IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Bayesian NeRF: Quantifying Uncertainty With Volume Density for Neural Implicit Fields

Sibaek Lee, Kyeongsu Kang, *Graduate Student Member, IEEE*, Seongbo Ha, and Hyeonwoo Yu, *Member, IEEE*

***Abstract*—We present a Bayesian Neural Radiance Field (NeRF), which explicitly quantifies uncertainty in the volume density by modeling uncertainty in the occupancy, without the need for additional networks, making it particularly suited for challenging observations and uncontrolled image environments. NeRF diverges from traditional geometric methods by providing an enriched scene representation, rendering color and density in 3D space from various viewpoints. However, NeRF encounters limitations in addressing uncertainties solely through geometric structure information, leading to inaccuracies when interpreting scenes with insufficient real-world observations. While previous efforts have relied on auxiliary networks, we propose a series of formulation extensions to NeRF that manage uncertainties in density, both color and density, and occupancy, all without the need for additional networks. In experiments, we show that our method significantly enhances performance on RGB and depth images in the comprehensive dataset. Given that uncertainty modeling aligns well with the inherently uncertain environments of Simultaneous Localization and Mapping (SLAM), we applied our approach to SLAM systems and observed notable improvements in mapping and tracking performance. These results confirm the effectiveness of our Bayesian NeRF approach in quantifying uncertainty based on geometric structure, making it a robust solution for challenging real-world scenarios.**

***Index Terms*—Deep learning for visual perception, mapping, SLAM.**

I. Introduction

THE advent of Neural Radiance Fields (NeRF) [18] has significantly advanced the field of novel view synthesis, allowing for the continuous synthesis of views from given images at unseen positions and orientations. As opposed to traditional geometric approaches [23], [25], NeRF leverages a learned neural network model to predict color and density in 3D space from coordinates and view directions, thus better handling complex scenes, intricate textures, and lighting variations. Subsequent research has propelled its real-world application in robotics [22], [37], virtual reality [4], digital twins [14], and autonomous driving [7], [39], underscoring the importance of real-time performance and efficiency when data is limited. However, advancements are hindered by challenges like accurately predicting scenes in unobserved views with limited data availability [41]. This is inherently challenged by the limitations of the sensor's field of view (FoV) and physical occlusions, such as buildings or other vehicles, frequently result in gaps in observable data. Moreover, the intrinsic nature of the real-world ensures that sensor data is invariably accompanied by errors [34]. To overcome these challenges, it is necessary to incorporate uncertainty consideration and this approach is crucial for precise interpretation of and adaptation to the variabilities encountered in real-world environments [1], [3], [9].

Given the importance of considering uncertainty in neural radiance fields, various studies have focused on addressing the variability of color in space. [17] tackled photometric variations to deal with images captured in various environments, while [20] used data selection strategies for enhanced learning. They incorporate an L1 regularization term to mitigate transient density issues. However, considering uncertainty in color presents several critical drawbacks. Firstly, it fails to address uncertainties in other sensor data such as density, which are essential for a comprehensive understanding of 3D spaces. Moreover, predicting pixel variance does not fully capture the uncertainty in the scene's radiance field and can lead to suboptimal results. In response to these challenges, [26], [27] suggested employing an additional network to consider the density uncertainty. However, the use of an extra network is inefficient for real-time applications and does not guarantee accurate estimations due to the approximation of the probability distribution.

We introduce a series of methodical approaches that allow us to incorporate density-related uncertainty into the NeRF model without altering its network structure. This methodology not only explicitly addresses density uncertainty but also renders the model suitable for real-time applications by leveraging Bayesian techniques to manage and interpret the uncertainties inherent in the data. Our approaches improves rendering in unobserved views, as shown in Fig. 1, where training data from limited viewpoints lead to suboptimal performance. Incorporating a Bayesian framework enhances the model's capacity to handle unseen data uncertainties, boosting rendering quality from various angles. Furthermore, we address the limitations of sensors that do not capture color data, such as Lidar and Thermal Imaging Cameras, which make traditional uncertainty methods ineffective. By

Received 5 September 2024; accepted 30 December 2024. Date of publication 6 January 2025; date of current version 22 January 2025. This article was recommended for publication by Associate Editor H. Blum and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported by the National Research Foundation of Korea under Grant RS-2024-00359937. *(Corresponding author: Hyeonwoo Yu.)*

The authors are with the Department of Intelligent Robotics, Sungkyunkwan University, Suwon 12345, South Korea (e-mail: lmjlss@skku.edu; thithin0821@skku.edu; sobo3607@skku.edu; hwyu@skku.edu).

The code is available at: https://github.com/Lab-of-AI-and-Robotics/BayesianNeRF.

Digital Object Identifier 10.1109/LRA.2025.3526572

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Fig. 1. Quantitative results. Ground Truth (Left), baseline model prediction (Middle), our method's prediction (Right). All models were trained on 4 images from the NeRF dataset and predict unobserved viewpoints.

focusing on density as a universal attribute detectable across different sensor types, our method extends the applicability of NeRF models beyond the realm of RGB cameras. Integrating uncertainty across different sensors broadens NeRF's flexibility and significantly improves performance in real-world scenarios, demonstrating the robustness and adaptability of our enhanced framework.

II. Related Work

A. Novel-View Synthesis and NeRF

The field of novel-view synthesis has seen remarkable advancement as computer vision technologies have evolved. Initially, traditional methods were concentrated on understanding the structure of 3D scenes through camera poses, utilizing techniques such as structure-from-motion (SFM) [23] and Multi-View Stereo (MVS) [25]. These methods, based on geometric approaches, aimed to synthesize images from novel viewpoints by accurately modeling 3D structures. The advent of NeRF marked a significant leap forward, enabling the creation of realistic renderings from a smaller set of images through 3D representations learned by neural networks. NeRF estimates volume density and color information based on 3D coordinates and viewing directions, with its network output facilitating image generation via a rendering function. These images reflect observations within the modeled space, informed by the direction of the camera view [17], [20], [26]. Efforts have since been directed at addressing the original NeRF's limitations, spurring a variety of enhancements. For instance, changes to encoding methods have sped up the learning process [2], [6], [19], while large-scale scenes are now depicted using multiple implicit neural networks [42], [44]. Additionally, the introduction of new priors has facilitated learning from sparse data [32], [35]. NeRF's application scope has also expanded, encompassing areas such as scene editing [36], [45], converting text to 3D models [13], [21], and enhancing visual scene-based SLAM technologies [10], [22], [30], [43], [49], demonstrating its versatility and potential in various domains.

B. Uncertainty Estimation in NeRF

Estimating uncertainty is a well-known and important issue in deep learning. Similarly, in NeRF, considering uncertainty is crucial for reducing errors stemming from sensor noise, sensor field of view (FOV), occlusions, and other factors, enabling the representation of space with reliability and robustness. Due to this importance, research efforts are underway to integrate uncertainty estimation into existing NeRF-based techniques. For instance, [17] accurately render scenes from images taken in various environments, [20] select data for subsequent learning, and [12] consider uncertainty in color to measure the reliability of generated images. While these approaches explicitly estimate uncertainty and do not require more extensive training than baseline models, they have the drawback of not considering uncertainty for volume density. Other recent works [5] represent rendering cleanup by measuring uncertainty in trained models. [31] proposes a method that utilizes an ensemble of multiple NeRF models to quantify uncertainty, which can be applied to next view selection. However, these approaches can only capture epistemic uncertainty and cannot capture inherent aleatoric uncertainty in the data. Addressing volume density uncertainty, [26], [27] presents a method that employs an additional network to measure depth uncertainty. This method is difficult to apply in real-time situations due to additional network usage. To address this challenge, we introduce a series of methods based on Bayesian approaches that explicitly estimate volume density uncertainty, tackling the complex issue of depth uncertainty and effectively predicting unobserved views. Our methodologies formulate uncertainty for volume density and offer ways to quantify uncertainty without the need for additional estimation networks.

III. Methodology

A. Background

NeRF is a coordinate-based neural representation of a 3D volumetric scene [18]. The goal is to find a continuous non-linear regression $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \rho)$, where Θ represents the network parameters, $\mathbf{x} = (x, y, z)$ are the spatial 3D coordinates, and $\mathbf{d} = (\theta, \phi)$ are the viewing directions of the camera. The output of F_Θ estimates the color $\mathbf{c}$ and density ρ at a given point in the neural field. To render an image, the color of each pixel is determined by integrating colors along a ray in the neural field. The expected image color $C(\mathbf{r})$ for a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ starting from the camera origin $\mathbf{o}$ with near and far bounds t_n and t_f is given by:

$$C(r) = \int_{t_n}^{t_f} T(t)\rho(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \tag{1}$$

where $T(t)$ is the accumulated transmittance along the ray. For practical implementation, the integral is approximated by sampling N discrete points along the ray at positions t_i, spaced by $\delta_i = t_{i+1} - t_i$. This discretization maintains a continuous scene representation, and the rendering equation becomes:

$$\begin{aligned}\hat{C}(r) &= \sum_i^N \underbrace{T_i(1 - \exp(-\delta_i\rho_i))}_{\alpha_i} c_i \\ &= \sum_i^N \alpha_i c_i, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1} \delta_j\rho_j\right).\end{aligned} \tag{2}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

B. Uncertainty Estimation

Training with limited observations presents a challenge in accurately predicting unobserved parts of the 3D scene. To effectively tackle the problem, it is essential to adopt the Bayesian learning [9] for considering the observation and estimation uncertainty. For example, [17], [20] simply assume that only the color $\mathbf{c}$ follows a Gaussian distribution $\mathbf{c} \sim \mathcal{N}(\mu_{\mathbf{c}}, \sigma_{\mathbf{c}}^2)$, while treating density ρ and transmittance T as constants. As in (2), since the cumulated color $C(\mathbf{r})$ along the ray $\mathbf{r}$ is a linear combination of Gaussian random variable $\mathbf{c}$, it also follows the Gaussian distribution

$$C(\mathbf{r}) \sim \mathcal{N}\left(\sum_{i=1}^{N_s} \mu_i \alpha_i, \sum_{i=1}^{N_s} \sigma_i^2 \alpha_i^2\right). \tag{3}$$

This assumption makes it easy to model the variability in color predictions, facilitating the estimation of color uncertainty and its impact on the rendered scene. However, this color-only approach is limited by considering only RGB images, making it impossible to extend to other sensor values such as depth or intensity. To address the general uncertainty in estimating 3D space from sensors, one can consider incorporating a density field in neural representation. Unfortunately, unlike color-only case, when we assume that density ρ follows a Gaussian distribution $\rho \sim \mathcal{N}(\mu_d, \sigma_d^2)$ the problem becomes highly complicated. Since $T_i = \exp(-\sum_{j=1}^{i-1} \delta_j \rho_j)$ as in (2), we have $T_i \sim Lognormal(\mu_{T_i}, \sigma_{T_i}^2)$ where $\mu_{T_i} = -\sum_{j=1}^{i-1} \delta_j \mu_j$ and $\sigma_{T_i}^2 = \sum_{j=1}^{i-1} \delta_j^2 \sigma_j^2$. Meanwhile, α in (2) can be rewritten as:

$$\begin{aligned}
\alpha_i &= \exp\left(-\sum_{j=1}^{i-1} \delta_j \rho_j\right)\left(1 - \exp\left(-\delta_i \rho_i\right)\right) \\
&= \exp\left(-\sum_{j=1}^{i-1} \delta_j \rho_j\right) - \exp\left(-\sum_{j=1}^{i} \delta_j \rho_j\right) \\
&= T_i - T_{i+1},
\end{aligned} \tag{4}$$

which is the difference of two lognormal random variables T_i and T_{i+1}. It is known to be challenging to have the closed-form distribution of the difference of two lognormal random variables, even the two random variables are independent and thus uncorrelated [15], [40]. Therefore, even if we assume that the color $\mathbf{c}$ in (2) is a constant, $C(\mathbf{r})$ still follows an intractable distribution since $C(\mathbf{r})$ is a linear combination of α. To model this complex distribution, recent works such as [26], [27] utilize generative models and additional network structures, resulting in complex frameworks and challenging learning tasks. In the following, we introduce a series of our analytic but simple and effective approaches with consideration of density uncertainty without the need of additional network structures or trainings.

1) Gaussian Approximation of Density Uncertainty: We propose a method to optimize the NeRF model by considering density uncertainty through a series of assumptions. Using α_i represented in (4), expected color $C(\mathbf{r})$ and expected depth $D(\mathbf{r})$ of the ray $\mathbf{r}$ can be derived as [8], [16], [33], [47]:

$$C(\mathbf{r}) = \sum_{i=1}^{N_s} \mathbf{c}_i \alpha_i, \tag{5}$$

$$D(\mathbf{r}) = \sum_{i=1}^{N_s} d_i \alpha_i = \sum_{i=1}^{N_s} \frac{t_{i+1} + t_i}{2} \alpha_i. \tag{6}$$

Here, let us make a slightly strong assumption regarding α, such that $\sum \delta_j \rho_j \ll 1$. In other words, we assume that the intervals between samples are very narrow. Furthermore, neural fields are generally sparse, and the volume density is forced to take values less than 1 using activation functions such as sigmoid. Then we can approximately represent (4) as the following:

$$\alpha_i \approx \left(1 - \sum_{j=1}^{i-1} \delta_j \rho_j\right) - \left(1 - \sum_{j=1}^{i} \delta_j \rho_j\right) = \delta_i \rho_i. \tag{7}$$

Now let k'th volume density ρ_k follows the Gaussian distribution $\rho_k \sim \mathcal{N}(\mu_k, \sigma_k^2)$ and the densities are i.i.d.. Assume that the color $\mathbf{c}$ is a constant. Substituting (7) into (5), we approximately have the rendered color as $C(\mathbf{r}) \approx \sum_{i=1}^{N_s} \mathbf{c}_i \delta_i \rho_i$. As the rendered color $C(\mathbf{r})$ is now a linear combination of Gaussian random variables, its distribution can be written as:

$$C(\mathbf{r}) \sim \mathcal{N}\left(\sum_{i=1}^{N_s} \mathbf{c}_i \delta_i \mu_i, \sum_{i=1}^{N_s} \mathbf{c}_i^2 \delta_i^2 \sigma_i^2\right), \tag{8}$$

and we can obtain the approximated solution $\{(\hat{\mu}_i, \hat{\sigma}_i)\}$ of the volume density by minimizing the following negative log-likelihood:

$$\ln \sum_{i=1}^{N_s} \mathbf{c}_i^2 \delta_i^2 \sigma_i^2 + \frac{\left(C(\mathbf{r}) - \sum_{i=1}^{N_s} \mathbf{c}_i \delta_i \mu_i\right)^2}{\sum_{i=1}^{N_s} \mathbf{c}_i^2 \delta_i^2 \sigma_i^2}. \tag{9}$$

Similarly, we can obtain the approximated solution $\{(\hat{\mu}_i, \hat{\sigma}_i)\}$ of $D(\mathbf{r})$ by minimizing the following negative log-likelihood:

$$\ln \sum_{i=1}^{N_s} d_i^2 \delta_i^2 \sigma_i^2 + \frac{\left(D(\mathbf{r}) - \sum_{i=1}^{N_s} d_i \delta_i \mu_i\right)^2}{\sum_{i=1}^{N_s} d_i^2 \delta_i^2 \sigma_i^2}. \tag{10}$$

By using (9) or (10), now we can consider the uncertainty of both color or depth, or other sensor information, by utilizing the density uncertainty.

2) Gaussian Approximation of Density and Color Uncertainty: In addition to the above method, we also propose a method that considers both color and density by assuming that color follows a Gaussian distribution, rather than being constant. Specifically, we assume that $\mathbf{c}_k \sim \mathcal{N}(\mu_{\mathbf{c}_k}, \sigma_{\mathbf{c}_k}^2)$, then (5) becomes the summation of the product of two Gaussian random variables, ρ and $\mathbf{c}$. While the distribution of this product is infeasible to compute explicitly, under certain conditions, specifically when the means of the Gaussian distributions are significantly larger than their standard deviations (i.e., $\mu_k \gg \sigma_k$ and $\mu_{\mathbf{c}_k} \gg \sigma_{\mathbf{c}_k}$), the product can be approximated by a Gaussian distribution [24]. Practically, this means that the relative variability of the random variables is low, and their distributions are sharply peaked around the mean. This allows us to approximate

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

the product $\rho_k \mathbf{c}_k$ as another Gaussian distribution with the following mean and variance:

$$\sigma_k \mathbf{c}_k \sim \mathcal{N}\left(\mu_k \mu_{\mathbf{c}_k}, \sigma_k^2 \mu_{\mathbf{c}_k}^2 + \sigma_{\mathbf{c}_k}^2 \mu_k^2 + \sigma_k^2 \sigma_{\mathbf{c}_k}^2\right).$$

Consequently, the distribution of rendered color $C(\mathbf{r})$ can be approximated as:

$$C(\mathbf{r}) \sim \mathcal{N}\left(\sum_{i=1}^{N_s} \delta_i \mu_i \mu_{\mathbf{c}_i}, \sum_{i=1}^{N_s} \delta_i^2 \left(\sigma_i^2 \mu_{\mathbf{c}_i}^2 + \sigma_{\mathbf{c}_i}^2 \mu_i^2 + \sigma_i^2 \sigma_{\mathbf{c}_i}^2\right)\right). \tag{11}$$

Using this approximated distribution, and similar to (9) and (10), we can obtain the approximated solutions $(\hat{\mu}_i, \hat{\sigma}_i), (\hat{\mu}_{\mathbf{c}_i}, \hat{\sigma}_{\mathbf{c}_i})$ for the volume density and color by minimizing the following negative log-likelihood:

$$\ln\left(\sum_{i=1}^{N_s} \delta_i^2 \left(\sigma_i^2 \mu_{\mathbf{c}_i}^2 + \sigma_{\mathbf{c}_i}^2 \mu_i^2 + \sigma_i^2 \sigma_{\mathbf{c}_i}^2\right)\right) + \frac{\left(C(\mathbf{r}) - \sum_{i=1}^{N_s} \delta_i \mu_i \mu_{\mathbf{c}_i}\right)^2}{\sum_{i=1}^{N_s} \delta_i^2 \left(\sigma_i^2 \mu_{\mathbf{c}_i}^2 + \sigma_{\mathbf{c}_i}^2 \mu_i^2 + \sigma_i^2 \sigma_{\mathbf{c}_i}^2\right)}. \tag{12}$$

3) Markov Approach for Density Uncertainty: The approximation in (7) hardly holds when the ray $\mathbf{r}$ travels a long distance, or the volume density σ takes a large value. As a result, the approximated solutions obtained from (9) and (12) become inaccurate. To fundamentally address such approximation errors, we apply the Markov assumption to the traversal of rays as follows: 1) In (2), α_i represents $i'th$ random variable involving $i'th$ density ρ_i and the past densities $\rho_0, \ldots, \rho_{i-1}$ expressed as T_i. Since $\rho_0, \ldots, \rho_{i-1}$ are variables observed along the path traveled by the ray in temporal order, they can be considered deterministic variables at time step i. Therefore, the probability distribution for α_i can be expressed as $p(\alpha_i|\rho_i, \ldots, \rho_0) \simeq p(\alpha_i|\rho_i) = p(\alpha_i|o)$. Here, $o = 1 - \exp(-\delta_i \rho_i)$ is defined as the occupancy variable, which effectively normalizes density to indicate the presence of material. 2) Similarly, if the summation in the rendering process (5) and (6) is performed in temporal order, then α_i can also be regarded as a variable determined in certain time step. In other words, at time step $i+1$, $\alpha_0, \ldots, \alpha_i$ are already obtained and determined, making them independent of α_{i+1}. 3) We assume that occupancy o follows a Gaussian distribution $o \sim \mathcal{N}(\mu_o, \sigma_o^2)$. Following these reasonings 1)-3), the rendered color $C(\mathbf{r})$ in (5) and (6) can be treated as the linear combination of the Gaussian random variables, simplifying its distribution as follows:

$$C(\mathbf{r}) \sim \mathcal{N}\left(\sum_{i=1}^{N_s} \mathbf{c}_i T_i \mu_{o_i}, \sum_{i=1}^{N_s} \mathbf{c}_i^2 T_i^2 \sigma_{o_i}^2\right). \tag{13}$$

Using the distribution of the rendered color, we can derive the approximate solution $\{(\hat{\mu}_i, \hat{\sigma}_i)\}$ for volume density by minimizing the following negative log-likelihood:

$$\ln \sum_{i=1}^{N_s} \mathbf{c}_i^2 T_i^2 \sigma_{o_i}^2 + \frac{\left(C(\mathbf{r}) - \sum_{i=1}^{N_s} \mathbf{c}_i T_i \mu_{o_i}\right)^2}{\sum_{i=1}^{N_s} \mathbf{c}_i^2 T_i^2 \sigma_{o_i}^2}. \tag{14}$$

Note that for $D(\mathbf{r})$, by replacing $\mathbf{c}_i$ to d_i, we can have the similar formulations for the depth estimation since we have the generalized approximation for the occupancy o. We assume the probability variables to be independent with respect to the time sequence and validate the efficiency of such approximation methods through experiments.

IV. Experiment

A. Experimental Setup

1) Evaluation: In our experiments, we explore the concept of unobserved and observed views in the context of NeRF. Unobserved views are perspectives not captured by the camera due to limited trajectories or self-occlusion. In contrast, observed views are near those captured by the camera. Our method estimates unobserved views without an additional network, focusing on uncertainty related to density. Additionally, we compare the predictions of observed views to analyze the performance improvements in rendering unobserved views. We evaluated five different methods: (a) *Baseline*, using the vanila NeRF method with only photometric loss; (b) *Color* [17], [20], integrating color uncertainty into NeRF; (c) *Density*, emphasizing density uncertainty during NeRF training; (d) *Den_cf* [26], which uses a generative model to estimate density uncertainty via an additional network; (e) *Color + Density*, addressing both color and density uncertainties; and (f) *Occupancy*, advancing NeRF by considering occupancy uncertainty based on Markov assumptions. In addition to these experiments, we extend our approach to scenarios such as SLAM, where uncertainty is prevalent. Specifically, we apply our method to NICE SLAM [49], which represents maps using NeRF, allowing us to validate the effectiveness of our uncertainty modeling strategy.

2) Metric: In our study, we use PSNR [11], SSIM [46], and LPIPS [48] as primary metrics for RGB image quality. For depth image evaluation, we apply Absolute Relative Error (AbsRel) and Root Mean Square Error in log space (RMSElog) to measure depth deviation with a focus on precision. We also use Average log10 Error (log10) and Threshold Accuracy (δ_i) to assess depth accuracy. In the SLAM experiments, we evaluate both mapping and tracking performance. Mapping is assessed using the aforementioned metrics, along with additional 2D and 3D metrics for scene geometry. For 2D evaluation, we calculate L1 loss on depth maps by comparing reconstructed meshes to the ground truth. For 3D evaluation, we follow [30], focusing on Accuracy [cm], Completion [cm], and Completion Ratio [$<$ 5 cm %], excluding regions outside any camera's viewing frustum. Mapping evaluation is conducted on images not used in keyframes, ensuring unbiased assessment. Camera tracking performance is measured using ATE RMSE.

B. Experiment Results and Discussion

1) Results on NeRF Synthetic and LLFF Dataset: In Table I, we show the RGB image metric results for the six methods. The synthetic dataset shows significant improvements, while the real-world dataset shows modest gains for forward-facing images. Additionally, settings with fewer training images yield

TABLE I
QUANTITATIVE RESULTS ON NERF SYNTHETIC AND LLFF DATASETS [18]

PSNR ↑

Dataset	Base	Col	Den	Den_cf	Col+Den	Occu
Syn2	11.75	11.29	13.61	12.8	14.72	**15.12**
Syn4	15.40	16.18	17.02	15.9	18.01	**18.30**
Syn8	18.04	18.91	19.19	18.5	**20.87**	20.75
SynF	25.57	24.20	24.25	21.2	25.64	**25.68**
Real4	12.88	12.97	12.88	14.0	**13.51**	13.45
Real8	22.60	22.66	22.26	20.3	23.01	**23.10**
RealF	24.81	24.03	23.86	21.4	24.78	**24.97**

SSIM ↑

Dataset	Base	Col	Den	Den_cf	Col+Den	Occu
Syn2	0.61	0.61	0.66	**0.71**	0.67	0.69
Syn4	0.74	0.75	0.76	**0.79**	0.78	**0.79**
Syn8	0.79	0.80	0.79	0.80	**0.82**	**0.82**
SynF	**0.88**	0.86	0.86	0.82	**0.88**	**0.88**
Real4	0.52	0.53	0.52	**0.58**	0.54	0.55
Real8	0.69	0.69	0.67	0.66	0.69	**0.71**
RealF	0.76	0.75	0.73	0.70	0.76	**0.77**

LPIPS ↓

Dataset	Base	Col	Den	Den_cf	Col+Den	Occu
Syn2	0.44	0.42	0.36	0.39	0.34	**0.33**
Syn4	0.27	0.26	0.24	0.29	0.21	**0.20**
Syn8	0.22	0.21	0.21	0.24	**0.17**	**0.17**
SynF	0.11	0.14	0.14	0.22	0.11	**0.10**
Real4	0.62	0.63	0.62	0.63	0.60	**0.59**
Real8	0.22	0.24	0.25	0.47	0.20	**0.19**
RealF	0.17	0.21	0.22	0.42	0.17	**0.16**

We compare our methods with the baseline [18], color uncertainty methods Col [17], [20], and Den cf, which estimates density uncertainty using a generative model [26]. Experiments are conducted across all scenes, and average values are reported. Significant improvements are observed in both unobserved and observed views, with methods addressing occupancy and combined colordensity uncertainties showing superior performance. "Full" indicates using the entire training dataset.

greater performance improvements than those using the full number of images in the dataset. This outcome indicates that in scenarios where data is limited, the inherent uncertainty within the data becomes more pronounced, making the integration of uncertainty into the learning process significantly more impactful.

While considering uncertainty generally improves performance, the (b) *Color* and (c) *Density* approaches show relatively low performance, especially with fewer training images. The (b) *Color* method struggles to estimate density accurately under these conditions, leading to visual artifacts, such as objects appearing faded or vanishing. This limitation highlights the (b) *Color* method's inability to capture detailed density variations with limited data. In contrast, methods that account for uncertainty in density demonstrate improved robustness. These methods preserve objects' visibility and structural integrity, emphasizing the critical role of managing uncertainty in geometric structures.

However, even methods that consider density have their own issues. In our experiments, the (c) *Density* method revealed that high-density datasets lead to challenges for accurate estimation. This results in a decline in performance and a less reliable MLE formulation, with the (c) *Density* method underperforming compared to the (e) *Color + Density* and (f) *Occupancy* approaches. Additionally, the (d) *Den_cf* method, which employs a generative model to estimate density uncertainty via an additional network, shows inferior performance in PSNR and LPIPS metrics, especially on synthetic datasets with fewer training images. This occurs because neural networks simplify complex probability distributions, hindering the capture of fine textures and details. However, it performs well in the SSIM metric by fitting the complex distribution as closely as possible, effectively capturing the overall structure of the object despite not precisely modeling the probability distribution. These issues are effectively addressed by the (e) *Color + Density* approach, which integrates both methods and shows significant performance improvements. However, it occasionally fails to meet the fundamental assumptions of the (c) *Density* approach, leading to suboptimal performance when volume density assumptions are not fully satisfied. In contrast, the (f) *Occupancy* approach, based on reasonable Markov assumptions, exhibits remarkable stability across both synthetic and real-world datasets, managing inherent uncertainty without relying heavily on specific assumptions.

2) Results on ModelNet Dataset: In our ModelNet dataset analysis, as shown in Table II, we present the metric results for depth images, where the absence of RGB information requires focusing on geometric reconstruction. We compare the (a) *Baseline*, (c) *Density*, (d) *Den_cf*, and (f) *Occupancy* methods. Our findings align with the trends seen in RGB images, showing significant improvements in unobserved views, while observed views exhibit more modest gains. However, unlike the results in RGB images, in the unobserved view setting with limited training images, the (d) *Den_cf* method shows reasonably good performance. This is because depth image prediction does not require the estimation of textures or fine details; instead, the primary goal is to predict the object's shape from a limited number of training images. In such cases, approximating the complex distribution as accurately as possible using a generative model proves to be effective. For instance, in the sofa and dresser scene from Fig. 4, blurring in empty spaces is significantly reduced by incorporating uncertainty, particularly with the (d) *Den_cf* and (f) *Occupancy* methods, which refine the clarity of the depth images more effectively than the (c) *Density* method. In settings where all training images are used, the importance shifts to predicting detailed aspects beyond just the object's shape. Here, our (f) *Occupancy* method, which explicitly estimates occupancy uncertainty, demonstrates superior performance.

3) Results in SLAM Environment: In our previous experiments, we demonstrated the effectiveness of training neural fields by incorporating uncertainty, particularly by adjusting the level of uncertainty inherent in the data. Building on these results, we further validate our approach in NICE SLAM [49], a scenario with varying levels of uncertainty. However, existing methods that estimate uncertainty using generative models, such as CF-NeRF [26], require significantly more computational resources. Under the vanilla NeRF settings, we observed that CF-NeRF's training time is approximately 21 times longer, and the runtime per image is about 11 times longer than the baseline NeRF. These substantial computational demands make such methods unsuitable for real-time applications like SLAM, where both training and rendering need to be performed efficiently. In contrast, our approach explicitly estimates uncertainty without relying on additional networks, resulting in training and runtime that are only approximately 1.01 times and 1.04 times longer than the baseline. Therefore, we choose to focus on occupancy

TABLE II
QUANTITATIVE RESULTS ON MODELNET [38]

Unobserved View Setting

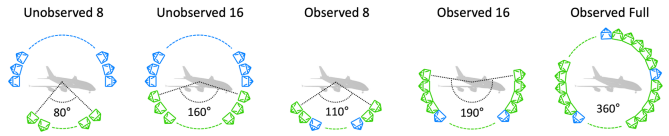
Dataset	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$	AbsRel↓	RMSE(log)↓	log10↓
Baseline 8 [18]	0.497	0.672	0.755	0.537	0.467	0.159
Uncert den 8	0.506	0.685	0.809	0.492	0.440	0.151
Uncert den_cf 8 [26]	**0.568**	0.709	**0.841**	**0.448**	**0.403**	**0.137**
Uncert occu 8	0.543	**0.712**	0.837	0.451	0.418	0.140
Baseline 16 [18]	0.740	0.906	0.962	0.203	0.262	0.076
Uncert den 16	0.762	0.911	0.961	0.192	0.257	0.073
Uncert den_cf 16 [26]	**0.788**	0.914	**0.968**	0.182	0.249	0.070
Uncert occu 16	0.785	**0.922**	0.964	**0.177**	**0.245**	**0.068**

Observed View Setting

Dataset	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$	AbsRel↓	RMSE(log)↓	log10↓
Baseline 8 [18]	0.880	0.939	0.971	0.102	0.192	0.044
Uncert den 8	0.886	0.943	0.973	0.099	0.188	0.043
Uncert den_cf 8 [26]	**0.894**	0.945	**0.978**	**0.095**	**0.183**	**0.039**
Uncert occu 8	0.891	**0.946**	0.975	0.096	0.184	0.041
Baseline 16 [18]	0.888	0.943	0.974	0.099	0.189	0.043
Uncert den 16	0.891	0.946	0.977	0.097	0.186	0.042
Uncert den_cf 16 [26]	0.892	**0.952**	0.977	**0.094**	**0.182**	**0.041**
Uncert occu 16	**0.895**	0.949	**0.978**	**0.094**	0.183	**0.041**
Baseline Full [18]	0.896	0.946	0.976	0.096	0.180	0.041
Uncert den Full	0.896	0.946	0.977	0.096	0.181	0.041
Uncert den_cf Full [26]	0.898	0.945	0.976	0.095	0.178	0.040
Uncert occu Full	**0.902**	**0.949**	**0.978**	**0.092**	**0.176**	**0.039**

Given that depth images lack color information, we compare the (a) Baseline, (c) Density, and (e) Occupancy methods. Experiments were conducted on two instances across all classes, with average values provided. The dataset setting follows fig. 2, and similar to the RGB dataset, considering uncertainty in the unobserved view prediction setting demonstrates significant performance improvements.

(a) NeRF synthetic and real-world dataset experiment setting



(b) ModelNet dataset experiment setting

Fig. 2. Dataset Settings. We conduct experiments on NeRF synthetic, real-world, and ModelNet datasets, using green cameras for training and blue cameras for testing. (a) The NeRF synthetic dataset evaluates unobserved predictions with limited training views, while the real-world dataset assesses observed predictions with forward-facing camera trajectories. (b) The ModelNet dataset evaluates both unobserved and observed predictions using 36 images spaced at 10-degree intervals.

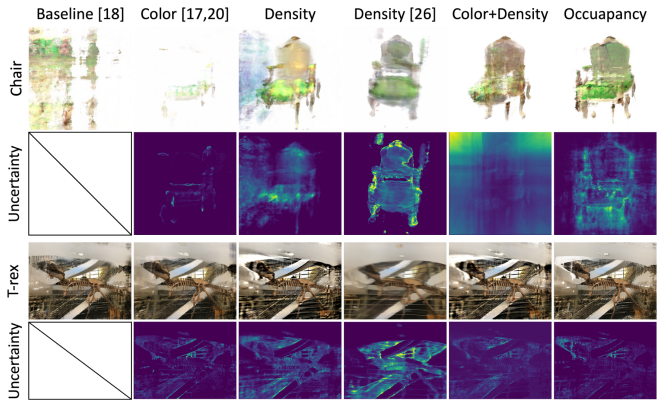


Fig. 3. Qualitative Comparisons on NeRF synthetic and LLFF datasets [18]. The Chair scene is from the NeRF synthetic dataset trained with the syn4 setting, and the T-rex scene is from the LLFF dataset trained with the real8 setting. These settings are described in Fig. 2. While incorporating uncertainty improves performance, the (b) *Color* method in synthetic datasets struggles with density estimation, causing objects to appear faded.

uncertainty, as it is suitable for real-time applications and has demonstrated good performance in previous experiments.

As demonstrated in Table III, when using all available images for training, the performance difference between methods that consider uncertainty and those that do negligible, since comprehensive data coverage minimizes spatial uncertainty. However, when only every second or third image is used, thereby increasing spatial uncertainty, our method that incorporates uncertainty significantly outperforms the baseline. To further test the robustness of our approach, we conducted additional experiments where we adjusted key hyperparameters, such as the mapping frequency and keyframe selection interval, to compensate for the reduced data. Even with these adjustments, our uncertainty method continued to outperform the baseline, demonstrating its effectiveness in challenging environments with limited data. This improved performance is visually evident in Fig. 5, which illustrates intermediate rendering results during the mapping process. The original method without uncertainty produces less accurate scene representations under limited data conditions. In contrast, our method provides more robust and accurate renderings, highlighting the benefits of incorporating uncertainty. Additionally, our estimated trajectory aligns more closely with the ground truth, enhancing tracking performance. These results emphasize that considering uncertainty improves both mapping and tracking, especially in limited data scenarios.

TABLE III
QUANTITATIVE RESULTS ON REPLICA [28] AND TUM [29]

Methods	Metrics	RAvg$_1$	OAvg$_1$	TAvg$_1$	RAvg$_2$	OAvg$_2$	TAvg$_2$	RAvg$_3$	OAvg$_3$	TAvg$_3$
Baseline Method [49]	PSNR ↑	22.48	24.49	14.66	17.35	20.31	12.64	13.82	15.86	13.21
	SSIM ↑	0.75	0.82	0.47	0.57	0.68	0.35	0.43	**0.56**	**0.38**
	LPIPS ↓	0.43	0.34	**0.54**	0.56	0.54	0.61	0.66	0.56	0.60
	RMSE ↓	**0.05**	0.05	**0.04**	0.49	0.68	0.21	1.08	0.87	0.20
	Acc ↓	**3.12**	4.20	–	32.1	17.7	–	59.6	40.8	–
	Comp ↓	**3.16**	4.01	–	13.6	9.44	–	36.4	19.6	–
	Comp Ratio ↑	**87.9**	81.2	–	38.5	56.3	–	18.8	39.0	–
	Depth L1 ↓	**8.12**	24.3	–	104.5	75.6	–	136.3	150.6	–
Uncertainty Method	PSNR ↑	**22.74**	**24.94**	**15.05**	**18.82**	**20.45**	**13.67**	**15.25**	**16.63**	**13.33**
	SSIM ↑	**0.77**	**0.84**	**0.49**	**0.62**	**0.68**	**0.43**	**0.49**	0.55	0.37
	LPIPS ↓	**0.42**	**0.33**	0.55	**0.53**	**0.44**	**0.47**	**0.62**	**0.55**	**0.59**
	RMSE ↓	0.06	**0.04**	0.06	**0.43**	**0.28**	**0.10**	**0.78**	**0.73**	**0.17**
	Acc ↓	3.71	**3.28**	–	**20.7**	**13.3**	–	**42.5**	**33.3**	–
	Comp ↓	4.29	**3.87**	–	**12.7**	**8.87**	–	**35.9**	**17.6**	–
	Comp Ratio ↑	79.7	**83.1**	–	**49.7**	**57.4**	–	**19.5**	**40.8**	–
	Depth L1 ↓	12.2	**13.6**	–	**57.8**	**74.3**	–	**132.3**	**137.2**	–

We compare the mapping and tracking performance of the uncertainty method with the original method. The results are averaged across the room and office scenes in the Replica dataset and three scenes in the TUM dataset. Scenarios are denoted as follows: _1 for all data used, _2 for even-indexed images only, and _3 for images indexed by multiples of 3.

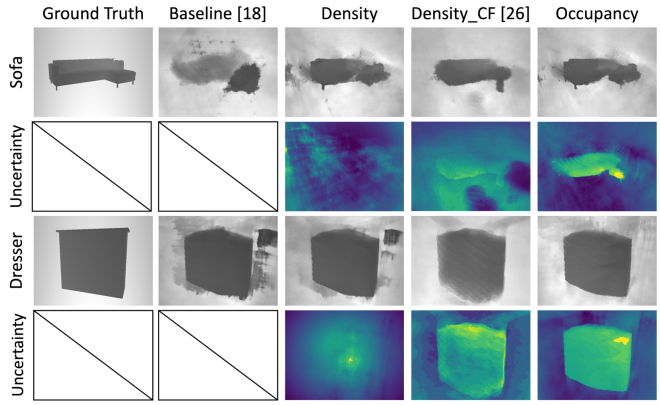


Fig. 4. Qualitative Comparisons on ModelNet [38]. The sofa and dresser scene employs an unobserved view setting and is trained using 8 images. In this scene, the (d) *Den_cf* and (f) *Occupancy* methods effectively addresses the blurring issue.

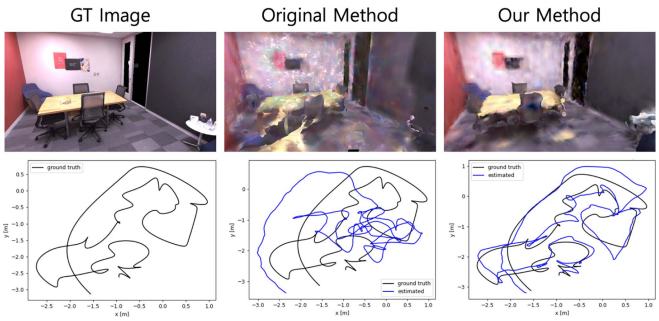


Fig. 5. Mapping and Tracking Visualization Results on the Replica Office2 scene. The figure shows intermediate rendering results during the mapping process, which are not used for training. Our method demonstrates improvements over the original method in both mapping and tracking, even with limited data.

V. CONCLUSION

In this work, we introduced a Bayesian Neural Radiance Field (NeRF) method, significantly advancing 3D scene representation from various viewpoints. Our approach effectively quantifies uncertainty in geometric volume structures without relying on additional networks, making it robust in handling challenging observations. By implementing generalized approximations and defining density-related uncertainty, we have extended the application of Bayesian NeRF beyond RGB images to depth images, achieving enhanced performance across extensive datasets. Furthermore, we validated our method in a SLAM environment, demonstrating that incorporating occupancy uncertainty enhances both mapping and tracking performance. Overall, our proposed extensions to NeRF offer a scalable and efficient solution for managing uncertainty in complex environments, contributing to improved performance in real-world applications with limited data availability.

REFERENCES

[1] M. Abdar et al., "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Inf. Fusion*, vol. 76, pp. 243–297, 2021.
[2] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Zip-NeRF: Anti-aliased grid-based neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 19697–19705.
[3] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1613–1622.
[4] N. Deng et al., "FoV-NeRF: Foveated neural radiance fields for virtual reality," *IEEE Trans. Visual. Comput. Graph.*, vol. 28, no. 11, pp. 3854–3864, Nov. 2022.
[5] L. Goli, C. Reading, S. Selllán, A. Jacobson, and A. Tagliasacchi, "Bayes' Rays: Uncertainty quantification for neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 20061–20070.
[6] W. Hu et al., "Tri-MipRF: Tri-Mip representation for efficient anti-aliasing neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 19774–19783.
[7] X. Hu, G. Xiong, Z. Zang, P. Jia, Y. Han, and J. Ma, "PC-NeRF: Parent-child neural radiance fields using sparse LiDAR frames in autonomous driving environments," *IEEE Trans. Intell. Vehicles*, 2024.
[8] S. Huang et al., "Neural LiDAR fields for novel view synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 18236–18246.
[9] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5580–5590.
[10] X. Kong, S. Liu, M. Taher, and A. J. Davison, "vMAP: Vectorised object mapping for neural field SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 952–961.

[11] J. Korhonen and J. You, "Peak signal-to-noise ratio revisited: Is simple beautiful?," in *Proc. 4th Int. Workshop Qual. Multimedia Experience*, 2012, pp. 37/38.

[12] S. Lee, K. Kang, and H. Yu, "Just flip: Flipped observation generation and optimization for neural radiance fields to cover unobserved view," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 2704–2711.

[13] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3D object," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 9298–9309.

[14] Y. Liu et al., "Visualization of mobility digital twin: Framework design, case study, and future challenges," in *Proc. IEEE 20th Int. Conf. Mobile Ad Hoc Smart Syst.*, 2023, pp. 170–177.

[15] C. F. Lo, "The sum and difference of two lognormal random variables," *J. Appl. Math.*, vol. 2012, no. 1, 2012, Art. no. 838397.

[16] A. Malik, P. Mirdehghan, S. Nousias, K. N. Kutulakos, and D. B. Lindell, "Transient neural radiance fields for LiDAR view synthesis and 3D reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 36, pp. 71569–71581.

[17] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the wild: Neural radiance fields for unconstrained photo collections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7210–7219.

[18] B. Mildenhall, P.P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, Dec. 2021.

[19] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, 2022, Art. no. 102.

[20] X. Pan, Z. Lai, S. Song, and G. Huang, "ActiveNeRF: Learning where to see with uncertainty estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 230–246.

[21] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "DreamFusion: Text-to-3D using 2D diffusion," 2022, *arXiv:2209.14988*.

[22] A. Rosinol, J. J. Leonard, and L. Carlone, "NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields," in *Proc. 2023 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 3437–3444.

[23] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[24] A. Seijas-Macías and A. Oliveira, "An approach to distribution of the product of two normal variables," *Discussiones Mathematicae Probability Statist.*, vol. 32, no. 1/2, pp. 87–99, 2012.

[25] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. 2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 1, pp. 519–528.

[26] J. Shen, A. Agudo, F. Moreno-Noguer, and A. Ruiz, "Conditional-flow NeRF: Accurate 3D modelling with reliable uncertainty quantification," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 540–557.

[27] J. Shen, A. Ruiz, A. Agudo, and F. Moreno-Noguer, "Stochastic neural radiance fields: Quantifying uncertainty in implicit 3D representations," in *Proc. IEEE 2021 Int. Conf. 3D Vis.*, 2021, pp. 972–981.

[28] J. Straub et al., "The replica dataset: A digital replica of indoor spaces," 2019, *arXiv:1906.05797*.

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. 2012 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[30] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6229–6238.

[31] N. Sünderhauf, J. Abou-Chakra, and D. Miller, "Density-aware NeRF ensembles: Quantifying predictive uncertainty in neural radiance fields," in *Proc. 2023 IEEE Int. Conf. Robot. Automat.*, 2023, pp. 9370–9376.

[32] M. Tancik et al., "Block-NeRF: Scalable large scene neural view synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8248–8258.

[33] T. Tao et al., "LiDAR-NeRF: Novel LiDAR view synthesis via neural radiance fields," in *Proc. 32nd ACM Int. Conf. Multimedia*, 2024, pp. 390–398.

[34] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[35] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-NERF: Scalable construction of large-scale NeRFs for virtual fly-throughs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 12922–12931.

[36] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "CLIP-NeRF: Text-and-image driven manipulation of neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3835–3844.

[37] H. Wang, J. Wang, and L. Agapito, "Co-SLAM: Joint coordinate and sparse parametric encodings for neural real-time SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13293–13302.

[38] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.

[39] Z. Wu et al., "MARS: An instance-aware, modular and realistic simulator for autonomous driving," in *Proc. CAAI Int. Conf. Artif. Intell.*, 2023, pp. 3–15.

[40] T. Wutzler, "Functions for the lognormal distribution in R," Accessed: Sep. 19, 2023. [Online]. Available: https://github.com/bgctw/lognorm

[41] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3D voxel patterns for object category recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1903–1911.

[42] J. Yang, M. Pavone, and Y. Wang, "FreeNeRF: Improving few-shot neural rendering with free frequency regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8254–8263.

[43] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation," in *Proc. 2022 IEEE Int. Symp. Mixed Augmented Reality*, 2022, pp. 499–507.

[44] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "PixelNeRF: Neural radiance fields from one or few images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4578–4587.

[45] Y.-J. Yuan, Y.-T. Sun, Y.-K. Lai, Y. Ma, R. Jia, and L. Gao, "NeRF-editing: Geometry editing of neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18353–18364.

[46] J. C. Yue and M. K. Clayton, "A similarity measure based on species proportions," *Commun. Statist.-Theory Methods*, vol. 34, no. 11, pp. 2123–2131, 2005.

[47] J. Zhang, F. Zhang, S. Kuang, and L. Zhang, "NeRF-LiDAR: Generating realistic LiDAR point clouds with neural radiance fields," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 7, pp. 7178–7186.

[48] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.

[49] Z. Zhu et al., "NICE-SLAM: Neural implicit scalable encoding for SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12786–12796.