Neural Visibility Field for Uncertainty-Driven Active Mapping

Shangjie Xue Jesse Dill Pranay Mathur Frank Dellaert Panagiotis Tsiotras Danfei Xu
Georgia Institute of Technology
{xsj, jdill33, pranay.mathur, frank.dellaert, tsiotras, danfei}@gatech.edu

Abstract

This paper presents Neural Visibility Field (NVF), a novel uncertainty quantification method for Neural Radiance Fields (NeRF) applied to active mapping. Our key insight is that regions not visible in the training views lead to inherently unreliable color predictions by NeRF at this region, resulting in increased uncertainty in the synthesized views. To address this, we propose to use Bayesian Networks to composite position-based field uncertainty into ray-based uncertainty in camera observations. Consequently, NVF naturally assigns higher uncertainty to unobserved regions, aiding robots to select the most informative next viewpoints. Extensive evaluations show that NVF excels not only in uncertainty quantification but also in scene reconstruction for active mapping, outperforming existing methods. More details can be found at https://sites.google.com/view/nvf-cvpr24/.

1. Introduction

Active 3D reconstruction plays a pivotal role in robotics. The challenge lies in enabling the robot to precisely reconstruct a target using the fewest views possible. Consider the example, illustrated in Figure 1, where the agent's objective is to thoroughly explore an unknown object (the Hubble telescope). To achieve this, the robot assesses the uncertainty of potential views, choosing actions that significantly diminish this uncertainty. A crucial aspect of this process is the representation of the scene. It should not only facilitate high-quality reconstruction but also be cognizant of uncertainties.

Recently, implicit scene representations, notably NeRF [31] have shown remarkable ability in high-quality scene reconstructions. The result has motivated applying NeRF for active reconstruction [34, 54, 57]. However, due to the opaque nature of neural networks, estimating the uncertainty of NeRF remains challenging. Previous works have developed various proxy measurements to represent the uncertainty in NeRF, in which they aim to maximize the NeRF's reconstruction accuracy and geometric faithfulness to the scene. However, these approaches neglect a crucial factor to optimize for, namely, visual coverage.

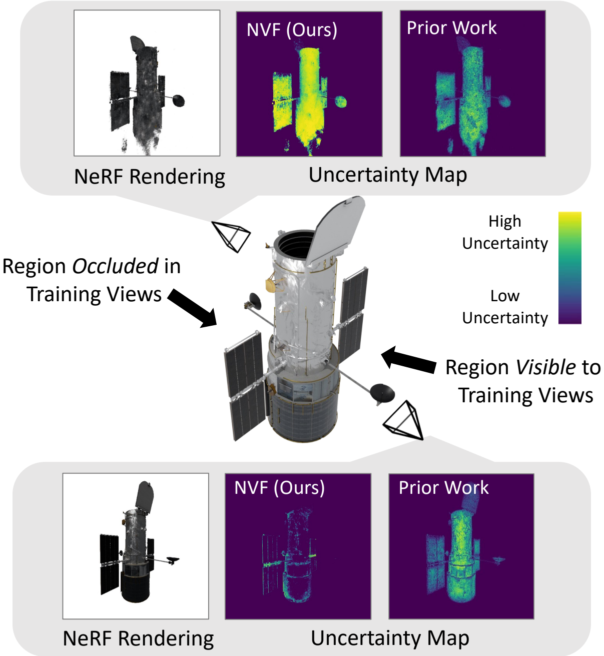

Figure 1. Neural Visibility Field (NVF) is an uncertainty estimation framework for NeRF that accounts for *visibility*: whether a region is covered by the training views of a NeRF. Visible regions should have low uncertainty (bottom row), and unobserved should have high uncertainty (top row). In this paper, we show that many existing methods in NeRF uncertainty quantification can be viewed as special cases of our framework, and NVF outperforms them empirically in uncertainty quantification and active mapping tasks.

In active reconstruction, an agent makes a tradeoff between exploring new areas of a scene and revisiting previously explored ones. Since NeRF is a multiview reconstruction method, a natural strategy is to explore regions that have not been observed by previous views and have these regions hold a high degree of uncertainty. Surprisingly, prior methods have largely failed to account for visibility and instead focus on estimating uncertainty via density or NeRF-predicted position-based RGB variance. Another gap in prior research is the integration of position-based uncertainty factors (e.g., emitted color, opacity, and visibility) into ray-based observation uncertainty. Previous approaches typically employ a simple (weighted) average or sum of position-based uncertainties to approximate the observation uncertainty. However, these methods often lack a solid theoretical foundation and can underperform in complex scenarios.

To address these challenges, we propose Neural Visibility Field (NVF). Our key insight is that if a region has never been visible in the training views, the color prediction for this point by NeRF is unreliable. To effectively integrate this location-based uncertainty into ray-based camera observations, we view NeRF through the lens of a Bayesian Network. Within this framework, the distribution of a color along a ray can be interpreted as a Gaussian mixture Model. Subsequently, we calculate the entropy of the GMM and employ it as a cost function, guiding the agent to select the next best view for active mapping. We observed that all previous methods can be interpreted as specific approximations within our proposed theoretical framework, yet, they consistently overlook a crucial aspect, namely, visibility.

Our evaluation of the proposed approach is multi-faceted and spans a range of environments, encompassing objects, indoor rooms, and spaces. We illustrate how our method offers a superior metric for assessing uncertainty in NeRF. We also apply our approach to active mapping tasks. Specifically, we demonstrate that employing our metric in Next-Best-View (NBV) planning facilitates the planning of trajectories that not only enhance reconstruction quality but also maximize visual coverage of the scene. As a result, our proposed method demonstrates significant improvements over these prior approaches in experimental evaluations. To summarize, our main contributions are:

- We propose a principled uncertainty estimation method for NeRF that takes into account visibility, called Neural Visibility Field (NVF).
- We provide a unified lens of prior methods in uncertainty estimation for NeRF using NVF.
- We apply the NVF framework to active mapping tasks demonstrating superior performance compared to the existing state-of-the-art methods.

2. Related Work

Active Mapping. Research on active mapping or NBV selection is a long-studied problem [4, 37] with the goal of searching for observation poses to create an optimal reconstruction of an environment. Scott et al [42] categorizes these approaches as model-based approaches, which utilize knowledge of the geometry and appearance of a scene [11, 41], and model-free approaches, which use information extracted from data gathered online [2, 4, 37]. More relevant are viewpoint selection strategies, including frontier-based [6, 9], sampling-based [7, 12, 39], and uncertainty based [44, 45]. In particular, our method is inspired by the line of work that uses probabilistic volumetric occupancy to facilitate visibility operations [16, 21], which employs the concept of entropy to estimate uncertainty.

Implicit Scene Representation. Implicit neural fields [29, 35, 53] represent 3D scenes as a continuous differentiable signal parameterized via a neural network. The seminal work of Neural Radiance Fields (NeRFs) [30] learns a density and a radiance field supervised by multiview 2D images. New views can be queried from a trained NeRF through volumetric rendering. Along this direction, significant progress has been made in novel view rendering [28, 30, 49], 3D reconstruction [1, 23], 3D generation [17, 38] and videos [10, 24, 25, 36, 52]. Despite their success, the quality of representation hinges on using a large number of well-posed images which limits their applicability in real-time applications. To counter these problems, recent work has focused on few-shot neural rendering [3, 8, 33, 55], handling unknown or noisy camera pose estimates [26, 51], using heuristic camera placement strategy [20], or adding a notion of uncertainty [18, 22, 27, 45] to quantify information gain for next-best-view selection.

Uncertainty Estimation for NeRF. This work focuses on quantifying the epistemic uncertainty of a NeRF model to determine the next best view for improving its reconstruction. Direct approaches such as ensemble-based methods [47] are conceptually simple but computationally expensive or require prior data collection [14, 18]. Our method improves upon and unifies a recent line of work [22, 34, 40, 45, 54, 57]. ActiveNerf [34] and NeurAR [40] model RGB color distribution at a specific spatial point as a Gaussian distribution, and directly use NeRF to predict its variance. However, the predicted variance tends to be inaccurate in instances where a region has never been visible from the training views. In comparison, [22, 45, 54, 57] ignore the spatial RGB uncertainty, and approximating the entropy through the probability of occupancy by using NeRF's density prediction. In particular, [22] treats the sampled points in volumetric rendering that are displayed by pixels as discrete random variables and computes the entropy based on it. In [54, 57], the entropy is approximated by utilizing the probability of a ray being occluded at a point. However, it is worth noting that a remaining gap exists in all previous methods as they lack theoretical grounding for bridging the position-based uncertainty or occupancy uncertainty with ray-based observation uncertainty. In our work, we proposed a theoretically principled method based on Bayesian Network to address this challenge. Moreover, a crucial aspect of uncertainty estimation is that if a region is never visible by any of the previous views, the NeRF prediction at this region is not reliable, and high uncertainty should be associated with these regions, yet this aspect is overlooked by all relevant previous works. Our proposed theoretical framework enables us to properly model this aspect through visibility, and all previous work could be viewed as special cases under our proposed framework while lacking certain key aspects.

3. Method

Active mapping aims to reduce uncertainties in the reconstructed map and achieve visual coverage of the entire scene. This is achieved by assessing the current uncertainties in the reconstructed map and predicting the potential information gained from proposed viewpoints. However, the challenge arises when utilizing NeRF for active mapping. The opaque and complex nature of neural networks presents significant challenges for accurately quantifying uncertainty within NeRF. Although several methods have been proposed to approximate uncertainty in NeRF, they often lack a theoretical foundation and may underperform in complex scenarios. Our key insight is that if a region has never been visible in the training views, the color prediction for this point by NeRF is inherently unreliable. To effectively incorporate this position-based uncertainty into camera observations, we propose to use a Bayesian network. This allows for the seamless integration of uncertainty from the implicit field into the observed image's uncertainty. In this section, we will start with a review of NeRF, followed by a detailed explanation of how to model NeRF's volume rendering process using a Bayesian network. Subsequently, we will delve into the integration of visibility aspects within the framework. Finally, we will discuss the application of this framework to active mapping.

3.1. Problem Formulation

A NeRF [31] is defined as an implicit function $F_\Theta : (\boldsymbol{x}, \boldsymbol{d}) \rightarrow (\boldsymbol{c}, \sigma)$, where $\boldsymbol{x}$ represents the 3D position, $\boldsymbol{d} = (\theta, \phi)$ the viewing direction, $\boldsymbol{c}$ the emitted RGB color at $\boldsymbol{x}$, and σ the volume density at $\boldsymbol{x}$. The volume density function $\sigma(\boldsymbol{x})$ is a differentiable measure of the probability that a ray is occluded at position $\boldsymbol{x}$. Considering a ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ with near and far bounds t_n, t_f, the observed color at the ray's origin is given by:

$$C(\boldsymbol{r}) = \exp\left(-\int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})\mathrm{d}t\right) \quad (1)$$

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\boldsymbol{r}(s))\mathrm{d}s\right) \quad (2)$$

where $T(t)$ is the transmission probability from t_n to t without occlusion. Empirically, the algorithm approximates the integral with N discrete samples $\{t_i\}_{i=0}^{N-1}$ along the ray, where t_0 denotes the ray's origin where the camera is located. Consequently, Eq. (1) becomes:

$$\hat{C}(\boldsymbol{r}) = \sum_i w_i \boldsymbol{c}(t_i), \text{ where } w_i = \alpha_i \prod_{j=0}^{i-1}(1 - \alpha_i), \quad (3)$$

where $s_i = t_{i+1} - t_i$ denotes the distance between two adjacent sampled points along the ray, $\alpha_i = 1 - \exp(-s_i\sigma(t_i))$ is the alpha value in alpha composition, which can also be viewed as the probability of occlusion at the ith point.

3.2. Volume Rendering as Bayesian Network

While NeRF synthesizes novel views, NeRF cannot estimate the uncertainty in the views. We introduce a method that composites position-based uncertainty into ray-based uncertainty using a probabilistic graphical model. This framework enables the integration of visibility factors into the uncertainty estimation process (see Sec. 3.3). We consider the observed color along a ray, $C(\boldsymbol{r})$, as a random variable instead of a constant. In this subsection, we detail the computation of this variable's distribution by using a Bayesian network to model the volume rendering process. We use a binary random variable D_i to denote whether the ray is occluded in the interval $[t_i, t_{i+1}]$ ($D_i = 1$ for occluded, $D_i = 0$ for transparent). The continuous random variable C_i then represents the emitted color at t_i in the direction $\boldsymbol{d}$, and $\boldsymbol{Z}_i$ is a continuous random variable for the observed color at t_i. Here, $\boldsymbol{Z}_0$ corresponds to the camera's observed color at the origin, and hence, the goal is to compute $p(\boldsymbol{z}_0)$. Notice that, although both $\boldsymbol{C}_i$ and $\boldsymbol{Z}_i$ represent colors, their difference lies in the objects they represent: $\boldsymbol{C}_i$ represents the color distribution associated with a specific position in $\mathbb{R}^3$, whereas $\boldsymbol{Z}_i$ corresponds to the color distribution of a camera observation, associated with a particular ray. For simplicity, we omit the ray index $\boldsymbol{r}$ for D_i, $\boldsymbol{C}_i$, and $\boldsymbol{Z}_i$ in Sections 3.2 and 3.3 below.

Note that the value of $\boldsymbol{Z}_i$ only depends on D_i, $\boldsymbol{C}_i$, and $\boldsymbol{Z}_{i+1}$. Specifically, if the interval $[t_i, t_{i+1}]$ occludes the ray, $\boldsymbol{Z}_i$ assumes the emitted color $\boldsymbol{C}_i$; otherwise, $\boldsymbol{Z}_i$ equals $\boldsymbol{Z}_{i+1}$, as the interval is transparent. The conditional probability $p(\boldsymbol{z}_i | D_i, \boldsymbol{c}_i, \boldsymbol{z}_{i+1})$ is thus:

$$p(\boldsymbol{z}_i | D_i, \boldsymbol{c}_i, \boldsymbol{z}_{i+1}) = \begin{cases} \delta(\boldsymbol{z}_i - \boldsymbol{c}_i), & \text{if } D_i = 1, \\ \delta(\boldsymbol{z}_i - \boldsymbol{z}_{i+1}), & \text{otherwise,} \end{cases} \quad (4)$$

where $\delta(\cdot)$ is the Dirac delta function. Thus, the volume rendering process could be modeled as a hybrid Bayesian network (illustration included in the Appendix). Moreover, we can express the marginal probability of $\boldsymbol{z}_i$ using the following recursion:

$$p(\boldsymbol{z}_i) = \alpha_i p(\boldsymbol{c}_i) + (1 - \alpha_i) p(\boldsymbol{z}_{i+1}). \quad (5)$$

Note that this formulation utilizes the relationship $P(D_i = 0) = 1 - \alpha_i = \exp(-\sigma_i s_i)$. where α_i was previously defined as the probability of occlusion at the ith point along the ray. If we assume $p(\boldsymbol{c}_i)$ is a Gaussian distribution with mean $\boldsymbol{\mu}_{\boldsymbol{c}_i}$ and covariance $\boldsymbol{Q}_{\boldsymbol{c}_i}$, as predicted by the NeRF model (see Sec. 3.4 for details), by using recursion Eq. (5), the marginal probability of $\boldsymbol{z}_0$ is computed as a Gaussian mixture model (GMM):

$$p(\boldsymbol{z}_0) = \sum_i w_i \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{c}_i}, \boldsymbol{Q}_{\boldsymbol{c}_i}), \quad (6)$$

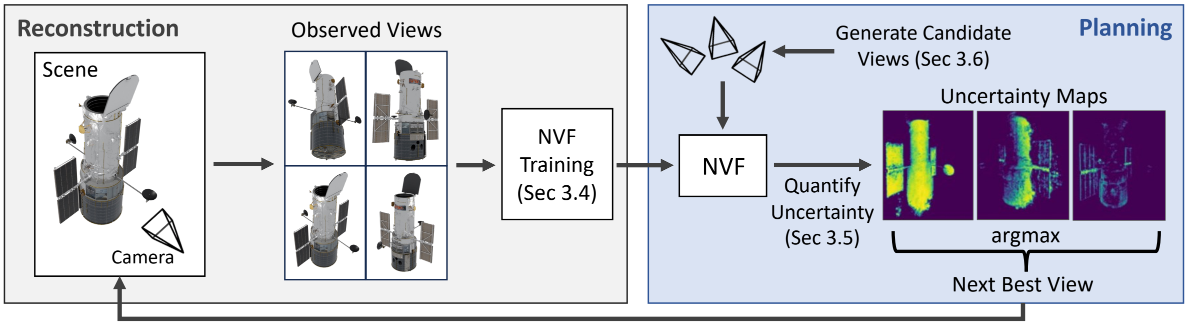

Figure 2. **Active Mapping with NVF.** Starting with a small set of initial views, a trained NVF is used to quantify uncertainties among sampled candidate views and chooses the view with maximum uncertainty as the next view to be observed by the agent.

with $w_i = \alpha_i \prod_{j=1}^{i-1}(1-\alpha_i)$. The distribution of the camera's observation, $p(\boldsymbol{z}_0)$, implies that $E[\boldsymbol{z}_0] = \sum_i w_i \boldsymbol{\mu}_{\boldsymbol{c}_i}$ which aligns with the original NeRF's volume rendering expression (Eq. (3)).

So far, we have developed a framework based on a probabilistic graphical model to bridge position-based uncertainty with ray-based observational uncertainty. In the following subsection, we will discuss the integration of the visibility factor into this framework.

3.3. Uncertainty with Visibility

With the Bayesian network formulation, we can now add visibility into the uncertainty estimation. Let a binary random variable V_i represent whether point i is visible to any camera in the training set. When a point is visible ($V_i = 1$), we can rely on NeRF's output for RGB and its variance. If a point is unobserved ($V_i = 0$), the NeRF's output at this point becomes unreliable, and we assign a prior color distribution $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{Q}_0)$ to it, as follows:

$$p(\boldsymbol{c}_i|V_i) = \begin{cases} \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{c}_i}, \boldsymbol{Q}_{\boldsymbol{c}_i}), & \text{if } V_i = 1, \\ \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{Q}_0), & \text{otherwise.} \end{cases} \tag{7}$$

Moreover, for invisible points, density prediction may also lack accuracy. Therefore, we define the conditional probability table for $P(D_i|V_i)$ as follows

$$\begin{array}{c|cc} & \multicolumn{2}{c}{\text{Occlusion } D_i} \\ \text{Visibility } V_i & 1 & 0 \\ \hline 1 & \exp(-\sigma_i s_i) & 1-\exp(-\sigma_i s_i) \\ 0 & \rho_i & 1-\rho_i \end{array} \tag{8}$$

where $\rho_i = (1-\beta)\exp(-\sigma_0 s_i) + \beta \exp(-\sigma_i s_i)$, and the hyperparameter β represents the accuracy of occlusion prediction, specifically indicating the likelihood that a prediction about occlusion is correct for points that are invisible. In situations where the occlusion prediction is incorrect, we resort to using a constant prior density σ_0 to estimate the occlusion probability. This approach helps in adjusting our model's predictions on density, particularly for points not visible to any camera in the training set.

By combining Eqs. (2), (7), (8), the marginal probability of $\boldsymbol{z}_i$ satisfies the recursive formula similar to Eq. (5): $p(\boldsymbol{z}_i) = \alpha_i^* \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{c}_i}, \boldsymbol{Q}_{\boldsymbol{c}_i}) + (1-\alpha_i^*)\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{Q}_0)$, where $\alpha_i^* = \big(v_i + (1-v_i)\beta\big)\big(1-\exp(-\sigma_i s_i)\big) + (1-\beta)(1-v_i)(1-\exp(-\sigma_0 s_i))$, and $v_i = P(V_i = 1)$ is the probability of point i being visible to at least one camera in the training set. The marginal probability of $p(\boldsymbol{z}_0)$ can be computed similarly as follows

$$p(\boldsymbol{z}_0) = \sum_i w_i^* v_i \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{c}_i}, \boldsymbol{Q}_{\boldsymbol{c}_i}) + \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{Q}_0) \sum_i w_i^*(1-v_i), \tag{9}$$

resulting in a GMM as well, where $w_i^* = \alpha_i^* \prod_{j=1}^{i-1}(1-\alpha_j^*)$.

3.4. Neural Visibility Field

So far, we have established a framework that bridges position-based uncertainty with ray-based observation uncertainty, while also incorporating visibility factors. Next, we discuss a method to determine visibility v_i, which is the probability that a point $\boldsymbol{x}_i$ is visible to at least one camera in the training set. Let $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots\}$ be the set of camera poses in the training set. If point $\boldsymbol{x}_i$ is within the field of view of a camera $\boldsymbol{p}$ in $\mathcal{P}$, the visibility of $\boldsymbol{x}_i$ to camera $\boldsymbol{p}$ can be expressed as $v_{\boldsymbol{p}}(\boldsymbol{x}_i) = T^p(t_i^p)$, where $\boldsymbol{x}_i = \boldsymbol{o}_{\boldsymbol{p}} + t_i^p \boldsymbol{d}_{\boldsymbol{p}}$ is on a ray from camera $\boldsymbol{p}$, and $T_p(t_i^p)$ denotes the probability of the ray being transmitted from $\boldsymbol{o}_{\boldsymbol{p}}$ to t_i^p without occlusion, as defined in Eq. (2). Therefore, the probability that point $\boldsymbol{x}_i$ is visible to at least one camera in the set $\mathcal{P}$ is given by:

$$v_{\mathcal{P}}(\boldsymbol{x}) = 1 - \prod_{\boldsymbol{p} \in \mathcal{P}} (1 - v_p(\boldsymbol{x}_i)). \tag{10}$$

However, directly computing Eq. (10) during volume rendering is impractical. For each point along a ray $\boldsymbol{r}$, it would require generating an additional ray r_p from camera $\boldsymbol{p}$ and sampling points along this ray to determine the point's visibility to camera $\boldsymbol{p}$. Doing this for all existing views is computationally expensive. To address this, we propose to amortize the cost by training an implicit model to predict the visibility.

We introduce Neural Visibility Field (NVF), an augmented NeRF that outputs both color uncertainty and visibility. The enhanced model is defined as $F_\Theta : (\boldsymbol{x}, \boldsymbol{d}) \rightarrow (\sigma, \boldsymbol{\mu_c}, \boldsymbol{Q_c}, v)$, where v represents the visibility with respect to the training views, and $\boldsymbol{\mu_c}$ and $\boldsymbol{Q_c}$ denote the mean and covariance of the color vector, respectively. The parameters $\boldsymbol{\mu_c}$ and σ are trained with Mean Square Error loss as in [30]. To train $\boldsymbol{Q_c}$, we employ the Negative Log-Likelihood Loss as follows:

$$\mathcal{L}_{cov} = -\sum_{r \in \mathcal{R}} \log \left(\sum_i w_i \mathcal{N}(C_g(\boldsymbol{r}); \boldsymbol{\mu_{c_i}}, \boldsymbol{Q_{c_i}}) \right), \quad (11)$$

where $\mathcal{R}$ denotes the set of rays in each batch, and $C_g(\boldsymbol{r})$ represents the ground truth color of ray $\boldsymbol{r}$. For training, we randomly sample points within the scene. The ground truth visibility, derived using Eq. (10), is then utilized to train the visibility head, using cross-entropy loss. Please refer to Supp. for further details on network architecture and training.

3.5. Active Mapping with NVF

In this section, we apply the PDF of ray color, derived from Sec. 3.4, for active mapping purposes. Let $Z_{\boldsymbol{p}}^{mn}$ be the color of the ray corresponding to pixel index m, n from camera pose $\boldsymbol{p}$. The PDF of $Z_{\boldsymbol{p}}^{mn}$, denoted as $p(\boldsymbol{z}_{\boldsymbol{p}}^{mn})$, can be obtained using the formulation provided in Eq. (9). We define $\boldsymbol{Z_p}$ as a random variable in $\mathbb{R}^{H \times W \times 3}$, representing the collective observation of all pixels in an image with height H and width W.

The goal of active mapping is to identify a camera pose, denoted as $\boldsymbol{p}^*$, that maximizes the entropy of the observation $\boldsymbol{Z}$ at that pose. This is formally expressed as:

$$\boldsymbol{p}^* = \arg\max_{\boldsymbol{p}} \mathcal{H}(\boldsymbol{Z_p}). \quad (12)$$

Note that we can deduce Eq. (12) from the information gain or mutual information $\mathcal{I}(\boldsymbol{Z_p}; \boldsymbol{M}) = \mathcal{H}(\boldsymbol{Z_p}) - \mathcal{H}(\boldsymbol{Z_p}|\boldsymbol{M})$, where $\boldsymbol{M}$ represents the random variable of the entire map. This assumes that $\mathcal{H}(\boldsymbol{Z_p}|\boldsymbol{M})$ is constant, specifically, that measurement noise remains constant given a known map.

To compute $\mathcal{H}(\boldsymbol{Z_p})$, we initially assume that the color of each pixel is independent of the others. Under this assumption, the entropy of $\boldsymbol{Z_p}$ can be calculated as $\mathcal{H}(\boldsymbol{Z_p}) = \sum_{m,n} \mathcal{H}(\boldsymbol{Z}_{\boldsymbol{p}}^{mn})$. However, this assumption of independence may not always hold true. For instance, when the camera is in close proximity to an object, the pixels in the image are often strongly correlated, particularly since they are measuring points that are spatially close. To account for this spatial correlation, we introduce a correction term:

$$\mathcal{H}(\boldsymbol{Z_p}) = \sum_{m,n} \left(\mathcal{H}(\boldsymbol{Z}_{\boldsymbol{p}}^{mn}) - f_{\text{corr}}(\mathcal{H}(\boldsymbol{Z}_{\boldsymbol{p}}^{mn}); d_{\boldsymbol{p}}^{mn}) \right) \quad (13)$$

Here, $f_{\text{corr}}(\mathcal{H}(\boldsymbol{Z}_{\boldsymbol{p}}^{mn}); d_{\boldsymbol{p}}^{mn})$ incorporates spatial correlation based on the expected depth $d_{\boldsymbol{p}}^{mn}$. Furthermore, we use the upper bound as proposed in [15] to closely approximate the entropy of the GMM $\mathcal{H}(\boldsymbol{Z}_{\boldsymbol{p}}^{mn})$, as it is known that there is no analytical solution for the entropy of GMM [15]. Further details on $f_{\text{corr}}(\mathcal{H}(\boldsymbol{Z}_{\boldsymbol{p}}^{mn}); d_{\boldsymbol{p}}^{mn})$ and entropy computation are included in supp material.

Within our theoretical framework for estimating uncertainty in NeRF and active mapping, all prior works, to our best knowledge, can be viewed as special cases. Specifically, if we drop the visibility factor, each prior work can be viewed as a specific approximation of our method. For instance, Lee et al [22] focuses only on the discrete random variable, computing the Shannon entropy with $-w_i \log w_i$, which can be regarded as a simplified version of ours, albeit excluding the differential entropy term Similarly, [34, 40] uses the weighted average of position-based color variance to approximate the rays-based observation variance, which lacks theoretical grounding and ignores the visibility factor in weight computation. In addition, Zhan et al [57] approximate the entropy of ray-based observation by directly summing the position-based entropy of occlusion, whereas, similarly, Yan et al [54] use the weighted average of position-based entropy of occlusion to approximate the ray-based observation entropy.

3.6. Active Mapping Pipeline

Here we briefly describe the active mapping pipeline using NVF (illustrated in Fig. 2). Please refer to Supp. material for more details. The process starts with training the NVF on a small batch of initial views. We employ two strategies for next view selection. In the sampling-based strategy, we sample N views from a prior distribution, estimate their uncertainty using Eq. (12), and select the view with maximum uncertainty. The gradient-based strategy is implemented by adjusting the selected pose through gradient-based optimization, aimed at maximizing entropy, leveraging the inherent differentiability of our uncertainty estimation method. Lastly, the agent proceeds to collect and integrate the new observations into the training views to re-train the NVF model and plan the next view.

4. Experiments

In this section, we seek to verify our hypothesis that (a) NVF outperforms existing methods in both uncertainty quantification and active mapping both quantitatively and qualitatively; and (b) the visibility term plays a vital role in this result.

Simulation Environments and Learning Setup. We conduct experiments on three datasets of varying difficulty levels for active mapping: all assets from the original NeRF dataset [31], the Hubble Space Telescope, and a custom synthetic indoor Room scene. In particular, the Room scene consists of two spaces divided by a wall. Successfully mapping the scene requires traversing both spaces. We assume access to a coarse bounding box that contains the region of

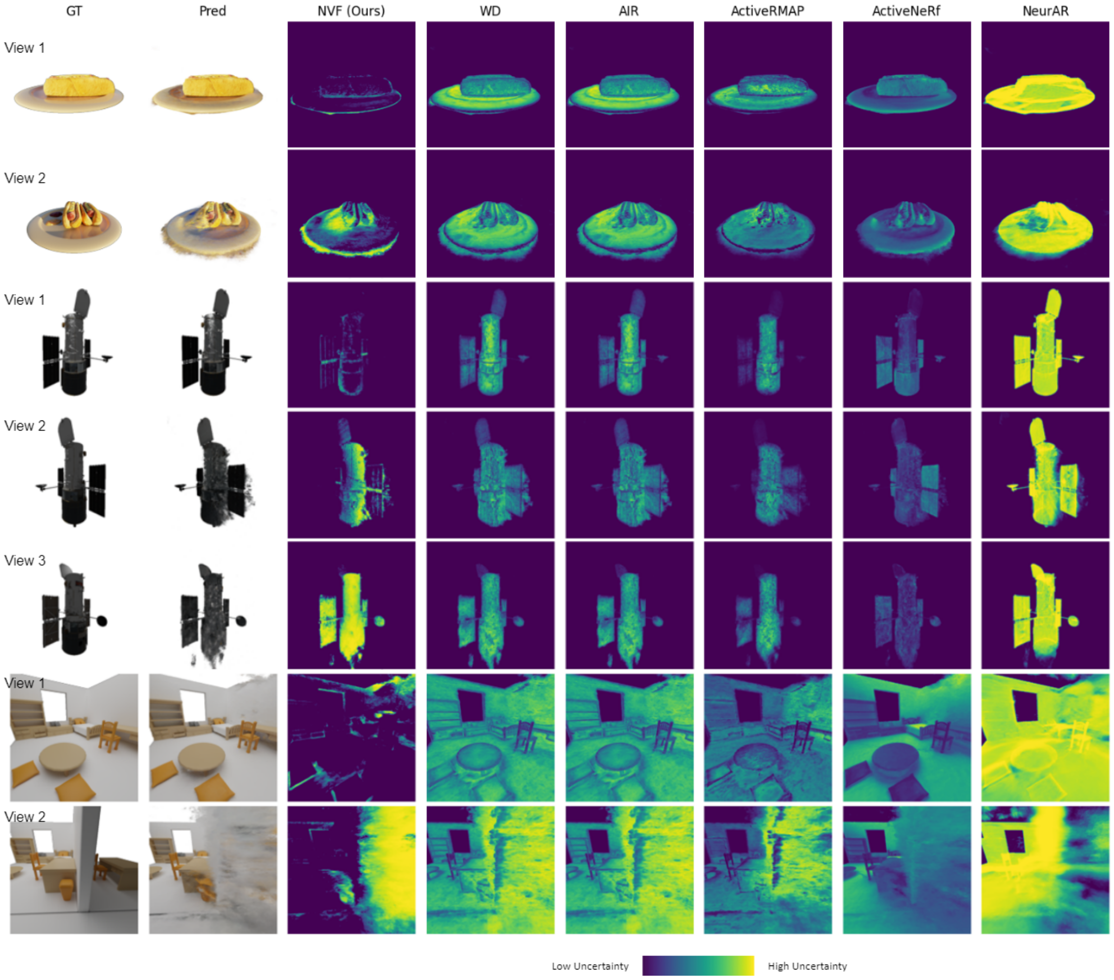

Figure 3. **Qualitative results of entropy estimation:** NVF assigns a higher entropy to previously unobserved regions while the baselines do not distinguish between the observed (View 1) and unobserved regions (View 2/3). Schematic illustrations of the poses of View 1, 2, and 3 can be found in supp. material. Note that within each method and scene, all rendered views share the same color bar.

interest. All ground truth images used for training NeRF and assessing reconstruction quality were rendered using Blender at a resolution of 512×512. We utilized Instant-NGP [32] as an efficient backbone for all uncertainty estimation methods. All NeRF models were trained for 5,000 iterations. For NVF, it first trains the Instant-NGP backbone, freezes its weights, and then trains variance and visibility heads, to ensure the performance improvements are attributed to better entropy estimation instead of a change in the loss function.

Baselines. We compared our method with state-of-the-art NeRF uncertainty quantification and active mapping methods. This includes the weight distribution-based entropy approximation (WD) [22]; occlusion-based entropy approximation (ActiveRMAP) [57]; weighted occlusion-based entropy approximation - ActiveImplicitRecon (AIR) [54], and spatial RGB variance-based uncertainty estimation ActiveNeRF [34] and NeurAR [40]. As discussed earlier, all of these works can be viewed as a special case of our method, while missing key aspects that NVF introduces. In addition, we include an agent that randomly selects views from the candidate poses (Random).

4.1. Uncertainty Estimation

Setup. We qualitatively compare the uncertainty (entropy) maps produced by our method and the baselines given a set of training views. For each scene, we design scenarios where only certain regions are visible in the training views. We then train all methods on the same training views and query for uncertainty estimation at an unseen test view. An effective uncertainty estimation method should be able to differentiate regions unobserved in the training set, as reconstruction in these areas is noisy and inaccurate. For the ”Hotdog” scene from the original NeRF dataset, we randomly sample 20 training views from a 90-degree sector above the plate. In the Hubble scene, we sample 20 training views from a 90-degree sector on one side, keeping the opposite side of the

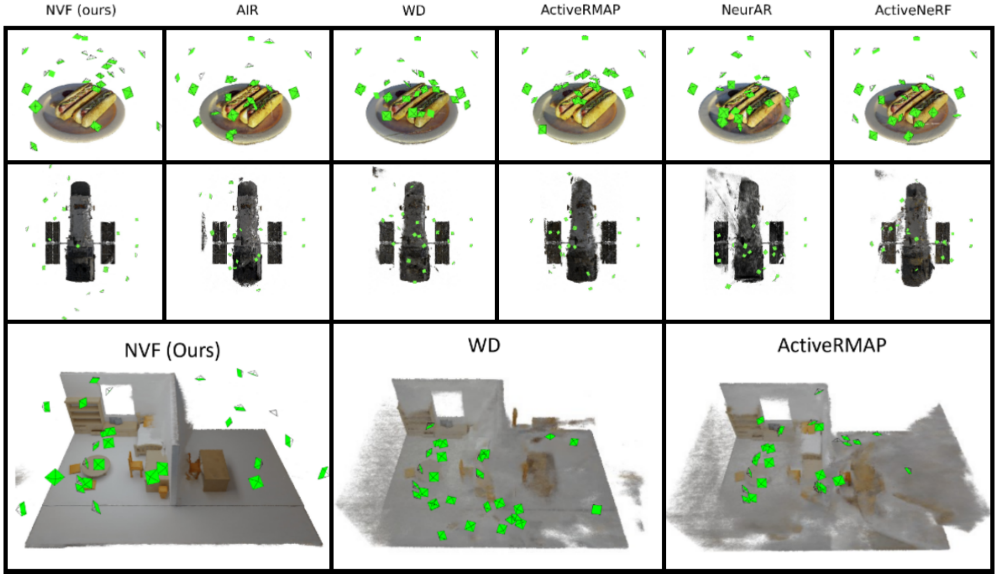

Figure 4. **Reconstruction results and camera view distribution:** NVF demonstrates superior reconstruction and scene coverage across all datasets in comparison to baselines. For room scene, only comparable baselines are presented, full results are provided in supp. material.

Hubble out of view. For the Room scene, we sample 30 training views oriented toward the back wall, the common wall, and the floor of one of the rooms, ensuring that the other room is unobserved.

Results. In the Hubble and Hotdog scenes, as illustrated in Fig. 3, all baseline methods fail to accurately capture the uncertainty in the unobserved areas of the scenes. Several baselines assign greater or similar uncertainty to the observed views as compared to the unobserved ones. A notable example is the Room scene. The first view focuses on the room observed in the training view, while the second view targets the common wall between the two rooms. The result indicates that our method differentiates between the uncertainties in regions seen in training views and unseen regions by modeling their visibilities. In Hubble and Hotdog, ActiveNeRF and NeurAR estimate a similar level of uncertainty for both unobserved and observed regions. However, in the Room scene, the uncertainty in the unobserved region is estimated to be lower than the observed. This shows that in complex scenarios, the uncertainty formulations of ActiveNeRF and NeurAR are ineffective, and such formulation alone is insufficient as guidance to explore unobserved regions.

4.2. Active Mapping

Setup. We deploy active mapping agents with the pipeline described in Sec. 3.6 with different uncertainty estimation methods. To ensure a fair comparison, all methods are evaluated under the same conditions during the comparison, to ensure that the planning is driven solely by the uncertainty estimation. Specifically, all candidate views are uniformly sampled within the space, without any prior constraints (such as the hemisphere constraint employed in ActiveNeRF). This approach ensures that a more accurate uncertainty estimation method will enable the robot to achieve more precise mapping results. For all original NeRF assets and Hubble scenes, we utilize 3-5 initial views covering only a portion of the scene, to realistically simulate active mapping scenarios. The Room scene presents the greatest challenge, with nine initial views sampled from one room, leaving the second room entirely unexplored. All agents start without knowledge of the second room's existence and are expected to discover it through uncertainty estimation and reconstruct the scene in 20 steps. Please refer to Supp. material for more details.

Evaluation metric. Our evaluation employs three types of metrics. For novel view synthesis quality, evaluations are performed at fixed testing viewpoints. We compare views synthesized by NeRF with ground truth renderings. The errors are quantified using Peak Signal-to-Noise Ratio (PSNR), Perceptual Image Patch Similarity (PIPS), Learned Perceptual Image Patch Similarity (LPIPS) [58], and RGB loss. For reconstructed mesh quality, we quantitatively evaluate the geometric accuracy of the scene reconstructions. We employ the metrics, Accuracy (Acc), Completion (Comp), and Completion Ratio (CR) as proposed in [46]. For visual coverage (Vis), we assess the proportion of faces in the ground truth mesh observed without occlusion during the experiments over all faces. The visibility of each face in the mesh is tracked using the ground truth mesh and a rasterizer.

Results. In Tab. 1, we show the quantitative results of our approach in comparison to other baselines. For the original NeRF Assets, we only include the average of the results across all scenes due to space limitation, detailed results are provided in supp. material. Our method significantly outperforms baseline methods achieving higher-quality re-

Table 1. Evaluation of Reconstructed Models Using Different Methods for Active Mapping

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	RGB↓	Acc.↓	Comp↓	CR↑	Vis↑
NeRF Assets (Avg.)	Random	17.63	0.766	0.264	0.0193	0.0426	0.0401	0.348	0.225
	WD	19.91	0.807	0.227	0.0121	0.0311	0.0204	0.479	0.466
	ActiveRMAP	20.03	0.807	0.219	0.0118	0.0292	0.0184	0.510	0.471
	AIR	19.86	0.807	0.230	0.0118	0.0290	0.0195	0.494	0.453
	ActiveNeRF	18.78	0.771	0.281	0.0157	0.0301	0.0238	0.433	0.415
	NeurAR	19.58	0.755	0.286	0.0134	0.0347	0.0251	0.452	0.424
	NVF (Ours)	**23.90**	**0.890**	**0.106**	**0.0045**	**0.0193**	**0.0111**	**0.685**	**0.532**
Hubble	Random	21.76	0.778	0.265	0.0113	0.0734	0.0262	0.329	0.291
	WD	24.15	0.855	0.184	0.0039	0.0297	0.0184	0.471	0.571
	ActiveRMAP	23.34	0.835	0.205	0.0048	0.0282	0.0162	0.465	0.570
	AIR	24.63	0.862	0.182	0.0035	0.0249	0.0140	0.525	0.586
	ActiveNeRF	23.33	0.824	0.250	0.0047	0.0355	0.0201	0.442	0.552
	NeurAR	25.19	0.772	0.265	0.0030	0.0480	0.0170	0.416	0.537
	NVF (Ours)	**27.99**	**0.919**	**0.100**	**0.0016**	**0.0225**	**0.0110**	**0.651**	**0.681**
Room	Random	12.95	0.800	0.378	0.0563	0.1837	0.5468	0.338	0.397
	WD	13.42	0.792	0.387	0.0533	0.2893	0.5415	0.317	0.428
	ActiveRMAP	13.91	0.786	0.412	0.0411	0.2233	0.4646	0.317	0.450
	AIR	15.19	0.829	0.386	0.0307	0.2710	0.3153	0.343	0.498
	ActiveNeRF	10.69	0.733	0.434	0.0853	0.1847	0.8181	0.292	0.338
	NeurAR	12.23	0.584	0.508	0.0599	0.3948	1.2647	0.178	0.375
	NVF (Ours)	**22.83**	**0.943**	**0.156**	**0.0053**	**0.1132**	**0.1997**	**0.464**	**0.586**

Table 2. Ablation Studies for Active Mapping with NVF

Ablations	PSNR↑	SSIM↑	LPIPS↓	Vis↑
w/o Vis.	21.11	0.844	0.187	0.382
w/o Var.	23.77	0.897	0.113	0.551
Ind. Rays	20.32	0.822	0.236	0.482
Loose	22.54	0.881	0.137	0.504
NVF (Ours)	24.42	0.902	0.108	0.546

construction and improved visual coverage. This is especially evident in challenging scenarios such as the Hubble and Room scenes, where our method successfully explores the entire scene and excels across all metrics. In contrast, baseline methods failed to fully explore these scenes, often revisiting previously explored areas (see Fig. 4) due to inadequate uncertainty estimation that overlooks visibility.

4.3. Ablation studies

We ablate key components in NVF to examine their role. First, we negate the visibility factor by presuming all sampled points as visible to the camera, setting the visibility head output to 1 for any input. Second, we disregard the spatial color variance estimation from NeRF, assuming a constant small uncertainty for all sampled points. Third, we omit the correlation correction factor, treating all rays as independent. Lastly, for entropy computation, we substitute the upper bound proposed by [15] with a looser bound, treating multiple Gaussians as a single Gaussian following [13]. The average results across all scenes are shown in Tab. 2, highlighting the crucial role in the visibility factor, removing it significantly drops the performance. We also observe that the correction of independence in Eq. (13) (”Ind. Rays.” in Tab. 2), and a tighter upper bound (”Loose”) positively impact performance. However, the position-based color uncertainty directly predicted by NeRF (”w/o Var.”) plays a less important role, underscoring visibility as the most critical factor in uncertainty estimation for active mapping.

5. Conclusion

In this work, we present Neural Visibility Field, a principled approach that accounts for visibility in uncertainty quantification and provide a unifying view of prior research in this direction. We empirically demonstrated that NVF significantly outperforms baselines in reconstruction quality and visual coverage across three scenes with varying levels of complexity. A limitation of our current active mapping pipeline is that it does not account for the constraints imposed on the planned trajectory of an agent. A possible future direction is to integrate NVF with cost-aware path planning.

Acknowledgments

This work is supported by NSF grant 2101250. We thank Mehregan Dor for the feedback on the preliminary version, and the anonymous reviewers for their comments and feedback on our manuscript.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022. 2

[2] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343–1377, 2011. 2

[3] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7911–7920, 2021. 2

[4] Cl Connolly. The determination of next best views. In *Proceedings. 1985 IEEE international conference on robotics and automation*, pages 432–435. IEEE, 1985. 2

[5] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999. 12

[6] Anna Dai, Sotiris Papatheodorou, Nils Funk, Dimos Tzoumanikas, and Stefan Leutenegger. Fast frontier-based information-driven autonomous exploration with an mav. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 9570–9576. IEEE, 2020. 2

[7] Tung Dang, Christos Papachristos, and Kostas Alexis. Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 2526–2533. IEEE, 2018. 2

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 2

[9] Christian Dornhege and Alexander Kleiner. A frontier-void-based approach for autonomous exploration in 3d. *Advanced Robotics*, 27(6):459–468, 2013. 2

[10] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14304–14314. IEEE Computer Society, 2021. 2

[11] Francis Engelmann, Jörg Stückler, and Bastian Leibe. Joint object pose estimation and shape reconstruction in urban street scenes using 3d shape priors. In *Pattern Recognition: 38th German Conference, GCPR 2016, Hannover, Germany, September 12-15, 2016, Proceedings 38*, pages 219–230. Springer, 2016. 2

[12] Georgios Georgakis, Bernadette Bucher, Anton Arapin, Karl Schmeckpeper, Nikolai Matni, and Kostas Daniilidis. Uncertainty-driven planner for exploration and navigation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11295–11302. IEEE, 2022. 2

[13] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, pages IV–317. IEEE, 2007. 8, 13

[14] Matthew D Hoffman, Tuan Anh Le, Pavel Sountsov, Christopher Suter, Ben Lee, Vikash K Mansinghka, and Rif A Saurous. Probnerf: Uncertainty-aware inference of 3d shapes from 2d images. In *International Conference on Artificial Intelligence and Statistics*, pages 10425–10444. PMLR, 2023. 2

[15] Marco F Huber, Tim Bailey, Hugh Durrant-Whyte, and Uwe D Hanebeck. On entropy approximation for gaussian mixture random vectors. In *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 181–188. IEEE, 2008. 5, 8, 13

[16] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484. IEEE, 2016. 2

[17] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 867–876, 2022. 2

[18] Liren Jin, Xieyuanli Chen, Julius Rückin, and Marija Popović. Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11305–11312. IEEE, 2023. 2

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12

[20] Georgios Kopanas and George Drettakis. Improving nerf quality by progressive camera placement for free-viewpoint navigation. 2023. 2

[21] Simon Kriegel, Christian Rink, Tim Bodenmüller, and Michael Suppa. Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, 10:611–631, 2015. 2

[22] Soomin Lee, Le Chen, Jiahao Wang, Alexander Liniger, Suryansh Kumar, and Fisher Yu. Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. *IEEE Robotics and Automation Letters*, 7(4):12070–12077, 2022. 2, 5, 6

[23] Kejie Li, Yansong Tang, Victor Adrian Prisacariu, and Philip HS Torr. Bnv-fusion: Dense 3d reconstruction using bi-level neural volume fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6166–6175, 2022. 2

[24] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 2

[25] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 6498–6508, 2021. 2

[26] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021. 2

[27] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2

[28] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2

[29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2

[30] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2, 5

[31] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 3, 5

[32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 6, 12

[33] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 2

[34] Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. In *European Conference on Computer Vision*, pages 230–246. Springer, 2022. 1, 2, 5, 6, 13

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2

[36] Sida Peng, Chen Geng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Implicit neural representations with structured latent codes for human body modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[37] Richard Pito. A solution to the next best view problem for automated surface acquisition. *IEEE Transactions on pattern analysis and machine intelligence*, 21(10):1016–1030, 1999. 2

[38] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[39] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 400–418. Springer, 2020. 2

[40] Yunlong Ran, Jing Zeng, Shibo He, Jiming Chen, Lincheng Li, Yingfeng Chen, Gimhee Lee, and Qi Ye. Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters*, 8(2):1125–1132, 2023. 2, 5, 6, 13, 18

[41] Korbinian Schmid, Heiko Hirschmüller, Andreas Dömel, Iris Grixa, Michael Suppa, and Gerd Hirzinger. View planning for multi-view stereo 3d reconstruction using an autonomous multicopter. *Journal of Intelligent & Robotic Systems*, 65: 309–323, 2012. 2

[42] William R Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys (CSUR)*, 35(1):64–96, 2003. 2

[43] James P Sethna. *Statistical mechanics: entropy, order parameters, and complexity*. Oxford University Press, USA, 2021. 13

[44] Jianxiong Shen, Adria Ruiz, Antonio Agudo, and Francesc Moreno-Noguer. Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. In *2021 International Conference on 3D Vision (3DV)*, pages 972–981. IEEE, 2021. 2

[45] Edward J Smith, Michal Drozdzal, Derek Nowrouzezahrai, David Meger, and Adriana Romero-Soriano. Uncertainty-driven active vision for implicit scene reconstruction. *arXiv preprint arXiv:2210.00978*, 2022. 2

[46] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021. 7

[47] Niko Sünderhauf, Jad Abou-Chakra, and Dimity Miller. Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9370–9376. IEEE, 2023. 2

[48] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 12, 13

[49] Mikaela Angelina Uy, Kiyohiro Nakayama, Guandao Yang, Rahul Krishna Thomas, Leonidas Guibas, and Ke Li. Nerf revisited: Fixing quadrature instability in volume rendering. *arXiv preprint arXiv:2310.20685*, 2023. 2

[50] Vlatko Vedral. The role of relative entropy in quantum information theory. *Reviews of Modern Physics*, 74(1):197, 2002. 13

[51] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2

[52] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021. 2

[53] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, pages 641–676. Wiley Online Library, 2022. 2

[54] Dongyu Yan, Jianheng Liu, Fengyu Quan, Haoyao Chen, and Mengmeng Fu. Active implicit object reconstruction using uncertainty-guided next-best-view optimization. *IEEE Robotics and Automation Letters*, 2023. 1, 2, 5, 6, 18

[55] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8254–8263, 2023. 2

[56] Javier Yu, Jun En Low, Keiko Nagami, and Mac Schwager. Nerfbridge: Bringing real-time, online neural radiance field training to robotics. *arXiv preprint arXiv:2305.09761*, 2023. 13

[57] Huangying Zhan, Jiyang Zheng, Yi Xu, Ian Reid, and Hamid Rezatofighi. Activermap: Radiance field for active mapping and planning. *arXiv preprint arXiv:2211.12656*, 2022. 1, 2, 5, 6

[58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

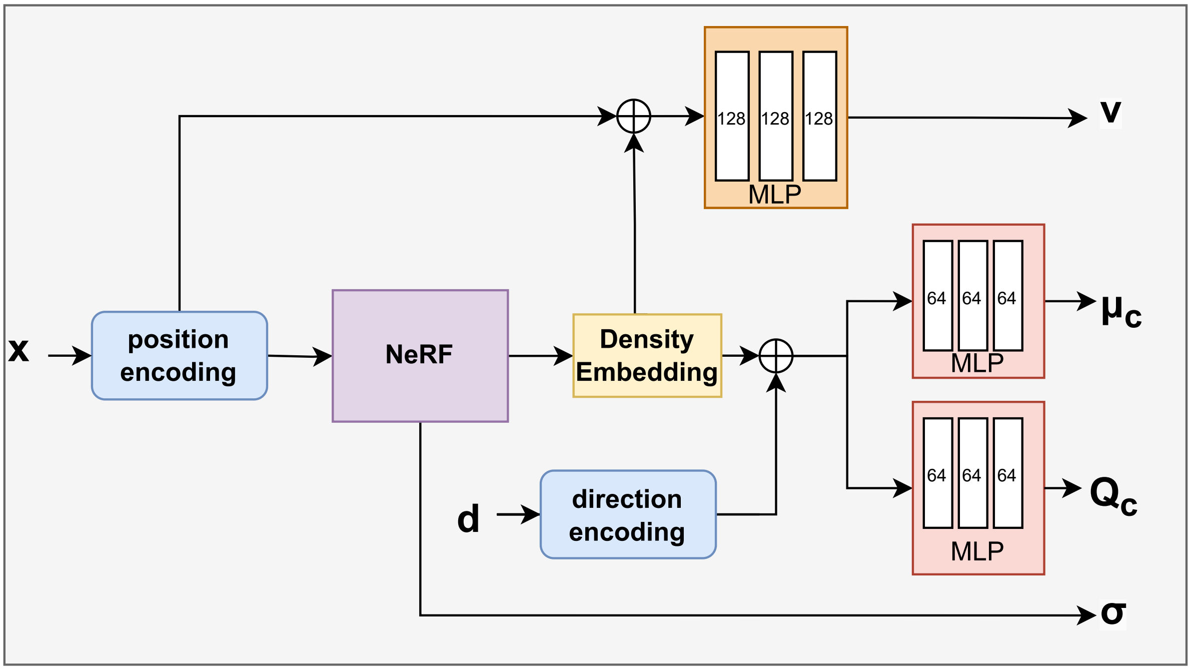

Figure 5. **NVF Architecture:** The MLP block consists of fully connected layers that use the ReLU activation function. The numbers inside the block denote the size of the layer. The final output from the visibility (v) MLP and RGB (μ_c) MLP are passed through the sigmoid activation function while the RGB Variance (Q_c) MLP uses softplus activation

A. Method Details

A.1. NVF architecture and training details

NVF is an augmentation of a NeRF consisting of two additional MLP heads for predicting RGB variance and visibility. Specifically, we implement NVF on top of a nerfstudio [48] implementation of Instant-NGP [32], where the color MLP head represents $\boldsymbol{\mu}_c$. Alongside the color head is a MLP head for RGB variance, outputting a 3x1 vector $\boldsymbol{Q}_c$. Similarly, the visibility MLP head is attached alongside the density head. For a visualization of the architecture, see Appendix Fig. 5. In practice, we train Instant-NGP, variance, and visibility separately and in sequence. First, we train the NeRF backbone for 5000 iterations using a learning rate of 0.01 and 4096 rays per batch. Next, the variance head is trained for 500 iterations using a learning rate of 0.001 and 4096 rays per batch. Finally, the visibility head is trained for 500 iterations using a learning rate of 0.001 and 65536 samples per batch. We train all modules using the Adam optimizer [19].

A.2. Entropy computation details

Joint Entropy of the Camera Observation. We discuss the details on the computation of the joint entropy $\mathcal{H}(\boldsymbol{Z_p})$ as formulated in Eq. (13). For simplicity in this discussion, we denote the joint entropy as $\mathcal{H}(\boldsymbol{Z})$ in this section. We model the joint observation of all rays as a Bayesian network, where the observation of each pixel only depends on its adjacent neighboring pixel. Consequently, the joint probability can be factorized as $p(z) = \prod_{mn} p(z_{mn}|z_{m+1,n}, z_{m,n+1})$. Note that for the sake of brevity, boundary terms where a pixel lies at the image edge are omitted here. Then, by applying the chain rule of entropy[5], we obtain:

$$\mathcal{H}(\boldsymbol{Z}) = \sum_{m,n} \mathcal{H}(\boldsymbol{Z}_{mn}|\boldsymbol{Z}_{m+1,n}, \boldsymbol{Z}_{m,n+1}) \tag{14}$$

We then apply the inequality $\mathcal{H}(\boldsymbol{Z}_{mn}|\boldsymbol{Z}_{m+1,n}, \boldsymbol{Z}_{m,n+1}) \leq \mathcal{H}(\boldsymbol{Z}_{mn}|\boldsymbol{Z}_{m+1,n})$ and $\mathcal{H}(\boldsymbol{Z}_{mn}|\boldsymbol{Z}_{m+1,n}, \boldsymbol{Z}_{m,n+1}) \leq \mathcal{H}(\boldsymbol{Z}_{mn}|\boldsymbol{Z}_{m,n+1})$. These allow us to derive an upper bound for $\mathcal{H}(\boldsymbol{Z})$:

$$\mathcal{H}(\boldsymbol{Z}) \leq \sum_{m,n} \frac{1}{2}\big(\mathcal{H}(\boldsymbol{Z}_{m,n}|\boldsymbol{Z}_{m+1,n}) + \mathcal{H}(\boldsymbol{Z}_{m,n}|\boldsymbol{Z}_{m,n+1})\big) \tag{15}$$

Then we connect the conditional entropy with the correlation between the two adjacent rays. We let

$$\rho_{m+1,n} = 1 - \frac{\mathcal{H}(\boldsymbol{Z}_{m,n}|\boldsymbol{Z}_{m+1,n})}{\mathcal{H}(\boldsymbol{Z}_{m+1,n})} \tag{16}$$

where ρ is as a measure of correlation. Specifically, a ρ value closer to 1 indicates $\boldsymbol{Z}_{m,n}$ and $\boldsymbol{Z}_{m+1,n}$ are strongly correlated, whereas a ρ tends to 0 suggests they are not correlated. It is important to note that this definition of correlation, based on entropy, differs from the widely recognized

Pearson correlation coefficient, and is commonly used in quantum information[50]. Consequently, we can obtain the upper bound:

$$\mathcal{H}(\boldsymbol{Z}) \leq \sum_{m,n}(1-\rho_{mn})\mathcal{H}(\boldsymbol{Z}_{mn}) \tag{17}$$

We assume that the correlation between two points in the scene can be modeled as a function of their spatial distance, a concept commonly referred to as the correlation function in statistical physics [43]. We adopt a truncated least-square form for this correlation function.

$$\rho(x) = \begin{cases} 1-(\frac{x}{\xi})^2 \text{ , if } x<\xi, \\ 0 \text{ , otherwise} \end{cases} \tag{18}$$

This formula indicates that two points located within a distance threshold ξ of each other are strongly correlated, whereas those beyond this threshold are considered independent. It is noteworthy that this term bears resemblance to the correlation function $\rho(x) = \exp(-\frac{x}{\xi})$, which is commonly applied in statistical physics [43], and ξ represents the correlation length. Empirical evaluations indicate that the use of either correlation function expression significantly outperforms the scenario where all rays are assumed to be independent ($\rho = 0$). Notably, a marginal improvement was observed when utilizing Eq. (18).

Therefore, we can approximate the correlation between two adjacent rays based on their expected depth, expressed as $\rho_{mn} = \rho(d_{mn}\Delta\phi)$, where $\Delta\phi$ is the angular resolution of each pixel, d_{mn} is the expected depth of ray at pixel (m, n). This implies that when the camera is closer to an object, the observations in adjacent pixels of the camera exhibit stronger correlation. Hence the actual total information gain is smaller than the sum of the information gain of each pixel. Accordingly, the correction function f_{corr} in Eq. (13) can be defined as:

$$f_{corr}(\mathcal{H}(\boldsymbol{Z}_{mn}); d_{mn}) = \rho(d_{mn}\Delta\phi)\mathcal{H}(\boldsymbol{Z}_{mn}) \tag{19}$$

In our experiments, we let the correlation length $\xi = kD\Delta\phi$, where D represents the diameter of the coarse bounding box enclosing the object, and k is a hyperparameter, and we let $k = 0.25$.

Entropy of GMM. We then introduce the details to compute the entropy for each ray, which is modeled as a Gaussian Mixture Model (GMM). For the sake of simplicity, we denote the GMM's distribution as $p(\boldsymbol{x}) = \sum_i w_i \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{Q}_i)$. We use the upper bound proposed in [15] to closely approximate the entropy of the GMM $\mathcal{H}(\boldsymbol{X})$:

$$\mathcal{H}(\boldsymbol{X}) \leq \sum_i w_i \left(-\log w_i + \frac{1}{2}\log\left((2\pi e)^N |\boldsymbol{Q}_i|\right)\right) \tag{20}$$

This upper bound is expected to provide a more accurate approximation of the true entropy of the GMM compared to the conventional method which approximates the entropy using a single Gaussian that matches the first two moments of the GMM[13], given by $\mathcal{H}(\boldsymbol{X}) \leq \frac{1}{2}\log\left((2\pi e)^N|\boldsymbol{\Sigma}|\right)$ where $\boldsymbol{\Sigma}$ is calculated as:

$$\boldsymbol{\Sigma} = \sum_i w_i \left(\boldsymbol{Q}_i + (\boldsymbol{\mu_i} - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu_i} - \bar{\boldsymbol{\mu}})^T\right) \tag{21}$$

and $\bar{\boldsymbol{\mu}}$ is the weighted mean of the Gaussian components, defined as $\bar{\boldsymbol{\mu}} = \sum_i w_i \boldsymbol{\mu_i}$. It is worth mentioning that the baseline method [34, 40] use the weighted average of position-based color variance to approximate the rays-based observation variance by employing a single Gaussian whose mean and variance are the weighted averages of the means and variances of all samples along the rays, respectively ; in other words, $\boldsymbol{\Sigma} = \sum_i w_i \boldsymbol{Q}_i$. This approach resembles the first term in Eq. (21) but misses the covariance term $(\boldsymbol{\mu_i} - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu_i} - \bar{\boldsymbol{\mu}})^T$. Additionally, it does not take into account the visibility to the training views.

In summary, we derive an upper bound for the pixel-wise entropy, and consequently, for the joint entropy of each view, this upper bound is utilized to closely approximate the information gain at a given pose. In the planning phase, given a candidate pose, we first apply Appendix Eq. (20) to compute the entropy for each ray, subsequently, we compute the joint entropy of the image observation as per Eq. (13) at that pose, which then serves as a reward function in the planning process.

A.3. Active mapping implementation details

Active Mapping Pipeline. To train NVF within an active mapping framework, we build our pipeline on top of nerfstudio [48] and NerfBridge [56]. Every time a new view is added to NVF, the model is trained from scratch on the collection of its observed views.

After training, we sample candidate poses in the scene, without collision with the object, by filtering all poses within a density threshold. In the Room scene, the sampler additionally thresholds for collisions between view poses and the current pose, to make sure the agent could move to the new pose without collision. After candidate view poses are generated, NVF computes the entropy of each pose. The view with the highest entropy is next rendered in the scene and added to the observations. This procedure repeats until the horizon step is met, as is shown in Alg. 1. In the experiments, we sample $N = 512$ candidate views and run the active mapping for 20 steps; the evaluations are performed after the last planning step.

Gradient-based Optimization for Planning. In addition to the method of finding the best view among a randomly sampled candidate poses set, we also performed experiments on 6 DoF pose-refinement on the camera poses, $\boldsymbol{p} \in SE(3)$,

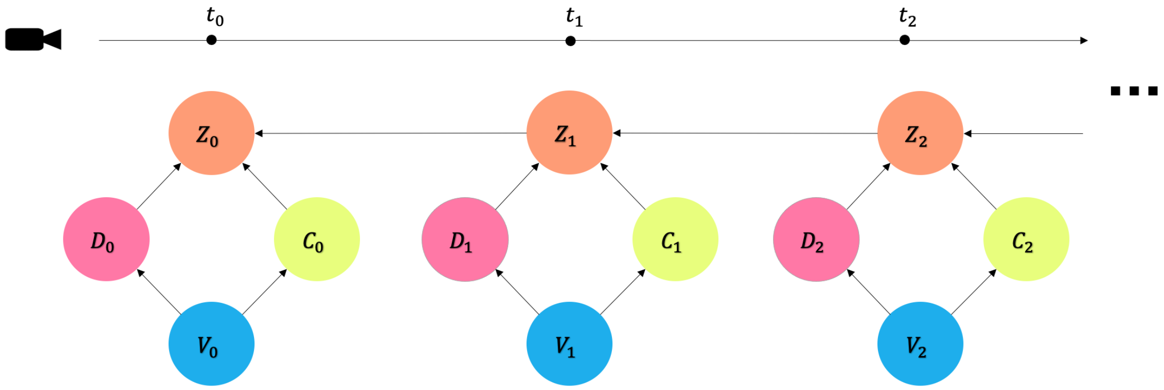

Figure 6. Schematics of the proposed Bayesian Network: Z represents the observed (ray-based) color, C represents the emitted (position-based) color, D represents if the interval is occluded, V represents the visibility,

.

as the entropy function $\mathcal{H}$ is a fully differentiable differentiable function of $\boldsymbol{p}$. We find the optimal $\boldsymbol{p}$ such that

$$\boldsymbol{p}^* = \underset{\boldsymbol{p} \in SE(3)}{\arg\max}\, \mathcal{H}(\boldsymbol{Z_p}) \tag{22}$$

We first find the top k poses with the highest entropy $\mathcal{P}_k$ and perform gradient-based optimization to refine the poses. To reduce the size of the computation graph and the memory requirements, a subset of pixels $\boldsymbol{Z}_i \subset \boldsymbol{Z_p}$ with an image is used to estimate the expected entropy, instead of the full image. We perform backpropogation on this estimated entropy using an Adam optimizer with a learning rate of $1e-4$, to find the optimum pose.

Algorithm 1 Active Mapping with NVF

```
Input:
    $\mathcal{P} \leftarrow$ initial poses
    $\boldsymbol{Z} \leftarrow$ initial images
for $i = 1$ to $n_{horizon}$ do
    $F_\Theta \leftarrow$ trainNVF($\mathcal{P}, \boldsymbol{Z}$)                ▷ train NVF
    $\mathcal{P}_c \leftarrow$ samplePoses($F_\Theta$)  ▷ sample candidate poses
    $\boldsymbol{p}_i \leftarrow \arg\max_{\boldsymbol{p} \in \mathcal{P}_c} \mathcal{H}(\boldsymbol{Z_p} | F_\Theta)$
    $\mathcal{P} \leftarrow \{\boldsymbol{p}_i\} \cup \mathcal{P}$
    $\boldsymbol{Z} \leftarrow$ takeImageAt($\{\boldsymbol{p}_i\}$) $\cup\, \boldsymbol{Z}$ ▷ update training set
return $F_\Theta$
```

B. Experiments Details

B.1. Uncertainty Estimation details

As for the entropy comparison experiments shown in Fig. 3 of the main paper, Appendix Fig. 7 provides an illustration of the pose of the training views and evaluation views.

Algorithm 2 Gradient-Based Optimization for Planning

```
Input:
    $\mathcal{P} \leftarrow$ sampled poses
$\mathcal{P}_k \leftarrow$ getTopKPoses($\mathcal{P}, \mathcal{H}$)
for $i = 1$ to $n_{iterations}$ do
    for $\boldsymbol{p}$ in $\mathcal{P}_k$ do
        $\boldsymbol{Z_p} \leftarrow$sampleRays($\boldsymbol{p}$)
        $\boldsymbol{p} \leftarrow \boldsymbol{p} + \eta \frac{\partial(\mathcal{H}(\boldsymbol{Z_p}|F_\Theta))}{\partial \boldsymbol{p}}$
$\tilde{\boldsymbol{p}} = \arg\max_{\boldsymbol{p} \in \mathcal{P}_k} \mathcal{H}(\boldsymbol{Z_p} | F_\Theta)$
return $\tilde{\boldsymbol{p}}$
```

B.2. Mesh metrics implementation details

For computing Accuracy, Completion, and Completion Ratio metrics, ground truth points are sampled from the ground truth scene meshes. Points from NVF's reconstructed mesh are sampled from the observation view rays. Accuracy measures the mean distance of sampled points from the reconstructed mesh to the nearest corresponding points in the ground truth mesh. Completion instead measures the mean distance of sampled ground truth points to the nearest reconstructed mesh points. Completion Ratio calculates the percentage of completion distances being below a threshold. For the original NeRF assets and Hubble scene, the threshold is set to 0.01. For the Room scene, as the scale is larger, the threshold is set to 0.1.

Visual coverage quantifies the surface area a trajectory of views covers a scene. We compute this with rasterization. Given a ground truth mesh of the scene, we project the mesh onto all of the observation views. In each rendered image, we record the number of mesh faces visible to the corresponding view. We append all observed faces to a visible set.

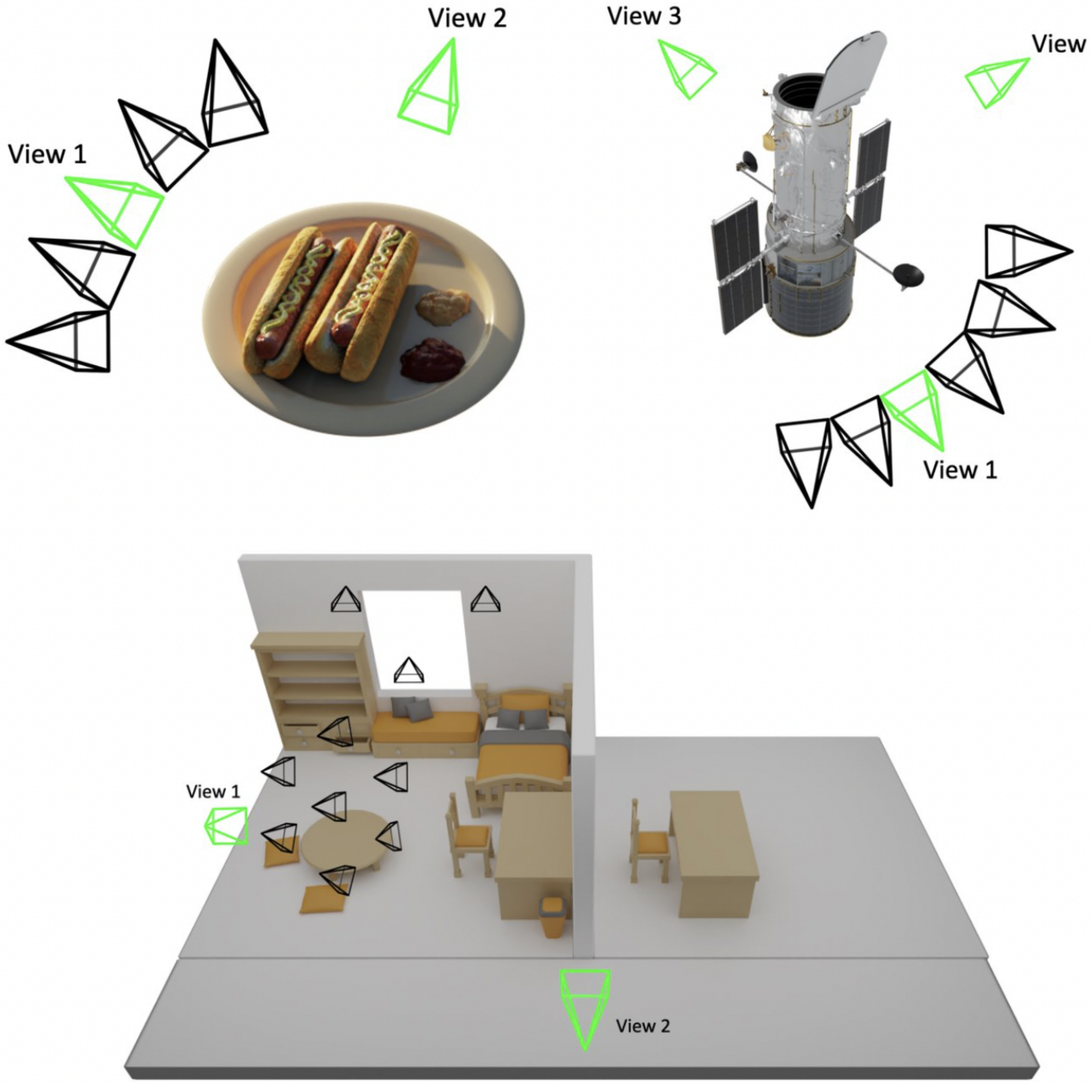

Figure 7. **Uncertainty Experiment Scene Setups:** Illustration of the training views and evaluation views in Fig. 3. The black frustums correspond to the training views, the green frustums are the evaluation views. For more video results, please refer to `https://sites.google.com/view/nvf-cvpr24/`

.

Computing visual coverage is then the ratio of the length of the visible set to the total number of faces in the mesh.

C. More Qualitative Results

C.1. Active Mapping

In addition to the results in Tab. 1, more qualitative results are presented in Appendix Fig. 8. As shown, our method achieves better novel view synthesis quality compared to baseline methods.

C.2. Gradient-based Pose-Optimization results

Certain methods compare uncertainty among a finite set of pre-defined scene-specific view candidates. This limits their applicability to previously unseen scenes as well as their ability to reach an optimal solution. Gradient-based pose estimation aims to find the next-best-view (NBV) on a continuous manifold which broadens its applicability to different scenarios and results in optimal view selection.

The results in Tab. 1 highlight our approach's ability to select the optimal view from proposed candidates, intentionally omitting gradient-based optimization to ensure a fair comparison. To extend our analysis, we conducted a further

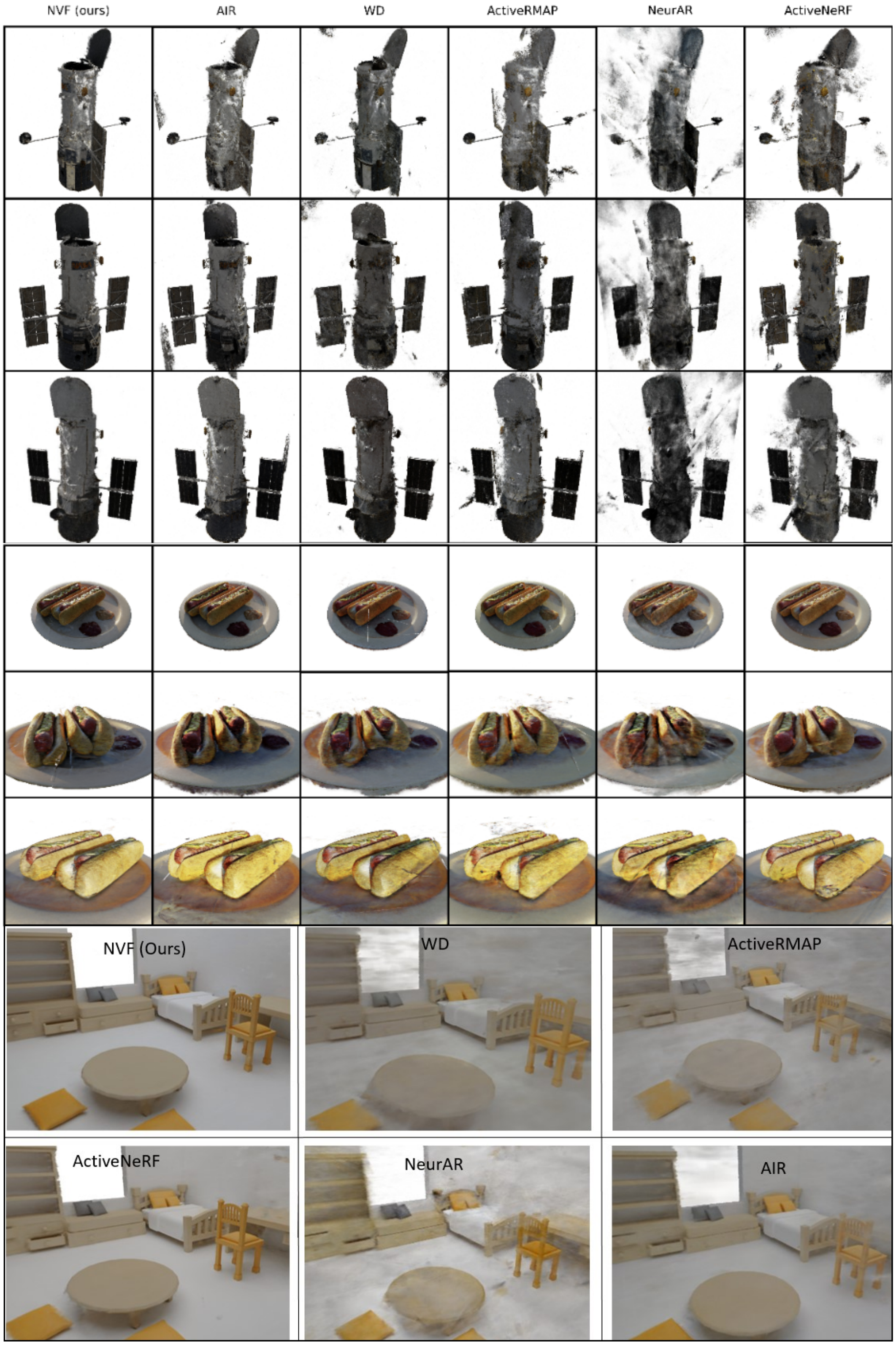

Figure 8. **Qualitative Results:** Comparisons on novel view synthesis results. Our method demonstrates superior novel-view synthesis rendering fine details in comparison to all baselines. For more video results, please refer to https://sites.google.com/view/nvf-cvpr24/

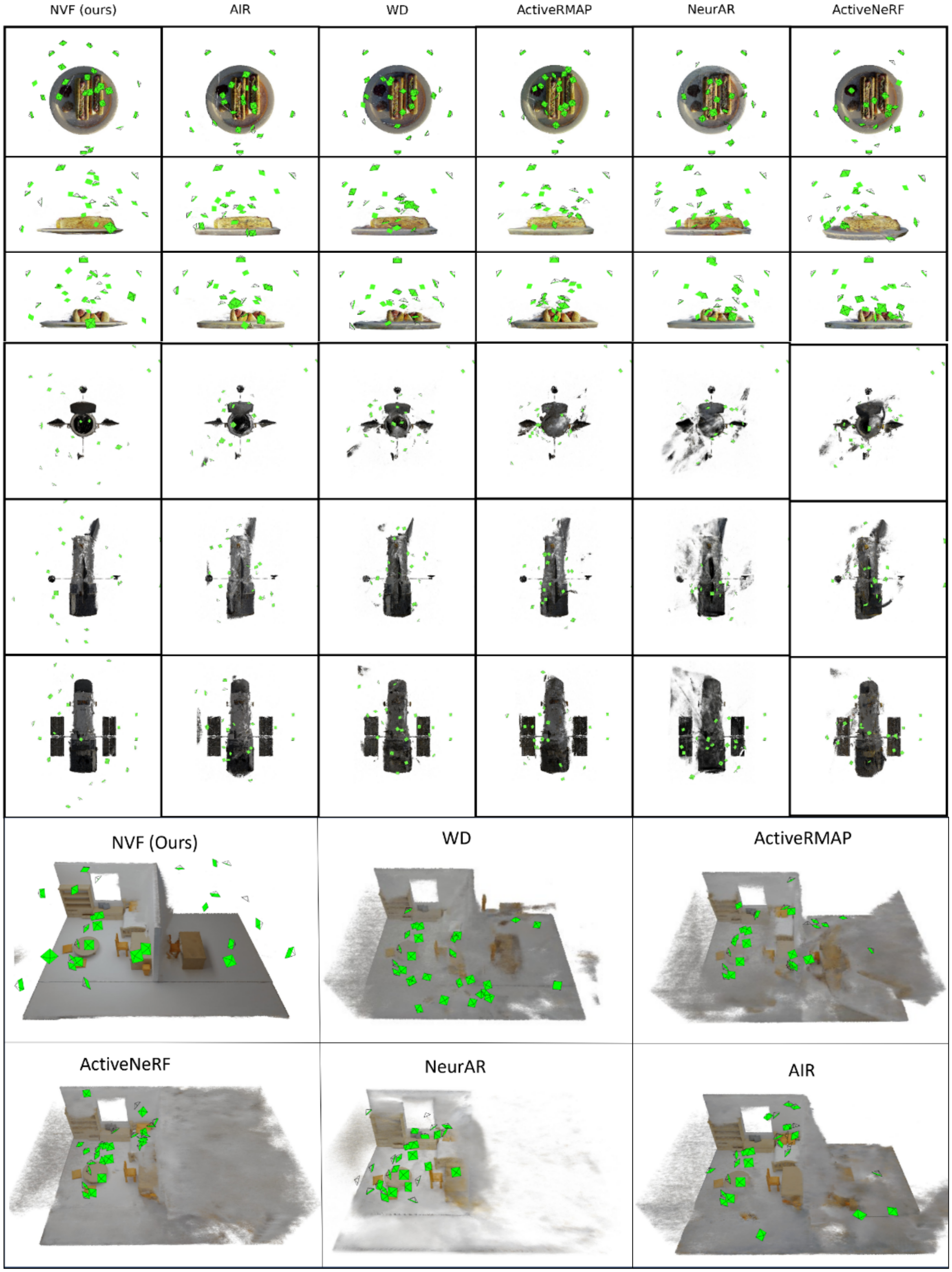

Figure 9. **Additional reconstruction results and camera view distribution** For more video results, please refer to https://sites.google.com/view/nvf-cvpr24/

.

Table 3. Performance of gradient-based methods

Method	PSNR↑	SSIM↑	LPIPS↓	RGB↓	Acc.↓	Comp.↓	C.R.↑	Vis.↑
AIR	24.63	0.862	0.182	0.0035	0.0249	0.0140	0.525	0.586
NeurAR	25.19	0.772	0.265	0.0030	0.0480	0.0170	0.416	0.537
NVF	27.99	0.919	0.100	0.0016	0.0225	0.0110	0.651	0.681
AIR-OPT	24.41	0.858	0.183	0.0037	0.0267	0.0159	0.450	0.548
NeurAR-OPT	25.42	0.794	0.245	0.0029	0.0461	0.0180	0.381	0.563
NVF-OPT	**29.33**	**0.930**	**0.086**	**0.0012**	**0.0196**	**0.0106**	**0.666**	**0.690**

Table 4. Ablation Studies

Ablations	PSNR↑	SSIM↑	LPIPS↓	RGB↓	Acc.↓	Comp.↓	C.R.↑	Vis.↑
w/o Vis.	21.11	0.844	0.187	0.0119	0.0466	0.0765	0.479	0.382
w/o Var.	23.77	0.897	0.113	0.0049	0.0276	0.0305	0.639	0.551
Ind. Rays	20.32	0.822	0.236	0.0125	0.0560	0.0506	0.451	0.482
Loose	22.54	0.881	0.137	0.0100	0.0247	0.0609	0.600	0.504
NVF (Ours)	24.42	0.902	0.108	0.0041	0.0287	0.0324	0.628	0.546

comparison with gradient-based optimization methods for view selection, detailed in Tab. 3. This comparison, which includes our method and two others [40, 54], utilizes gradient descent to refine the selection of views. As demonstrated in Appendix Tab. 3, the integration of gradient-based optimization considerably improves our method's performance, allowing it to surpass competing gradient-based approaches. This superior performance is attributed to our method's more precise estimation of uncertainty.

C.3. Additional Results

We present the complete results of all original NeRF assets in Appendix Tab. 5 & 6. We also present the complete results of the ablation study in Appendix Tab. 4. The result is averaged across all scenes.

Table 5. Results of original NeRF assets (1)

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	RGB↓	Acc.↓	Comp.↓	C.R.↑	Vis.↑
Chair	Random	17.17	0.835	0.190	0.0193	0.0470	0.0470	0.250	0.311
	WD	18.07	0.853	0.197	0.0163	0.0386	0.0167	0.499	0.582
	ActiveRMAP	18.67	0.863	0.183	0.0136	0.0277	0.0144	0.584	0.614
	AIR	18.47	0.859	0.176	0.0155	0.0296	0.0135	0.568	0.614
	ActiveNeRF	15.90	0.806	0.257	0.0280	0.0295	0.0223	0.407	0.503
	NeurAR	19.24	0.817	0.231	0.0127	0.0427	0.0155	0.485	0.596
	NVF (Ours)	**23.89**	**0.937**	**0.057**	**0.0041**	**0.0209**	**0.0089**	**0.763**	**0.705**
Drums	Random	17.08	0.753	0.286	0.0198	0.0378	0.0162	0.518	0.193
	WD	19.07	0.796	0.252	0.0126	0.0288	0.0130	0.575	0.444
	ActiveRMAP	18.77	0.784	0.264	0.0134	0.0385	0.0128	0.574	0.443
	AIR	19.00	0.789	0.277	0.0126	0.0319	0.0115	0.596	0.464
	ActiveNeRF	18.35	0.767	0.305	0.0147	0.0325	0.0160	0.479	0.393
	NeurAR	18.22	0.722	0.328	0.0151	0.0434	0.0158	0.453	0.401
	NVF (Ours)	**21.00**	**0.866**	**0.142**	**0.0079**	**0.0186**	**0.0069**	**0.836**	**0.541**
Ficus	Random	19.86	0.826	0.202	0.0103	0.0254	0.0141	0.671	0.355
	WD	17.98	0.777	0.316	0.0163	0.0299	0.0172	0.553	0.601
	ActiveRMAP	19.40	0.803	0.263	0.0122	0.0260	0.0122	0.653	0.637
	AIR	18.75	0.772	0.325	0.0134	0.0237	0.0145	0.575	0.554
	ActiveNeRF	18.75	0.762	0.366	0.0134	0.0210	0.0202	0.560	0.529
	NeurAR	20.27	0.755	0.337	0.0094	0.0254	0.0189	0.545	0.513
	NVF (Ours)	**22.76**	**0.900**	**0.089**	**0.0053**	**0.0112**	**0.0062**	**0.896**	**0.649**
Hotdog	Random	19.87	0.861	0.166	0.0107	0.0379	0.0565	0.239	0.361
	WD	21.84	0.892	0.131	0.0066	0.0186	0.0395	0.344	0.455
	ActiveRMAP	22.75	0.895	0.130	0.0053	0.0197	0.0415	0.338	0.466
	AIR	22.35	0.897	0.124	0.0058	0.0197	0.0381	0.351	0.470
	ActiveNeRF	21.57	0.885	0.145	0.0070	0.0234	0.0335	0.324	0.461
	NeurAR	22.90	0.866	0.171	0.0051	0.0279	**0.0320**	0.317	0.450
	NVF (Ours)	**26.10**	**0.928**	**0.084**	**0.0025**	**0.0157**	0.0356	**0.371**	**0.472**

Table 6. Results of original NeRF assets (2)

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	RGB↓	Acc.↓	Comp.↓	C.R.↑	Vis.↑
Lego	Random	16.49	0.720	0.265	0.0229	0.0599	0.0504	0.161	0.115
	WD	18.54	0.771	0.217	0.0142	0.0305	0.0283	0.257	0.224
	ActiveRMAP	17.49	0.752	0.234	0.0180	0.0238	0.0237	0.280	0.227
	AIR	19.33	0.797	0.189	0.0118	0.0262	0.0249	0.296	0.230
	ActiveNeRF	17.59	0.736	0.263	0.0176	0.0265	0.0317	0.222	0.199
	NeurAR	15.12	0.713	0.277	0.0314	0.0246	0.0357	0.319	0.189
	NVF (Ours)	**23.97**	**0.896**	**0.082**	**0.0040**	**0.0131**	**0.0167**	**0.426**	**0.270**
Materials	Random	15.90	0.802	0.220	0.0266	0.0409	0.0800	0.117	0.089
	WD	19.38	0.845	0.174	0.0122	0.0197	0.0275	0.343	0.304
	ActiveRMAP	19.68	0.843	0.174	0.0117	0.0213	0.0271	0.345	0.303
	AIR	19.45	0.844	0.171	0.0138	0.0238	0.0320	0.318	0.289
	ActiveNeRF	18.73	0.833	0.191	0.0135	0.0207	0.0290	0.322	0.287
	NeurAR	19.68	0.833	0.182	0.0109	0.0196	0.0339	0.348	0.255
	NVF (Ours)	**25.36**	**0.931**	**0.061**	**0.0029**	**0.0107**	**0.0134**	**0.564**	**0.396**
Mic	Random	21.18	0.851	0.205	0.0081	0.0294	0.0276	0.468	0.257
	WD	26.79	0.942	0.067	0.0022	0.0176	0.0087	0.755	0.564
	ActiveRMAP	26.60	0.940	0.069	0.0022	0.0187	0.0095	0.752	0.532
	AIR	24.81	0.927	0.107	0.0034	0.0165	0.0091	0.728	0.508
	ActiveNeRF	24.96	0.926	0.101	0.0033	0.0198	0.0105	0.709	0.497
	NeurAR	25.15	0.889	0.159	0.0031	0.0304	0.0099	0.679	0.528
	NVF (Ours)	**27.99**	**0.956**	**0.053**	**0.0016**	**0.0161**	**0.0070**	**0.854**	**0.566**
Ship	Random	15.75	0.578	0.483	0.0281	0.0580	0.0456	0.250	0.252
	WD	19.54	0.663	0.369	0.0112	0.0525	0.0313	0.374	0.540
	ActiveRMAP	19.61	0.665	0.343	0.0112	0.0487	0.0290	0.385	0.543
	AIR	19.22	0.658	0.367	0.0121	0.0515	0.0307	0.378	0.513
	ActiveNeRF	17.19	0.569	0.485	0.0197	0.0606	0.0370	0.329	0.496
	NeurAR	19.38	0.556	0.491	0.0115	0.0569	0.0461	0.331	0.483
	NVF (Ours)	**22.32**	**0.742**	**0.254**	**0.0059**	**0.0445**	**0.0188**	**0.454**	**0.596**