}_\theta$). Concretely $\boldsymbol{D}_\theta$ is a function sample, a function defined over all points $\boldsymbol{x} \in \mathbb{R}^3$, that is sampled from the stochastic field $\mathcal{D}_\theta$, i.e. $\boldsymbol{D}_\theta \sim \mathcal{D}_\theta$. $\boldsymbol{D}_\theta$ maps each point to one possible sample in its provenances. We also denote the distribution of the provenances at point $\boldsymbol{x}$ as $\mathcal{D}_\theta(\boldsymbol{x})$. Moreover, a provenance sample $\boldsymbol{D}_\theta(\boldsymbol{x})$ from $\mathcal{D}_\theta(\boldsymbol{x})$ is equivalent to evaluating the function $\boldsymbol{D}_\theta \sim \mathcal{D}_\theta$ at $\boldsymbol{x}$. 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 $\boldsymbol{x}$ can be represented as *samples* from its provenance *distribution* $\mathcal{D}_\theta(x)$. That means that the likelihood of sampling a location $\boldsymbol{y} \in \mathbb{R}^3$ from $\mathcal{D}_\theta(x)$ determines how likely $\boldsymbol{x}$ is visible from location $\boldsymbol{y}$. Because such distributions are defined for each 3D point $\boldsymbol{x} \in \mathbb{R}^3$, the collection of per-point provenances forms a stochastic field $\mathcal{D}_\theta$ indexed by coordinates $\boldsymbol{x} \in \mathbb{R}^3$.

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

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

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

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

---

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

4
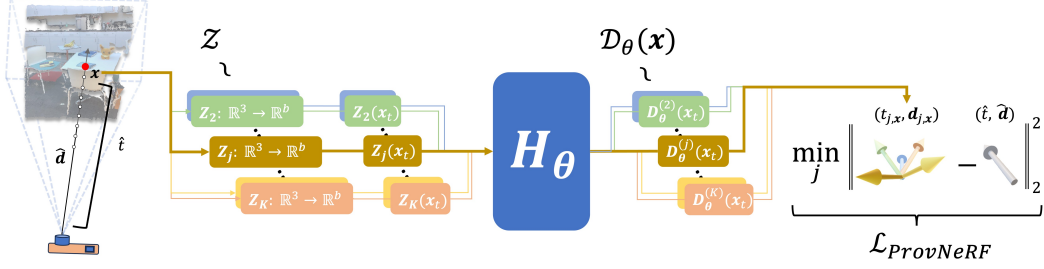
Figure 3: **Training pipeline for ProvNeRF.** For each point $\boldsymbol{x}$ seen from provenance tuple $(\hat{t}, \hat{\boldsymbol{d}})$, with direction $\boldsymbol{d}$ at distance t, we first sample K latent random functions $\{\boldsymbol{Z}_j\}$ from distribution $\mathcal{Z}$. The learned transformation $\boldsymbol{H}_\theta$ transforms each $\boldsymbol{Z}_j(\boldsymbol{x})$ to a provenance sample $\boldsymbol{D}_\theta^{(j)}(\boldsymbol{x})$. Finally $\boldsymbol{H}_\theta$ is trained with $\mathcal{L}_{\textbf{ProvNeRF}}$ 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 $\mathcal{D}_\theta(\boldsymbol{x})$ 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 $\mathcal{D}_\theta(\boldsymbol{x})$ as a distance-direction tuple $(t, \boldsymbol{d})$ as defined in Eq 5. We optimized our network with the empirical distribution $\hat{\mathcal{D}}$ as training signals.

$\mathcal{D}_\theta(\boldsymbol{x})$ defines a distribution for all point $\boldsymbol{x} \in \mathbb{R}^3$. Treating $\mathbb{R}^3$ as the index set, $\mathcal{D}_\theta = \{\mathcal{D}_\theta(\boldsymbol{x})\}_{\boldsymbol{x} \in \mathbb{R}^3}$ defines a stochastic field on $\mathbb{R}^3$ as a collection of distributions $\mathcal{D}_\theta(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^3$. 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 $\boldsymbol{H}_\theta$ 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 $\boldsymbol{Z} \sim \mathcal{Z}$ is a function $\boldsymbol{Z}: \mathbb{R}^3 \to \mathbb{R}^b$. To transform $\mathcal{Z}$ to $\mathcal{D}_\theta$, fIMLE learns a deterministic mapping $\boldsymbol{H}_\theta$ that maps each latent function $\boldsymbol{Z} \sim \mathcal{Z}$ to a function $\boldsymbol{D}_\theta \sim \mathcal{D}_\theta$ via composition: $\boldsymbol{D}_\theta = \boldsymbol{H}_\theta \circ \boldsymbol{Z}$. $\boldsymbol{H}_\theta$ here is represented as a neural network to handle complex transformations from $\mathcal{Z}$ to $\mathcal{D}_\theta$.

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

$$\boldsymbol{Z}(\boldsymbol{x}) = \left[\frac{\boldsymbol{z}}{\boldsymbol{I}}\right] \boldsymbol{x}, \text{ where } \boldsymbol{z} \sim \mathcal{N}\left(\boldsymbol{0}, \lambda^2 \boldsymbol{I}\right), \boldsymbol{x} \in \mathbb{R}^3. \tag{6}$$

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

To train $\boldsymbol{H}_\theta$, we maximize the likelihood of the training provenances (Eq. 4) under $\mathcal{D}_\theta$ for each $\boldsymbol{x}$ 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.

5

Figure 4: **Visual Effect of** $\mathcal{L}_{\mathbf{ProvNVS}}$ **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 [58]	<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 $\hat{\boldsymbol{D}}_1, \ldots, \hat{\boldsymbol{D}}_M$ from the empirical stochastic field $\hat{\mathcal{D}}$ (defined in Sec. 4.1), and model samples $\boldsymbol{D}_\theta^{(1)}, \ldots, \boldsymbol{D}_\theta^{(K)}$ from the parameter stochastic field $\mathcal{D}_\theta$, we define the fIMLE objective as

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

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 $\boldsymbol{D}_\theta$ needs a forward pass through $\boldsymbol{H}_\theta$.

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 predictions [3]. This allows us to write the fIMLE objective as

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

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

---

[3]See the supplementary for the full derivation

6

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

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

where $(t_{j,\boldsymbol{x}},\boldsymbol{d}_{j,\boldsymbol{x}}) = \boldsymbol{D}_\theta^{(j)}(\boldsymbol{x})$, $(\hat{t},\hat{\boldsymbol{d}})$ are i.i.d. samples from $\hat{\mathcal{D}}(\boldsymbol{x})$, 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 $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ for a training camera ray parameterized as $\hat{\boldsymbol{r}}_x(t)$. Here we denote point $\boldsymbol{x}_i = \hat{\boldsymbol{r}}_x(\hat{t}_i)$. We only take points $\boldsymbol{x}_i$ with transmittance greater than a selected threshold $\lambda = 0.9$. For each visible point $\boldsymbol{x}_i$, we sample provenances $(t_1^{(i)},\boldsymbol{d}_1^{(i)}),\ldots,(t_K^{(i)},\boldsymbol{d}_K^{(i)})$ from $\mathcal{D}_\theta(\boldsymbol{x}_i)$

	PSNR ($\uparrow$)	SSIM ($\uparrow$)	LPIPS ($\downarrow$)
Determinisic 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.**

with $\|\boldsymbol{d}_1^{(i)}\|_2 \geq 0.7$. Then each distance-direction tuple $(t_j^{(i)},\boldsymbol{d}_j^{(i)})$ gives a location $\boldsymbol{y}_j^{(i)} = \boldsymbol{x}_i - t_j^{(i)}\boldsymbol{d}_j^{(i)}$ from which $\boldsymbol{x}_i$ is observed. This in turn means $\boldsymbol{x}$ should be equally visible when rendered from ray parameterized as $\boldsymbol{r}_x^{(i)}(t) = \boldsymbol{y}_j^{(i)} + t\boldsymbol{d}_j^{(i)}, \forall j$. With this, we define our provenance loss for novel view synthesis as

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

where $[\ldots]_+$ denotes the hinge loss and $\alpha = 0.05$ is a constant margin. $\mathcal{L}_{\text{ProvNVS}}$ encourages the transmittance at $\boldsymbol{x}_i$ 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 $\mathcal{L}_{\textbf{ProvNeRF}}$ to optimize the NeRF representation and the provenance field, resulting in an improved scene geometry. We apply ProvNeRF to SCADE [58] 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

7
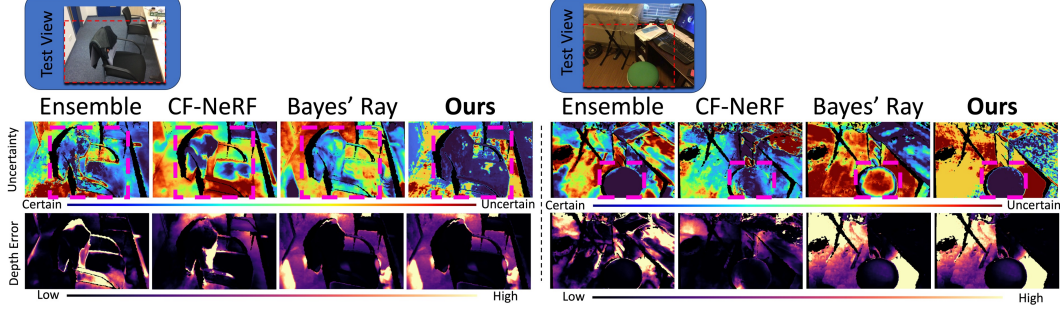
Figure 5: **Qualitative Results for Uncertainty Modeling.** We visualize our uncertainty maps obtained using the method described in Sec. 5.2. The uncertainty and depth error maps are shown with color bars specified. Uncertainty values and depth errors are normalized per test image for the result to 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	3.01	2.96	17.2	63.0	8.04	110	71.3
CF-NeRF [52]	660	430	571	980	507	799	488	233
Bayes' Rays [19]	5.47	5.11	5.23	6.07	5.49	5.67	5.77	5.91
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 x, we sample provenances $\{(t_j, d_j)\}_{j=1}^K$ from $\mathcal{D}_\theta(x)$. Treating d_j as the principal axes and t_j as the distances from x to the camera origin, each provenance sample defines a pseudo camera P_j that observes x at pixel location $x_j = \text{Proj}_j(x)$. Following chapter 12.6 of [20], we define the triangulation uncertainty of x as the probability of x given its noisy 2D pseudo observations: $\mathbb{P}(x|x_1, \ldots, x_K) \propto \mathbb{P}(x_1, \ldots, x_K|x)\,\mathbb{P}(x) = \prod_{j=1}^K \mathbb{P}(x_j|x)\,\mathbb{P}(x)$.
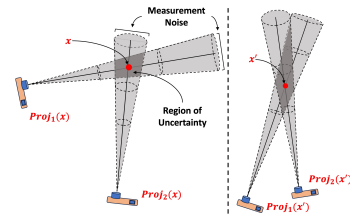
Figure 6: **Triangulation Uncertainty [20].** The figure shows that x' is more uncertain compared to x because the predicted provenances for x' give a narrower baseline than the baseline given by provenances of x.

8

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 $\mathbb{P}(x_j|\boldsymbol{x})$ follows a Gaussian distribution $\mathcal{N}(\text{Proj}_j(\boldsymbol{x}), \sigma^2)$. This assumption is equivalent to corrupting each 2D observation $\text{Proj}_j(\boldsymbol{x})$ by a zero-mean Gaussian noise with σ^2 variance, accounting for measurement noises in the capturing process. Assuming a uniform prior of $\mathbb{P}(\boldsymbol{x})$ 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 [58, 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 [58] 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 $(t, \boldsymbol{d})$ against ground truth provenances $(\hat{t}, \hat{\boldsymbol{d}})$ 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 $\boldsymbol{x}$ 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

9

| | Rendering | Depth Error | FisherRF + 3DGS | ProvNeRF + 3DGS |

NLL($\downarrow$)	FisherRF	**Ours**
# 710	5.19	**-3.93**
# 758	3.53	**-6.04**
# 781	2.73	**0.068**
Avg.	3.82	**-3.30**
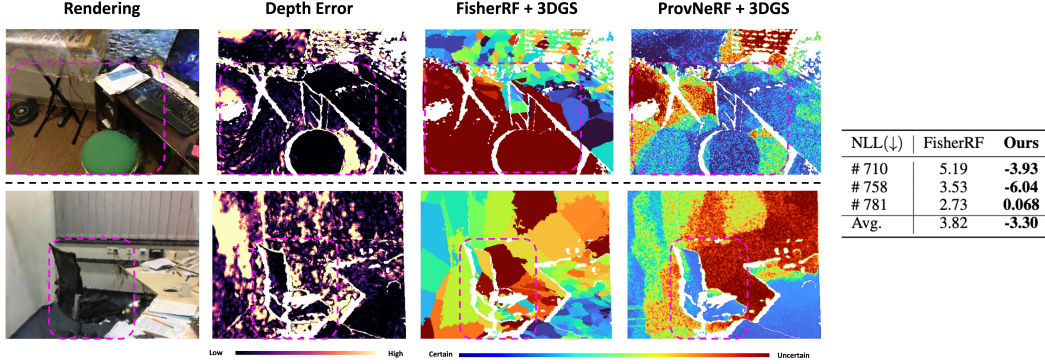
Figure 8: **Uncertainty Estimation Comparison with 3DGS.** Compared with FishRF, our method is able to estimate uncertainties that correlate more with the depth error as shown by the encircled regions. The right shows a quantitative comparison of uncertainty in negative log-likelihood.

assumption. Similarly, the VAE-based model suffers from mode-collapse while our fIMLE enables capturing a more complex distribution with a learned transformation $\boldsymbol{H}_\theta$.

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 $\mathcal{Z}^\star$ with $\mathcal{Z}^\star(\boldsymbol{x}) = [\boldsymbol{\varepsilon}, \boldsymbol{x}]\, \forall x$. Here, $\boldsymbol{\varepsilon}$ 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 $\mathcal{Z}^\star$ (**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 $\boldsymbol{H}_\theta$. 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).

10

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 Atilla 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] Jiazhong 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.

11

[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.

12

[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] Jiuhn 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, Xinchen 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] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*, SIGGRAPH '23. ACM, July 2023.

[58] Mikaela Angelina Uy, Ricardo Martin-Brualla, Leonidas Guibas, and Ke Li. Scade: Nerfs from space carving with ambiguity-aware depth estimates. In *CVPR*, 2023.

[59] 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.

[60] 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.

[61] 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.

[62] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[63] 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.

[64] 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.

13

[65] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. In *ECCV*, 2022.

[66] 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.

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

[68] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, 2022.

[69] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: Geometry editing of neural radiance fields. In *CVPR*, 2022.

[70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[71] 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.

[72] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021.

14

We provide additional results and visualizations and elaborate on implementation, metrics, baselines, and dataset details for **ProvNeRF** in this supplementary material. We organize the supplement as the following: in Sec. S2, we detail the implementation detail of **ProvNeRF**. In Sec. S3, we elaborate on the metrics, ablation, and implementation details and further provide additional visualizations on both the Scannet and Matterport datasets for our uncertainty modeling application. In Sec. S5, we provide an additional application using our provenance field that selects viewpoints given different criteria. In Sec. S4, we provide implementation details on novel view synthesis and further ablation results against other probabilistic methods. In Sec. S6, we conduct further ablation studies and explain the implementation details of the ablations introduced in the main paper. We also further provide more details on the derivation of our fIMLE objective in Sec S7. Finally, in Sec S8, we discuss the potential societal impact of our work on climate change.

S1   Provenance, Uncertainty, and NVS Visualization

We attach a video showing 3D visualization of provenances, triangulation uncertainty, and novel view synthesis results.

S2   ProvNeRF Implementation Details

Since our method can be plugged into any NeRF model, we apply our method to SCADE [58] and DDP [49], two recent state-of-the-art sparse, unconstrained NeRF models. On each of the backbones, we append an additional provenance prediction branch that is a 3-layer MLP. The provenance prediction branch takes the output of the latent random functions $Z(x)$ and outputs a provenance tuple (t_x, d_x) as defined in the main. In practice, the provenance branch outputs t_x as a normalized distance by the near and far plane of the training cameras, and the direction d_x is a 3D vector with its norm indicating the visibility/transmittance of x when seen from $y = x - td_x$.

Fig. 3 visualizes an iteration of the provenance distribution update. To train the distribution $\mathcal{D}_\theta$, we initialize **ProvNeRF** the weights of the pre-trained NeRF model and run stochastic gradient descent on the provenance prediction branch using our $\mathcal{L}_{\text{ProvNeRF}}$ loss (defined in Eq. 11 of the main paper) for 200K iterations. During the first 100K iterations we freeze the NeRF model and only update the provenance prediction branch. In each training iteration, we pick a training camera pose at random and shoot 256 rays within the image uniformly. The loss in Eq. 10 in main is then computed and backpropagated for all coarse and fine samples along the ray against the direction and camera distance from the rays. The set of random functions is resampled every 1000 iterations. The size of the provenance function pool is set to $K = 16$ and the latent space for latent random functions $\mathbb{Z}$ is set to $b = 32$. Additionally, following the usual NeRF training, the input and output of the latent random function both use the positionally encoded x instead of the actual 3D coordinate. The entire training process takes around 8 hours on an NVIDIA A5000. Once $\mathcal{D}_\theta$ is trained, provenances of each point x can be sampled by sampling random functions Z from the distribution $\mathcal{Z}$ and evaluate $H_\theta \circ Z(x)$. To obtain the visible provenances, we only use samples with the norm of the predicted directions $d_x > 0.7$.

S3   Uncertainty Estimation

Datasets & Baselines   We evaluate uncertainty quantification on two sparse, unconstrained views (outward-facing) datasets – Scannet [13] and Matterport3D [9] using the training and test splits released by [49]. Each dataset contains three scenes with $17 - 36$ training and 8 test images at a room scale. We compare our approach with the previous state-of-the-art CF-NeRF [52], concurrent work Bayes' Rays [19] and a standard derivation of uncertainty by training an ensemble of NeRFs [55]. We use the same backbone NeRF model as ours when comparing with Bayes' Rays and Ensemble. Following previous works [51, 52, 55], we report the negative log-likelihood (NLL) of $\mathbb{P}(x|\{I\})$ to measure uncertainty. We measure the NLL of the ground truth surface under the distributions given each model's uncertainty prediction. Specifically, for Ensemble, we run 10 identical models using different seeds and compute NLL of the ground truth depth under the empirical Gaussian distribution given by the 10 different depth renderings of each test view. For Bayes' Ray, we obtain the posterior likelihood by evaluating their Laplace approximated distribution $p(\theta|I)$ at $\theta = 0$. This is equivalent to evaluating the likelihood of 0 under $\mathcal{N}(0, \text{Diag}(\sigma_x, \sigma_y, \sigma_z))$ for each 3d location $x = (x, y, z)$.

1
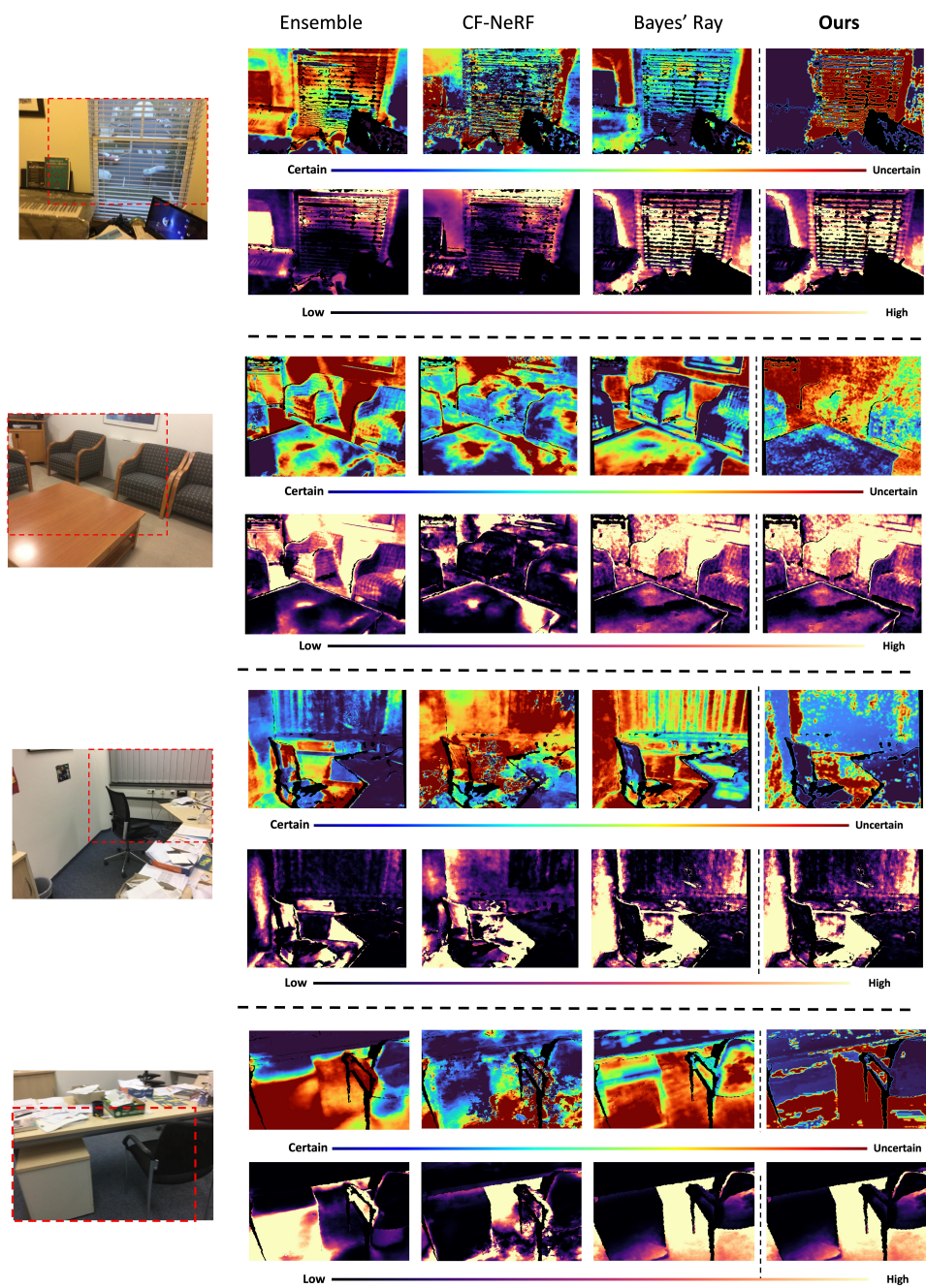
Figure S1: **Visualization of Uncertainty Estimation on the Scannet Dataset**.

2
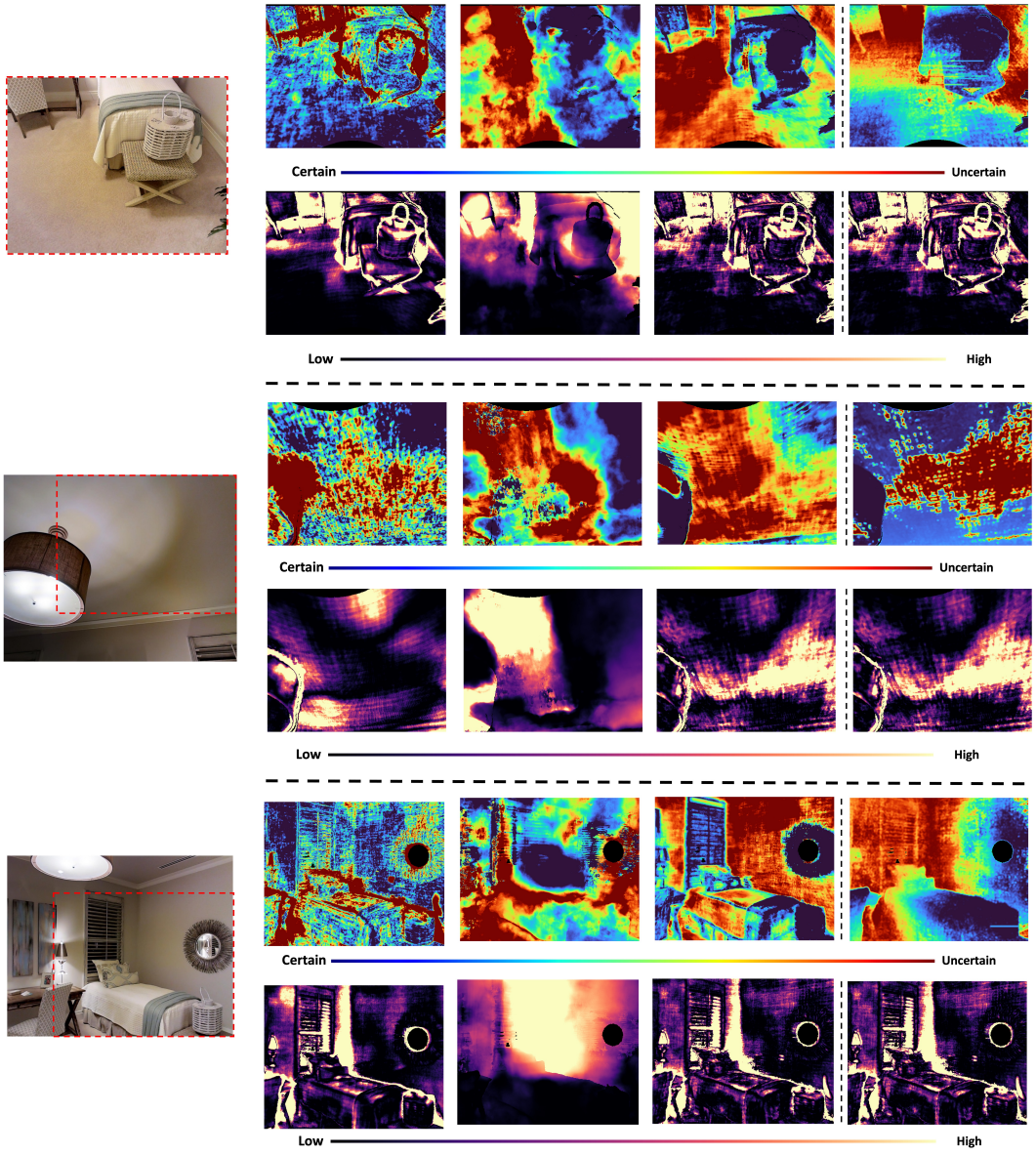
Figure S2: **Visualization of Uncertainty Estimation on the Matterport Dataset**.

3
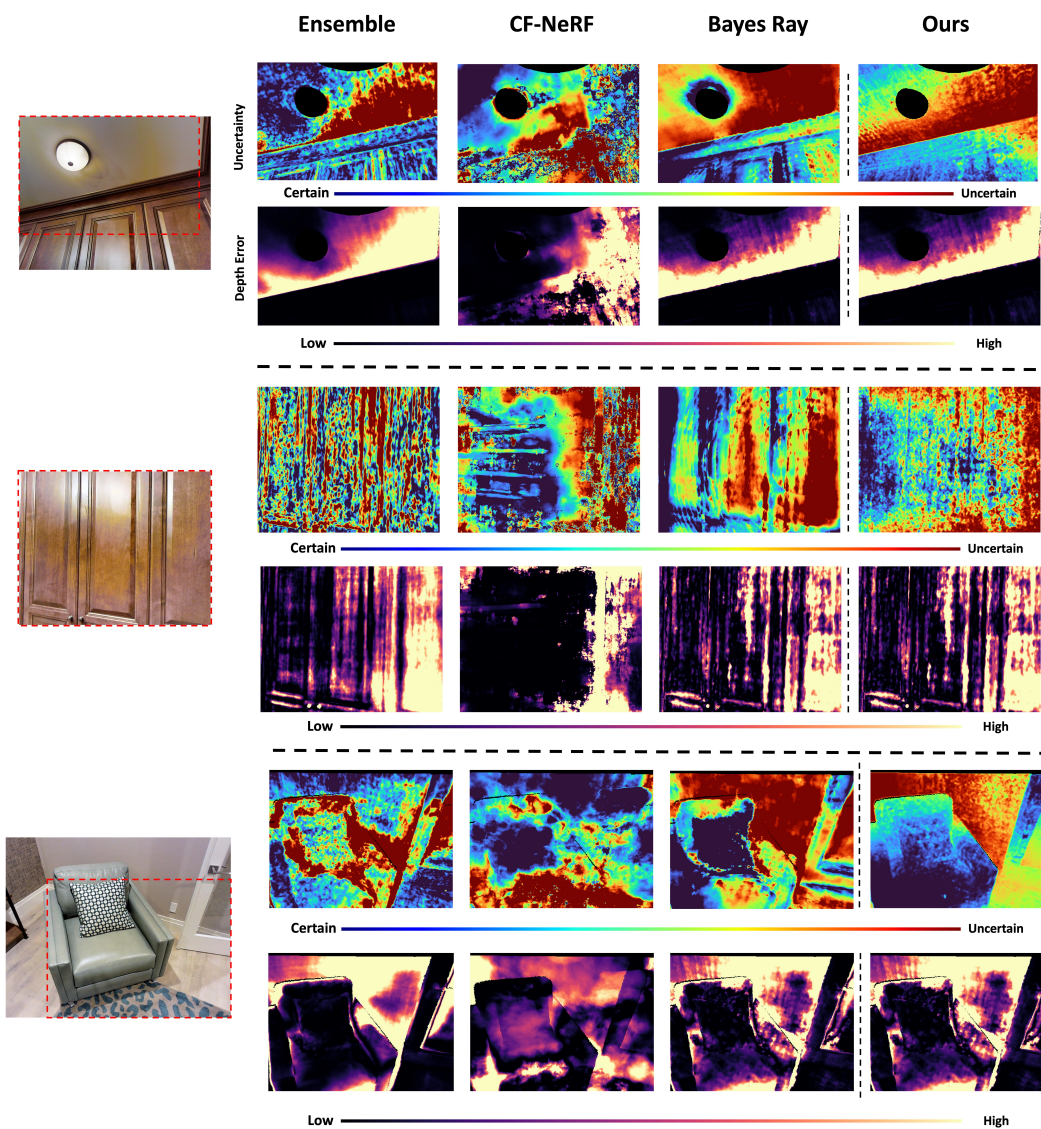
Figure S3: **Visualization of Uncertainty Estimation on the Matterport Dataset**.

4

For a fair comparison, we set Bayes' Ray's prior distribution, which is a zero-centered Gaussian, to have a standard deviation that is half of the diagonal length of the scene bound given by the training camera poses. Lastly for CF-NeRF, we follow their default setting and evaluate the ground truth depth under the empirical Gaussian given by 32 sample rays.

Additional Visualizations   Figures S2 and S3 demonstrate qualitative comparisons of baselines' and our uncertainty estimation on test images of the Matterport Dataset. Notice that in the last example of Figure S2, CF-NeRF's uncertainty does not respect the geometry of the scene, while Bayes' Ray and Ensemble mark the bed as uncertain despite it having low depth error. On the other hand, our method is able to correctly mark the bed as certain thanks to our derived uncertainty estimation based on classical photogrammetry. Moreover, even in challenging settings such as the ceiling of the first example in Figure S3, and the closeup view of the closet in the second example of Figure S3, our method is able to mark the regions with bad triangulation correctly (i.e., edge of the ceiling) using our formulation. On the other hand, approximated methods such as Bayes' Ray struggle to capture these small uncertainty variations, as Bayes' Ray simply marked the entire ceiling as uncertain.

Lastly, Figure S1 shows additional visualizations of uncertainty estimations on the Scannet Dataset. Notice that compared to baselines, our method's uncertainty estimation is able to achieve better correlations with the depth error map (e.g., The carpet under the desk in the last example, and the tabletop in the second example).

Implementation Details   To compute for the posterior likelihood shown in Eq. 13 in the main. Assuming that x is uniformly distributed within the scene bound Ω, and that the noise in 2D measurements are independent of each other given x, we rederive and simplify Eq. 13 as

$$\mathbb{P}\left(\boldsymbol{x}|x_1,\ldots,x_K\right)$$
$$=\frac{\mathbb{P}\left(x_1,\ldots,x_K|\boldsymbol{x}\right)\mathbb{P}\left(\boldsymbol{x}\right)}{\mathbb{P}\left(x_1,\ldots,x_K\right)}$$
$$=\frac{\mathbb{P}\left(x_1,\ldots,x_K|\boldsymbol{x}\right)\mathbb{P}\left(\boldsymbol{x}\right)}{\mathbb{E}_{\boldsymbol{x}'\sim\mathbb{P}}\left[\mathbb{P}\left(x_1,\ldots,x_K|\boldsymbol{x}'\right)\right]}$$

Since $x_j's$ are pairwise independent given $\boldsymbol{x}$,

$$=\frac{\mathbb{P}\left(\boldsymbol{x}\right)\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}\right)}{\mathbb{E}_{\boldsymbol{x}'\sim\mathbb{P}}\left[\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}'\right)\right]}$$
$$=\frac{\mathbb{P}\left(\boldsymbol{x}\right)\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}\right)}{\int_{\Omega}\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}'\right)}\mathbb{P}\left(\boldsymbol{x}'\right)\,d\boldsymbol{x}'$$
$$=\frac{\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}\right)}{\int_{\Omega}\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}'\right)}\,d\boldsymbol{x}'.$$

$$(\text{S3.1})$$

The last step is because $\mathbb{P}\left(\boldsymbol{x}\right)=\mathbb{P}\left(\boldsymbol{x}'\right)=\frac{1}{\text{Vol}(\Omega)}$. Since the numerator of the final expression in the above derivation is directly computable, we only need to compute the integral in the denominator. Because the computation of $\mathbb{P}\left(x_j|\boldsymbol{x}'\right)$ involves a non-linear projection operation of $\boldsymbol{x}'$ to the image space, the integral is not expressible in analytical form. Thus, we instead use importance sampling over the intersection of the camera frustums to efficiently approximate the integral.

Specifically, let $\Pi_{j,\boldsymbol{x}}$ be the camera frustum given by the camera projection matrix Proj_j as defined in Sec. 5.1 in the main, and let $\Pi_{j,\boldsymbol{x},\delta}$ be the δ neighbor camera frustum given by only pixels within δ range of the center pixel. Then, if we assume that the 2D measurements are corrupted by a zero mean, σ^2 variance Gaussian noise, it is safe to assume that the probability $\mathbb{P}\left(x_j|\boldsymbol{x}'\right)\approx 0$ for $\boldsymbol{x}'\notin\Pi_{j,\boldsymbol{x},5\sigma}$ because point outside would fall outside of 5 standard deviation of the Gaussian, and thereby has negligible probability. Thus, for the product $\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}'\right)$ to be nonzero, $\boldsymbol{x}'$ needs to be in the intersection of all 5δ neighbor frustums $\Pi_{j,\boldsymbol{x},5\sigma}$. Thus, let $\mathbb{Q}\left(\boldsymbol{x}\right)$ be a proposal distribution that is uniform within the intersection of all $\Pi_{j,\boldsymbol{x},5\sigma}\forall j=1,\ldots,K$. Then we can rewrite the last expression

5

in Eq. S3.1 as

$$\mathbb{P}\left(\boldsymbol{x}|x_1,\ldots,x_K\right)$$

$$= \frac{\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}\right)}{\int_{\Omega}\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}'\right)}\,d\boldsymbol{x}'$$

$$= \frac{\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}\right)}{\int_{\Omega_{5\delta}}\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}'\right)}\,d\boldsymbol{x}' \qquad\text{(S3.2)}$$

$$= \frac{\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}\right)}{\mathrm{Vol}\left(\Pi_{1:K,\boldsymbol{x},5\delta}\right)\mathbb{E}_{\boldsymbol{x}'\sim\mathbb{Q}}\left[\prod_{j=1}^{K}\mathbb{P}\left(x_j|\boldsymbol{x}'\right)\right]}$$

where

$$\Pi_{1:K,\boldsymbol{x},5\delta} = \bigcap_{j=1}^{K}\Pi_{j,\boldsymbol{x},5\sigma}.$$

Notice that $\Pi_{1:K,\boldsymbol{x},5\delta}$ is nonempty since $\boldsymbol{x}\in\Pi_{1:K,\boldsymbol{x},5\delta}$. The last expression in the final derivation can be efficiently and robustly approximated using Monte Carlo sampling because $\Pi_{1:K,\boldsymbol{x},5\delta}$ is the high-density region of $\mathbb{P}\left(x_j|\boldsymbol{x}'\right)$. In practice, we set the noise's standard deviation to 2 and sample 10^6 points to compute the denominator in the final expression of Eq. S3.2 for each $\boldsymbol{x}$. We set K to be 16 but only take the visible provenance samples defined by a predicted visibility of above 0.7. This means that each location $\boldsymbol{x}$ will have a variable number of associated projection cameras. Additionally, to speed up computation, we compute the posterior likelihood in the closed form if there is only one visible provenance sample from $\mathcal{D}_\theta\left(\boldsymbol{x}\right)$.
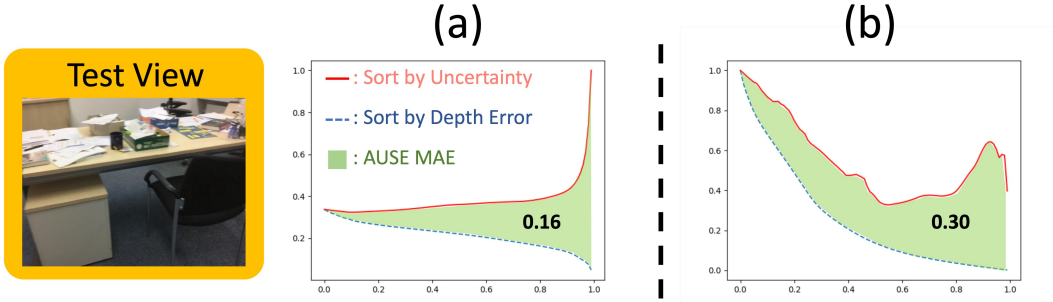
Figure S4: **Visualization of Sparsification Curves of depth MAE for Bayes' Ray (a) and Ours (b) for the test view on the left**. Notice that despite (a) assigning higher uncertainty to regions with a lower depth error as indicated by its red curve with a positive slope, it incurred a lower AUSE score than method (b)'s curve, which shows a rough correlation w.r.t. depth MAE as shown by its red curve with mostly a negative slope. For visualizations of their uncertainty estimation vis-á-vis depth error, see the last row of Figure S1.

Discussion on AUSE. We include a discussion on another metric reported in previous uncertainty estimation literature [19] – Area Under Sparsification Error (AUSE). In our work, we report the negative log likelihood (NLL) following other previous works [51, 52, 55] as we found AUSE to be not reflective of the uncertainty of the model in certain scenarios, which will be discussed in this section. AUSE was designed to measure the correlation between the uncertainty estimation and depth error. Concretely to compute for AUSE (see Figure S4), the pixels in each test image are removed gradually ("sparsified") first i) according to their depth error (blue dotted curve) and then ii) according to their uncertainty estimation (red solid curve). Then the area between the two curves is the the resulting value for the AUSE metric (green region), where lower is considered better. Here, we bring up two discussions on how the AUSE metric may not reflect the methods' uncertainty estimation.

First, the score for AUSE is good if regions with high uncertainty imply that they have high depth error. However, an uncertain region implies that its *variance* is high, which *does not* necessarily

6

imply that the region has large depth error. Such a region may have low depth error but with high variance, which will affect its AUSE score.

Second, a low AUSE number does not necessarily correspond to a good correlation between a method's uncertainty estimation and depth error. Following [19], Figure S4 shows sparsification curves for Bayes' Ray (a) and our method (b) uncertainty estimation w.r.t. to depth mean absolute error (MAE). Notice that, despite (a) obtaining a lower AUSE score (0.16) compared to ours (0.3), the curve sparsified by Bayes' Rays uncertainty has a positive slope throughout the course of sparsification. This behavior implies that for this test view, Bayes' Ray (a) actually assigned higher uncertainty values to regions with a lower depth MAE error. On the other hand, our method's sparsification curve in (b) mostly show negative slopes throughout the course of pixel removal. This indicates that high uncertainty was roughly assigned to regions with higher depth MAE error and vice versa. This observation is consistent with the visualized uncertainty maps in the last row of Figure S1. Notice that Bayes' Rays ((a) in Figure S4 and the third column from the left in Figure S1) assigned high uncertainty to the drawer and the back of the chair despite these regions having a low depth error as shown in the depth error mean below. On the other hand, our method ((b) in Figure S4 and the fourth column from the left in Figure S1) was able to correctly correlate the uncertainty estimation and the depth error despite it having a higher AUSE score than Bayes' Rays.

S4   Novel View Synthesis

Datasets & Baselines   Following previous works [49, 58, 54], we evaluate on a subset of Scan-Net [13] released by DDP [49] comprising of three scenes each with $17 - 20$ training and 8 test views. We also evaluate on a subset of Tanks and Temple released by SCADE [58] also comprising of three scenes. For quantitative evaluation, we report the standard metrics PSNR, SSIM [62], LPIPS [70] on the novel test views. We compare with recent NeRF-based works including state-of-the-art methods DDP [49], SCADE [58] and DäRF [54] on the sparse, unconstrained setting.

Implementation Details   We initialize our ProvNeRF with pre-trained SCADE [58] model and finetune the model with our proposed objective $\mathcal{L}_{\text{ProvNVS}}$ (Eq. 15 main paper) for another 200K iterations. The provenance loss is used together with the photometric loss and the space carving loss:

$$\mathcal{L}_{\textbf{ProvNeRF}} + \mathcal{L}_{\text{ProvNVS}} = \mathcal{L}_{\text{photometric}} + \beta_1 \mathcal{L}_{\text{space carving}} + \beta_2 \mathcal{L}_{\text{ProvNVS}} + \beta_3 \mathcal{L}_{\text{fIMLE}}. \tag{S4.1}$$

We set $\beta_1 = 0.005$ for all scenes and $\beta_2 = 0.005$ for all scenes except for $\beta_2 = 0.01$ for scene 758 in the Scannet Dataset. $\beta_3 = 0.1$ is set for all scenes. We set the learning rate to be $5e - 5$ for all scenes and we only enforce the provenance loss for points with rendered transmittance.

	PSNR ($\uparrow$)	SSIM ($\uparrow$)	LPIPS ($\downarrow$)
Determinisic Field	21.38	0.720	0.307
Gaussian-based w/ $C = 2$	20.77	0.702	0.336
Gaussian-based w/ $C = 5$	21.04	0.714	0.326
VAE-based	19.28	0.657	0.334
Frustum Check	21.56	0.728	0.297
ProvNeRF w/ Spatial Inv. $\mathcal{Z}$	21.63	0.731	0.294
Ours*	21.68	0.732	**0.291**
Ours	**21.73**	**0.733**	**0.291**

Table S1: **NVS Ablation Results on Scannet. Ours*** only updates the underlying NeRF representation without updating the provenance branch.

NVS Depth Comparison   In Fig. S5 we compare the rendered depth maps before and after applying our NVS regularizer. Notice that our method improves SCADE's depth map by removing floaters in the test views.

NVS Result Ablation   We also ablate our choice of fIMLE in the task of novel view synthesis against other probabilistic methods. We compare our approach against the baselines introduced in Section. 5.3 of the main paper in Tab. S1. Because of our ability to model complex provenance

7
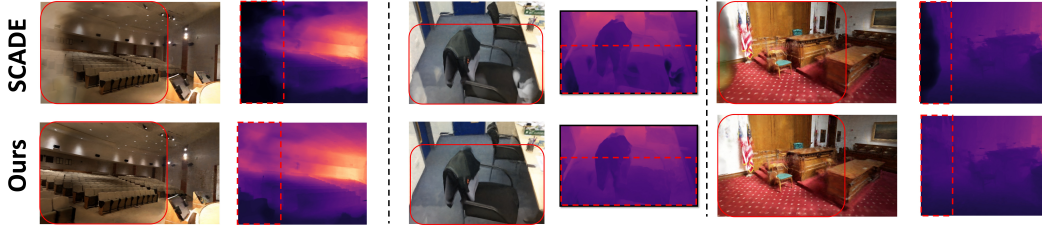
Figure S5: **NVS Results Shown with Depth Maps.** Compared to the pretrained SCADE, our method additionally improves scene reconstruction by removing clouds in the encircled region of the depth maps.

distributions, we achieve better NVS results compared to other probabilistic methods. Notably, jointly training the provenance branch and the underlying NeRF representation further improves the NVS quality, as shown by the comparison of our method with **Ours***, which freezes the provenance branch and only updates the underly NeRF weights. The improvement in joint optimization testifies to the mutual benefits between learning provenance and obtaining better geometry.

S5 Additional Provenance Application: View Selection based on Area Maximization

We further demonstrate the usefulness of Provenances for downstream tasks. We show that our per-point provenance can be applied to criteria-based viewpoint selection, which leverages a neural rendering framework to determine the most favorable camera perspective based on a predefined criterion. For instance, this can encompass orienting the camera to align with the normal vector of a specified target or achieving a detailed close-up view of the target. The first criterion can be attained by minimizing the negative dot product between the camera's principal axis and the object's normal, whereas the second can be achieved through the minimization of the negative area of the 2D projection from a 3D plane encasing the object. A naive solution is to directly run gradient descent to optimize each corresponding objective loss $\mathcal{L}_{\text{obj}}$.

We propose that the integration of two novel viewing loss functions within our model enhances the gradient's informativeness, thereby facilitating an approximation to the objective that is superior to conventional methods. More precisely, for each 3D point $\boldsymbol{x}_i$ lying on the object of interest, using our **ProvNeRF**, we sample provenances $(t_1^{(i)}, \boldsymbol{d}_1^{(i)}), \ldots, (t_K^{(i)}, \boldsymbol{d}_K^{(i)})$ from the distribution $\mathcal{D}_\theta(\boldsymbol{x}_i)$. Each distance-direction tuple sample $(t_j^{(i)}, \boldsymbol{d}_j^{(i)})$ gives a location $\boldsymbol{y}_j^{(i)} = \boldsymbol{x}_i - t_j^{(i)} \boldsymbol{d}_j^{(i)}$ from which $\boldsymbol{x}_i$ is seen, based on this, we define an additional loss $\mathcal{L}_{\text{select}} = -(\mathcal{L}_c + \mathcal{L}_d)$ as follows:

$$\mathcal{L}_c = \sum_i \max_j \|\boldsymbol{y}_j^{(i)} - \boldsymbol{c}\|_2^2; \ \mathcal{L}_d = \sum_i \max_j \left(R_{:,3}^\top \boldsymbol{d}_j^{(i)} \right), \tag{S5.1}$$

where $[R|\boldsymbol{c}]$ denotes the camera pose, $R_{:,3}$ corresponds to the third column of R, reflecting the principal axis of the camera in the global coordinate system. Intuitively, $\mathcal{L}_c$ seeks to align the camera center with the network's prediction, whereas $\mathcal{L}_d$ aims to orient the principal axis of the camera toward the points. By running gradient descent on $\mathcal{L}_{\text{select}} + \mathcal{L}_{\text{obj}}$ from an initial viewpoint of the target object, our method can achieve better PSNR while moving closer to the desired criteria. We benchmark against two baselines: 1) a naive approach that exhaustively searches all training views to locate the one that minimizes L_{obj}, and 2) a direct optimization approach that applies gradient descent solely on L_{obj} without considering L_{select}.

Results. Both the quantitative (Left) and qualitative (Right) outcomes, presented in Figure S6, demonstrate that our approach achieves a superior balance between PSNR and the intended objective [4]. Although the naive retrieval baseline attains a higher PSNR, it fails to satisfactorily meet the desired objective, especially when training views are sparse and limited.

---

[4]To compute for PSNR at arbitrary poses, we train an oracle NeRF using the full captured sequence. We use Nerfacto [57] as the oracle NeRF's representation.

8
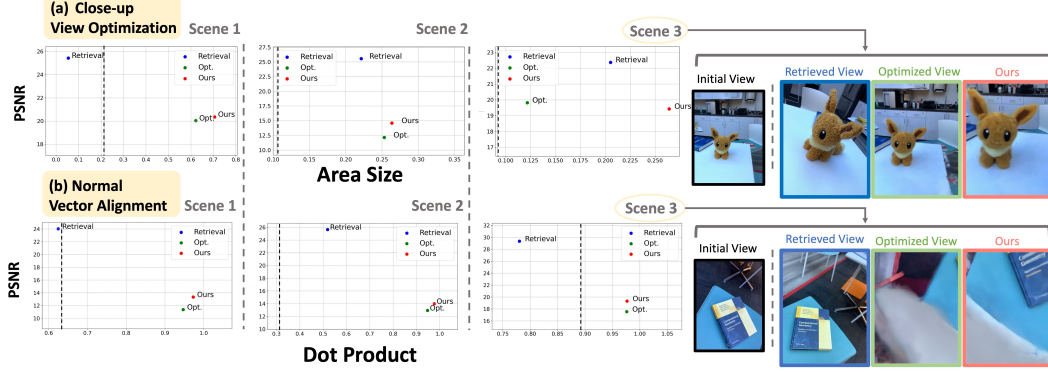
Figure S6: **(a)** PSNR and Area Size plots with the objective of maximizing the projected area of the target. **(b)** PSNR and Dot Product plots the objective of maximizing the dot product between the viewing angle and the target object's normal. The dotted lines denote the objective for the initial views. *(Right)* We show the final view comparison of our provenance-aided viewpoint selection compared to the baselines under the two objectives in scene 3. Notice that our method (in red) is able to arrive at an objective maximizing view while retaining reconstruction quality. On the other hand, both the retrieval and optimization baselines fail to balance between the two.

S6 Ablation Study

Metrics Detail In the Ablation section refsec_ablation of main, we computed the Average Precision (AP) and Area Under the Curve (AUC) scores with the confidence score given by the Euclidean distance between the predicted sample and ground truth observation. Here we detail our metric computation. We follow the usual definition of AP and AUC. Given a set of confidence thresholds $\delta_0, \ldots, \delta_{N-1}$, we define

$$\text{AP} = \sum_{k=0}^{N-1} \left[\text{Recall}\,(k) - \text{Recall}\,(k+1) \right] \cdot \\ \text{Precision}\,(k) \tag{S6.1}$$

where $\text{Recall}\,(N) = 0$ and $\text{Precision}\,(N) = 1$. On the other hand, AUC is defined as the area under the Precision and Recall curve of the set of thresholds. Thus, to compute for AP and AUC, we need to define how do we measure recall and precision in our setting. To do so, for 3D point $\boldsymbol{x}$, we use the set of provenances $\text{Prov}\,(\boldsymbol{x}) \setminus \{(0, \mathbf{0})\}$ as defined in Eq. 4 [5] of the main as the ground truth labels, and we sample $K = 128$ provenances from the learned provenance distribution $\mathcal{D}_\theta\,(\boldsymbol{x})$ and only take the samples with visibility above 0.9. If we define a discrete uniform distribution over $\widehat{\text{Prov}}(\boldsymbol{x})$ as $\mathcal{U}\,(\boldsymbol{x})$, for a particular confidence threshold δ, the precision and recall at $\boldsymbol{x}$ is defined as

- **Precision** $(\boldsymbol{x}) =$

$$\mathbb{E}_{(\hat{\boldsymbol{d}}_i, \hat{t}_i) \sim \mathcal{U}(\boldsymbol{x})} \left[\mathbb{1}_{\left\{ \min_j \left\| ((\boldsymbol{d}_j, t_j) - (\hat{\boldsymbol{d}}_i, \hat{t}_i)) \right\|_2^2 < \delta \right\}} \right] \tag{S6.2}$$

- **Recall** $(\boldsymbol{x}) =$

$$\frac{1}{K} \sum_{i=1}^{K} \mathbb{1}_{\left\{ \min_{\widehat{\text{Prov}}(\boldsymbol{x})} \left\| ((\boldsymbol{d}_j, t_j) - (\hat{\boldsymbol{d}}_i, \hat{t}_i)) \right\|_2^2 < \delta \right\}}, \tag{S6.3}$$

We note that **Precision**$(\boldsymbol{x})$ measures whether all the ground truth observations are covered by the predicted observations, while **Recall**$(\boldsymbol{x})$ measures if the predicted observations are close to one of the ground truth observations. We use a set of 500 thresholds log-linearly interpolated from e^{-20} to 1 for accurate and robust AP and AUC computation.

Further Ablation Study We further ablate on the number of random function samples K to take during each time of resampling. Tab. S2 shows quantitative AP and AUC results on the Scannet dataset. As shown in the table, the quality of the provenance samples improves as we increase the

---

[5]The visibility term is computed based on the ground truth geometry.

9

K	AP ($\uparrow$)	AUC ($\uparrow$)
1	0.197	0.199
2	0.534	0.536
4	0.644	0.646
8	0.738	0.740
16	**0.745**	**0.747**

Table S2: **Ablation Results on Scannet.**

number of latent random function samples. This is because a larger number of sample pools can help the deep transformation better transform the distribution from latent space to the space of provenances. We note that the improvement gradually becomes marginal as K increases beyond 8. Thus, we use $K = 16$ for all of our experiments.

Implementation of Deterministic Field   Deterministic Field (see entry "Deterministic Field" in Tab. 3 main paper) is implemented using the same architecture as the provenance branch of **ProvNeRF** (i.e., 3 layers of MLP). Input the deterministic field is set to the positionally encoded 3D location and the output is a direction and distance tuple, similarly parameterized as **ProvNeRF** (For detail of the parameterization, please refer to Sec. S2). The deterministic field is trained with the mean square distance of the predicted provenance samples at x with an empirical sample from $\hat{\mathcal{D}}(x)$. We train the deterministic field for 200K iterations using the same hyperparameters as our method.

Implementation of Gaussian-Based Models   The Gaussian-based models (see Gaussian-based w/ C=2, C=5 entries in Tab. 3 main paper) are parameterized using a 3-layer MLP that takes in a 3D location and outputs the means, variances, and weights of each Gaussian. The Gaussian models are trained with the negative log-likelihood of the empirical provenance samples from $\hat{\mathcal{D}}(x)$ under the parameterized Gaussian mixture. The models are trained for 200K iterations with the same set of hyperparameters as **ProvNeRF**.

S7   Derivation of Objective

In Sec. 4.3 of the main paper, we showed that the fIMLE objective in Eq. 8 of the main paper is equivalent to a pointwise matching loss using the theory of calculus of variation and Euler-Lagrange Equation. We outline the details here.

Specifically, assuming $\boldsymbol{D}_{\theta}^{(j)}$ is differentiable for all $j = 1, \ldots, K$, we show that any minimum for the L2 norm difference in Eq. 9 of the main paper is only achieved when $\boldsymbol{D}_i$ and $\boldsymbol{D}_{\theta}^{(j)}$ are pointwise equivalent. Fix M empirical function samples $\hat{\boldsymbol{D}}_1, \ldots, \hat{\boldsymbol{D}}_M$ from $\hat{\mathcal{D}}$, and K i.i.d. sample functions $\boldsymbol{D}_{\theta}^{(1)}, \ldots, \boldsymbol{D}_{\theta}^{K}$ from the distribution $\mathcal{D}_{\theta}$. Then, we define a functional J of K variables in $\boldsymbol{D}_{\theta}^{(1)}, \ldots, \boldsymbol{D}_{\theta}^{K}$ in the form of

$$J\left(\boldsymbol{D}_{\theta}^{(1)}, \ldots, \boldsymbol{D}_{\theta}^{(K)}\right) = \frac{1}{2}\sum_{i=1}^{M}\left\|\hat{\boldsymbol{D}}_i - \boldsymbol{D}_{\theta}^{(j_i)}\right\|_{L^2}^2 \tag{S7.1}$$

where

$$j_i = \arg\min_{j=1,\ldots,K}\left\|\boldsymbol{D}_i - \boldsymbol{D}_{\theta}^{(j)}\right\|_{L^2}^2$$

for each $i = 1, \ldots, M$ denote the argmin of $\left\|\boldsymbol{D}_i - \boldsymbol{D}_{\theta}^{(j)}\right\|_{L^2}^2$ over $j \in \{1, \ldots, K\}$. Then, by the Euler-Lagrange Equation, we know that if the tuple $\left(\boldsymbol{D}_{\theta^*}^{(1)}, \ldots, \boldsymbol{D}_{\theta^*}^{(K)}\right)$ is a minimizer to J, they must satisfy

$$\frac{\partial}{\partial \boldsymbol{D}_{\theta}^{(j)}}J\left(\boldsymbol{D}_{\theta^*}^{(1)}(\boldsymbol{x}), \ldots, \boldsymbol{D}_{\theta^*}^{(K)}(\boldsymbol{x})\right) = 0 \tag{S7.2}$$

10

for all $\boldsymbol{x} \in \mathbb{R}^3$ and $j = 1, \ldots, K$. On the other hand, taking the functional derivative with respect to each of the K variables of J, we derive that at point $\boldsymbol{x} \in \mathbb{R}^3$,

$$\frac{\partial}{\partial \boldsymbol{D}_\theta^{(k)}} J\left(\boldsymbol{D}_\theta^{(1)}(\boldsymbol{x}), \ldots, \boldsymbol{D}_\theta^{(K)}(\boldsymbol{x})\right)$$

$$= \begin{cases} 0 & \text{if } k \notin \{j_1, \ldots, j_M\} \\ \left\|\hat{\boldsymbol{D}}_i(\boldsymbol{x}) - \boldsymbol{D}_\theta^{(k)}(\boldsymbol{x})\right\|_2 & \text{if } k = j_i \end{cases}. \tag{S7.3}$$

Then, combining Eqs S7.1 and S7.3, we see that objective 8 in the main only can achieve its minimum when

$$\left\|\left(\hat{\boldsymbol{D}}_i - \boldsymbol{D}_\theta^{(j_i)}\right)(\boldsymbol{x})\right\|_2 = 0, \ \forall \boldsymbol{x} \in \mathbb{R}^3. \tag{S7.4}$$

The above derivation shows that minimizing the L^2 differences of functions is equivalent to minimizing the norm differences of the functions at all points. This shows the equivalence of the fIMLE objective in Eq. 8 and **ProvNeRF** objective Eq. 10 in the main.

S8 Societal Impact

Our models require the usage of GPUs both in training time and inference time. We acknowledge that this contributes to climate change which is an important societal issue. Despite this, we observe the improvement in the results that is theoretically grounded and addresses a gap in the existing literature on modeling the locations of likely visibility of NeRFs. To alleviate this issue, we only test our method on selected scenes before scaling up to different scenes across datasets to minimize its impact on climate change.

S9 Used Dataset and Codebases

In our project, we tested on publicly available dataset released by SCADE [58] and DDP [49]. We built our code by leveraging released codebases from SCADE, DDP, NeRF-Pytorch, and 3DGS.

11