2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Just Flip: Flipped Observation Generation and Optimization for Neural Radiance Fields to Cover Unobserved View

Sibaek Lee, Kyeongsu Kang and Hyeonwoo Yu

Fig. 1: Overview of our method. (Left) The baseline approach where the robot only observes one side of an object while driving. This case does not yield good rendering results in unobserved views that the robot has not explored. (Right) Our method generates the flipped observations from the actual observations. The robot exploits both input images and flipped images and estimated camera poses to learn 3D space using NeRF for unexplored regions as well. Our method obtains qualified rendering results in unobserved views, even without providing images from unobserved views as a training set.

***Abstract*— With the advent of Neural Radiance Field (NeRF), representing 3D scenes through multiple observations has shown significant improvements. Since this cutting-edge technique can obtain high-resolution renderings by interpolating dense 3D environments, various approaches have been proposed to apply NeRF for the spatial understanding of robot perception. However, previous works are challenging to represent unobserved scenes or views on the unexplored robot trajectory, as these works do not take into account 3D reconstruction without observation information. To overcome this problem, we propose a method to generate flipped observation in order to cover absent observation for unexplored robot trajectory. Our approach involves a data augmentation technique for 3D reconstruction using NeRF, by flipping observed images and estimating the 6DOF poses of the flipped cameras. Furthermore, to ensure the NeRF model operates robustly in general scenarios, we also propose a training method that adjusts the flipped pose and considers the uncertainty in flipped images accordingly. Our technique does not utilize an additional network, making it simple and fast, thus ensuring its suitability for robotic applications where real-time performance is crucial.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is one of the key methods in both the fields of robotics and computer vision. SLAM is the process of generating a 3D map of an unknown environment while simultaneously estimating the position and movement of a robot or camera. This technique is actively researched due to its applications in various real-world scenarios such as autonomous vehicles, unmanned robots, etc [1], [2], [3], [4]. Recently, in computer graphics, a novel technology called Implicit Neural Representation (INR) has emerged that utilizes neural networks to parameterize continuous and differentiable signals. As one of the INR, Neural Radiance Field (NeRF) [5] is a deep learning-based approach for 3D scene representation. In this approach, network learns 3D space by projecting a set of 2D views of a scene into a continuous 3D space, such that the color and density information from each view, thus new views of the scene is interpolated for 2D rendering. This technique can be applied to 3D mapping for SLAM as it can be exploited as a 3D scene representation. It also has several advantages over existing SLAM systems in that it can represent space continuously and is more memory-efficient as it utilizes neural networks to represent space. Due to these features, many recent works [6], [7], [8], [9], [10] have applied NeRF to SLAM and 3D mapping.

However, despite many advantages of NeRF, similar to the previous 3D perception methods it also has limitations in synthesizing unobserved views; from here, we define an unobserved view as a view that the robot has not explored or cannot explore. We also define the unobserved view problem as the task of estimating the synthesized view of an unexplored area. For example, in Fig. 1, the robot explores the orange camera trajectory while observing the lego truck, and the green cameras and trajectory represent unobserved views and the unexplored trajectory respectively. The view

Sibaek Lee, Kyeongsu Kang and Hyeonwoo Yu are with the Department of Intelligent Robots, Sungkyunkwan University, Suwon, South Korea. {lmjlss, thithin0821, hwyu}@skku.edu

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

from the green cameras is unobserved views because they have not been explored yet. These unobserved views cannot be obtained from the observation trajectory due to the 2.5D observation of the robot and the self-occlusions of the objects. To address the crucial issue of obtaining observations of the unexplored region, studies for various explicit spatial representations have also been conducted. [11], [12] utilized an encoder-decoder structure for completing limited 3D point clouds, highlighting the significance of extending our perceptual boundaries beyond current observations. [13] tried to overcome the occlusion problem by capturing visibility patterns of 3D voxel exemplars, but they are learning-based methods which require a massive training dataset. The robot needs to be adaptive in environments where it encounters various objects, so NeRF achieves some advantages compared to the method based on the object dataset [13] while it does not require pre-training in such challenging environments. However, as mentioned above, it is hard for the robot to achieve unobserved views while navigating and thus obtains limited input image data. Therefore, training NeRF with a few images becomes important and various data augmentation methods such as [14], [15] have been studied. Although these methods use geometric approaches to acquire views between or near explored views, they still have limitations in that they do not consider any information about unexplored areas, resulting in the unobserved view problem. Moreover, since they require model training for data augmentation, they may not be suitable for robotics applications that require real-time processing.

To address the unobserved view problem in NeRF, we propose a new unobserved view generation and optimization method. We assume that artificially created objects in indoor and outdoor environments mostly exhibit symmetrical 3D shapes [16], [17], [18]. Following this assumption, [19] presented a method that predicts unobserved views by leveraging a symmetry prior and an additional network. However, these data-driven methods require massive training dataset to learn the symmetric features, which are limited to the robotic application, where fast and robust operation in new environments without additional priors or dataset is crucial. Therefore, we propose a method to predict unobserved views that solely requires the observed sequence, without the need of training dataset involving massive 3D objects and an additional network. To achieve this, we suggest flipping object observation so that we can generate additional significant information of unobserved views. Since training NeRF requires both the images and camera poses, estimating the camera position for the flipped observation image is also imperative. Fortunately, various works [20], [21], [22], [23], [24], [25] that try to train NeRF without camera poses have been introduced. However, these methods cannot guarantee finding the global minimum of the camera poses where the camera pose is completely randomly initialized without additional model. To relax this problem of our method, we propose a method to estimate the 6DOF camera pose of the flipped image using the camera pose of the input image. However, apparently this flip and estimation strategy is not effective for complex objects. To catch symmetric parts and dismiss dissymmetric region, we refine the flipped camera pose and employ a Bayesian approach that incorporates the uncertainty of the flipped image. Therefore we show that it is possible to exploit the symmetric parts of the given 3D object and ignore the unnecessary part according to the estimated uncertainty, thereby we have the robust unobserved view prediction approach.

In summary, we propose a flipped observation generation and training method by considering robot trajectory to improve understanding of unobserved views Our contribution is as follows:

- We introduce the flip method for predicting unobserved views and propose an approach for the initial 6DOF pose estimation of the flipped images.
- Recognizing that the initialized flipped camera pose can be noisy depending on the object's shape, we propose a learning approach that performs bundle adjustment on the flipped pose. This method simultaneously optimizes the camera pose and network parameters.
- Flipped images also inherently have uncertainties depending on the object. To address this, we use Bayesian approach that integrates these uncertainties into the model's loss during the learning process.

II. Related work

In the traditional SLAM techniques, 3D representation was achieved using methods such as mesh [26], [27], point clouds [28], [29], and depth maps [30], [31]. However, these approaches have the drawback of having memory limitations since they require storing discrete information, resulting in the map being represented in a sparse manner. To address these limitations, recent research [6], [7], [8], [9], [10] has applied NeRF to SLAM for 3D mapping. NeRF defined as a continuous function by a neural network that takes in a spatial location and viewing direction as inputs and outputs the RGB values and volume density. By continuous neural network function, their approach has the advantages of high resolution and low memory. NeRF has shown remarkable performance in synthesizing photorealistic novel views of real-world scenes or objects. Afterward, NeRF have been extended to achieve fast training [32], [33], [34], few input data [35], [36], large scale scene [37], [38], [39], dynamic scene capture [40], [41], scene editing [42], [43].

In neural radiance fields, we attempt to effectively predict unobserved views in a few-shot setting for robots by estimating the camera pose and uncertainty associated with flipped images. Since the flipped image does not have a camera pose, NeRF needs to be trained without camera pose or simultaneously estimate the camera pose and neural radiance fields. This challenge has been tackled in some works. [20] implemented pose estimation by inverting a neural radiance field, while [21] adopted a two-stage strategy optimizing both camera parameters and the radiance field. With the use of the SIREN layer, [24] introduced sampling methods to counter the joint optimization in NeRF. However,

they can only optimize camera pose for relatively short camera trajectories. [22] can reconstruct neural radiance fields and estimate camera poses using generative model. [23] applies bundle adjustment, and [25] utilizes monocular depth estimation, have also been explored. Amidst this backdrop and given our approach to flipped images, the task of addressing inherent uncertainty becomes crucial. There are several prior studies that have incorporated uncertainty into NeRF. Specifically, [44] introduced the concept of uncertainty to render complex scenes from unstructured images. In this approach, they derive uncertainty from transient networks and consider it in the pixel levels of the images. [45] produce uncertainty estimates by modeling a distribution over all the possible radiance fields modeling the scene. They represent the distribution of all potential radiance fields for scene modeling using variational inference. [46] by the same authors address the limitations of [45] by integrating a conditional normalizing flow with latent variable modeling. Based on these methods, we introduce a flip data generation and training method within NeRF to tackle the unobserved view problem.

III. Preliminary

NeRF is a deep learning-based approach for 3D scene reconstruction from 2D images. It aims to model a 3D scene as a continuous function that maps a 3D point in space to its color and density in the image plane. This function is represented as a neural network, which is trained on a set of input images $\mathcal{I} = \{I_1, I_2, ..., I_N\}$ of a scene, with their associated camera parameters $\Pi = \{\pi_1, \pi_2, ..., \pi_N\}$. NeRF model is to learn a mapping of image radiance $\mathbf{c} = (r, g, b)$ and spatial density σ for 3D coordinates $\mathcal{X} = \{\mathbf{x}\}$ at viewing direction $\mathbf{d} = (\theta, \phi)$. This mapping is represented by a neural network and the mapping function can be represented mathematically as $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$, where Θ is network parameters.

In order to obtain the rendered image $\hat{I}_i$, the color at each pixel $\mathbf{p} = (u, v)$ is estimated by a rendering function $\mathcal{R}$. The expected image color $\hat{I}_i(\mathbf{p})$ of camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ that starts from camera origin $\mathbf{o}$ with near and far bound t_n and t_f can be written as:

$$\begin{aligned}\hat{I}_i(\mathbf{p}) &= \mathcal{R}(\mathbf{p}, \pi_i|\Theta) \\ &= \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \end{aligned} \tag{1}$$

where

$$T(t) = exp(-\int_{t_n}^{t} \sigma(r(s))ds). \tag{2}$$

$T(t)$ denotes the accumulated transmittance along the ray from t_n to t, *i.e.*, the probability that the ray travels from t_n to t without hitting any other particle. Therefore, the network can be trained by minimizing the difference between the predicted (or rendered) image $\hat{I}$ and the ground truth input image I. This difference can be defined as photometric loss L_p:

$$L_p(\Theta, \Pi) = \sum_i^N \|I_i - \hat{I}_i\|_2^2. \tag{3}$$

IV. Approach

A. NeRF with Camera Pose

1) Initial Camera Pose: Symmetry is prevalent in real-world objects due to its functional and aesthetic benefits, and can be manufactured easily in some cases [16], [17], [18]. Based on this premise, our idea starts flipping the input images $\{I\}$ in order to predict unobserved views for the unexplored area. Then Eqn. (3) can be written as follows:

$$L_p(\Theta, \Pi^{all}) = \sum_i^N \|I_i - \hat{I}_i\|_2^2 + \sum_i^N \|I'_i - \hat{I}'_i\|_2^2, \tag{4}$$

where $\{I'_i, ..., I'_n\} = \mathcal{I}'$ are flipped images and Π^{all} is a union set of the existing camera poses $\Pi = \{\pi_i, ..., \pi_n\}$ of $\mathcal{I}$ and camera poses $\Pi' = \{\pi'_i, ..., \pi'_n\}$ for $\mathcal{I}'$. In this case, a challenge arises since the flipped images $\{I'\}$ for the unobserved views do not possess defined camera poses. To relax this issue, approaches such as [47], [23] based on Bundle adjustment (BA) can be adopted. However, those BA-based approaches require a larger number of images than those acquired from scenarios such as exploration or navigation, or require a nice initial guess for the camera poses. Thus, it is not suitable to directly apply those approaches.

Thus, we propose a method to estimate the initial camera pose for the flipped images $\{I'\}$ by leveraging the existing camera pose and geometric constraints. This method commences by using the least squares approach to identify the optimal sphere that passes through the given input camera pose. The general equation of a sphere with its center at x_0, y_0, z_0 and radius r is represented by $x^2 + y^2 + z^2 = 2xx_0 + 2yy_0 + 2zz_0 + r^2 - x_0^2 - y_0^2 - z_0^2$. Using input camera poses, we can express the equation of a sphere in matrix form as $\vec{f} = A\vec{c}$, where $\vec{c}$ contains information regarding the radius and the center coordinates of the sphere. Through the matrix equation, we can find the value of $\vec{c}'$ that minimizes the norm of the residual as the following:

$$\vec{c}' = \underset{\vec{c}}{\operatorname{argmin}}(A\vec{c} - \vec{f})^T(A\vec{c} - \vec{f}).$$

Once the optimized sphere is obtained, the input camera pose coordinates x, y, z undergo a symmetrical transformation through the symmetric plane that intersects the center of the sphere. Subsequently, leveraging the transformed camera coordinates $\mathbf{c} = (x', y', z')$, the sphere's center point $\mathbf{at} = (x_0, y_0, z_0)$, and the up vector $\mathbf{up} = (0, 0, 1)$, we can compute the rotation matrix R. Naturally, transformed camera pose corresponds to the translation vector T and by integrating it with the rotation matrix R, we can derive the estimated initial 6DOF camera pose π'. With the assistance of methods such as [48], [19], we can further refine our 6DOF camera pose estimation.

2) Camera Pose and BA-NeRF: It is important to note that assuming object symmetry introduces a significant constraint; if the object is not symmetrical, the flipped images and their inferred camera poses can destabilize the NeRF model. To mitigate this, we perform BA-based NeRF [23] with estimated initial camera poses for refining flipped camera poses and learning 3D space simultaneously.

When training the model with the flipped images' camera poses initialized randomly, the learning process tends to proceed in an incorrect direction. However, by using our 6DOF pose estimation, it becomes suitable for BA-based NeRF, which requires moderately estimated camera poses. Considering our confidence in the initial input pose, the training strategy we adopted ensures that only the flipped pose undergoes bundle adjustment, while the original remains unaffected. We can formulate homogeneous coordinates of pixel coordinates as $\overline{\mathbf{p}} = [\mathbf{p}; 1]^T$. Then, we can represent a 3D point $\mathbf{x}$ on the viewing ray at depth z by using the equation $\mathbf{x} = z\overline{\mathbf{p}}$. Given a camera pose π', 3D point $\mathbf{x}$, within the camera's view space undergoes a transformation to align with the world coordinates using a rigid 3D transformation $\mathcal{W}$. Consequently, the RGB value rendered at a particular pixel, intrinsically dependent on the camera pose, is illustrated as the following:

$$\hat{I}'(\mathbf{p}; \pi') = \mathcal{R}\left(F_\Theta\left(\mathcal{W}\left(z_1\overline{\mathbf{p}}; \pi'\right)\right), \ldots, F_\Theta\left(\mathcal{W}\left(z_N\overline{\mathbf{p}}; \pi'\right)\right)\right).$$

Here, our goal is to optimize flipped camera poses $\Pi' = \{\pi'_i, ..., \pi'_n\}$ and network parameters Θ. Consequently, using Eqn. (4) we can have the optimal solution as:

$$\Pi'^{opt}, \Theta^{opt} = \underset{\Pi', \Theta}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{\mathbf{p}} \left\| I'_i(\mathbf{p}) - \hat{I}'\left(\mathbf{p}; \pi'_i, \Theta\right) \right\|_2^2.$$

Given the inherent non-linearity of the optimization problem, we employed a gradient-based optimization approach. We implemented the Lucas-Kanade algorithm used in Optical Flow and leveraging the Jacobian, conducted backpropagation.

B. NeRF with uncertainty

We started our approach based on the assumption of object symmetry, attempting to predict unobserved views by flipping images. However, not all objects exhibit symmetry, and depending on the viewing angle of an object, a simple flip can result in performance decrease. We handle this issue in the aspect of camera pose estimation, but similar challenge occurs in images as well. Since various objects are not perfectly symmetric, in Eqn. (1) some of rays $\mathbf{r}$ gathered from the flipped images I' gives us significant errors while training NeRF. To ensure robust predictions of unobserved views by ignoring wrong pixels and exploiting correct pixels from I', we propose a training method that takes into account the uncertainty associated with flipped images. Following similar approach to the bundle adjustment on the flipped pose, we consider uncertainty values solely for the flipped images.

To address the uncertainty of the predicted color value, we follow Bayesian framework [49]. To compute the loss for observed images I in Eqn. (4),

$$L(\mathbf{r}) = \left\| \mathbf{C}_i(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r}) \right\|_2^2 \tag{5}$$

In Eqn. (5) all rays $\mathbf{r}$ in I are fully exploited for training NeRF, which matters in our case, since 3D objects are not perfectly symmetric thereby some of the rays $\mathbf{r}'$ from I' are not matched to the 3D shape. Instead of representing the ray (or color value) from any given point in the scene as a singular value, we propose modeling it using a Gaussian distribution for the observation uncertainty. This predicted variance can be interpreted as an indicator of the data's uncertainty for a specific location. Consequently, the mapping function can be represented as $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\beta^2, \mathbf{c}, \sigma)$, where β^2 is the variance of color (uncertainty). To validate the variance value, the Softplus function $\bar{\beta^2} = \beta^2_{min} + \log(1 + exp(\beta^2(\mathbf{r}(t))))$ was adopted. Contrary to the flipped images, the original input image set the value of $\bar{\beta^2}$ to β^2_{min}. This approach compels the model to exhibit heightened uncertainty in unobserved regions, thereby facilitating effective predictions of unobserved views. Based on Eqn. (1), the rendering function $\mathcal{R}$, being a linear combination of sampled points, leads the rendered color value $\hat{C}(\mathbf{r})$ to follow a Gaussian distribution. Same way, we can derive the variance of rendered color $\hat{\beta}^2(\mathbf{r})$. Building on this, the loss for the ray $\mathbf{r}'$ in flipped image I', taking into account uncertainty, is defined as follows:

$$L(\mathbf{r}') = \frac{\left\| \mathbf{C}_i(\mathbf{r}') - \hat{\mathbf{C}}(\mathbf{r}') \right\|_2^2}{2\hat{\beta}^2(\mathbf{r})} + \frac{\log \hat{\beta}^2(\mathbf{r}')}{2} + \frac{w}{N_s} \sum_{j=1}^{N_s} \sigma\left(t_j\right).$$

The above cost function aims to minimize the negative log-likelihood, with the final term serving as a regularization term to prevent blurring effects.

We initialized the flipped camera pose using a geometric approach. Given the unreliability of the flipped information, we refined the camera pose, taking into account the uncertainty associated with the image pixels. Our method does not require an additional network, making it both efficient and rapid, and particularly apt for robotics applications. In the subsequent section, we compare our proposed method with the baseline method.

V. Experiments

A. Implementation and Setup

Given that we possess the ground-truth for unobserved views, we used the NeRF synthetic dataset [5] and Modelnet dataset [50] for experiments. This dataset comprises objects that exhibit a certain degree of structural symmetry and those without such symmetrical properties. Neither of these cases has the symmetry of lighting and color. Considering the input data scenario, as shown in Fig. 1 we assume that the robot is limited to observing only partial side of the object. In alignment with real-world robot applications, we trained with

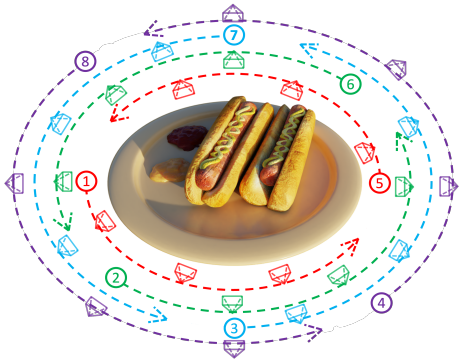

Fig. 2: Dataset configurations. To generalize performance across different views, we generate a total of eight dataset configurations for a single data scene. For each configuration, we selected 16 evenly spaced images by rotating around an object. The n^{th} dataset configuration encompasses images from the $(2n-1)^{th}$ to the $(2n+6)^{th}$. Through this setup, we demonstrate our method can effectively predict unobserved views, not just in specific views but in more general ones.

a limited image set, i.e., in our case eight images. Since flipping an image can lead to differing performance depending on the observation view, we considered it essential to ensure a comprehensive evaluation of general performance. Therefore, we constructed a total of eight different input dataset configurations, as illustrated in Fig. 2. We compared on five distinct methods (baseline, baseline + uncertainty, flip, flip + uncertainty, upper bound), discussed in detail in the analysis. For simplicity in all our experiments, we used a single network that samples 128 points. Considering that uncertainty during training can lead to instability, thus we perform conventional NeRF training up to 10k epochs and apply Bayesian approach up to 50k epochs subsequently. The remaining configurations were kept consistent with the vanilla NeRF setup, all experiments were conducted on NVIDIA RTX 4090 GPU. As our approach strives to predict the view that cannot be observed, we utilize the unobserved view from the opposite side as the validation and ground-truth image. For quantitative comparison, we used three metrics: Peak Signal-to-Noise Ratio (PSNR) [51], Structural Similarity Index Measure (SSIM) [52], and Learned Perceptual Image Patch Similarity (LPIPS) [53].

B. Analysis of the experiment results

First, we provide a description of the five methods in detail. Our dataset consists of four sets: the input image set obtained from the robot exploration, the flipped image set, the test image set, and the upper image set for the unexplored scene. Each set contains eight images, and we assume that the corresponding camera poses are already known except for the camera pose of the flipped image. The input image set consists of data that the robot has acquired while navigating, whereas the flipped image set is created by flipping the input images. The test set comprises unobserved views that the robot cannot explore. We construct the upper data by selecting views that are closest to those in the test data. We summarize the five methods in Table. I. (a) *baseline (B/L)* is trained using the image set obtained while the robot is navigating. (b) *B/L + uncertainty (U)* is applied uncertainty to all images used in the baseline. (c) *Flip* leverages both the input image and its flipped version, and utilizes the least squares method to estimate the flipped camera pose for training. Moreover, it refines the pose of the flipped image through bundle adjustment. Our proposed method, (d) *Flip + U*, extends (c) *Flip* by additionally considering the uncertainty of the flipped image. Meanwhile, (e) *Upper* method trains the model using both the input image, upper image, and corresponding upper camera pose. This approach is an upper bound since it directly uses the image of the unobserved view in training. To examine each element of our proposed method, we conducted an ablation study and show the results in Table. II. The results indicated that applying both bundle adjustment and uncertainty to the flipped image yielded the best performance.

In TABLE III, we display the metric results of the five methods. Naturally, for objects with inherent structural symmetry, flipping the image from certain views can effectively predict unobserved views. To validate performance in general settings, we performed experiments on eight dataset configurations as depicted in 2, and averaged the results. Our primary goal is to predict unobserved views. Therefore, (a) *B/L*, which relies solely on one side of an object, exhibits significantly low performance. As shown in Fig. 3, certain portions disappear, and in specific views, the resulting image is entirely blank. (b) *B/L + U*, which considers the uncertainty of the input image, demonstrates performance improvements even without flipping the image. However, as evidenced in the qualitative results, (b) *B/L + U* has an issue with black noise appearing in unobserved views. (c) *Flip*, on the other hand, does show an enhancement in performance compared to (c) *B/L* and effectively addresses the black noise issue. Yet, for non-symmetric objects, its performance lags behind that of (b) *B/L + U*. Furthermore, as demonstrated in the lego scene of 3, configurations that observe only the front face of an object do not always benefit from the flip method. Contrasting this with the chair scene, where the side of the object is observed, there are instances where the unobserved views aren't accurately predicted despite flipping.

TABLE I: Summary of the five experimental methods.

(a) *(B/L)*	input image
(b) *(B/L) + U*	input image + consider ucert
(c) *Flip*	input image + flipped and refine pose
(d) *Flip + U*	input image + flipped and refine pose + consider ucert
(e) *Upper*	input image + ground-truth image

TABLE II: ablation study. We report PSNR, SSIM, and LPIPS of the full model(the last row) and three configurations by removing the application of bundle adjustment, uncertainty on the flip image and both, respectively.

bundle adjustment	uncertainty	PSNR↑	SSIM↑	LPIPS↓
-	-	16.636	0.787	0.232
✓	-	17.205	0.789	0.230
-	✓	16.694	0.790	0.240
✓	✓	**17.620**	**0.794**	**0.229**

TABLE III: Quantitative comparisons of our flip methods with baseline method and upper bound method on the NeRF synthetic dataset [5]. The upper image is a case where an image observed from a viewpoint that the robot has not explored is added to the image used at the baseline. The experiments, as depicted in 2, were conducted using a total of 8 input dataset configurations for each scene, and the metric value averaged out. (* B/L mean baseline and U mean uncertainty)

Scene	PSNR ↑					SSIM ↑					LPIPS ↓				
	B/L	*B/L+U*	*Flip*	*Flip+U*	*Upper*	*B/L*	*B/L+U*	*Flip*	*Flip+U*	*Upper*	*B/L*	*B/L+U*	*Flip*	*Flip+U*	*Upper*
Chairs	14.07	16.46	19.69	**19.86**	26.96	0.75	0.78	0.87	**0.88**	0.92	0.48	0.28	0.13	**0.12**	0.05
Lego	11.57	14.85	15.24	**17.79**	23.91	0.65	0.67	0.74	**0.78**	0.87	0.50	0.34	0.23	**0.20**	0.11
Materials	9.66	16.74	17.29	**19.14**	25.60	0.66	0.75	0.81	**0.83**	0.90	0.50	0.31	0.18	**0.14**	0.09
Ship	6.73	16.97	18.29	**18.70**	24.77	0.55	0.65	0.71	**0.72**	0.80	0.61	0.40	0.27	**0.26**	0.18
Drum	10.24	11.91	11.90	**13.77**	20.19	0.60	0.59	0.63	**0.68**	0.83	0.55	0.43	0.47	**0.43**	0.16
Ficus	11.20	15.59	17.15	**18.02**	22.59	0.72	0.78	0.81	**0.82**	0.88	0.33	0.22	0.15	**0.13**	0.10
Hotdog	13.67	**19.07**	18.91	18.92	28.30	0.70	0.83	0.84	**0.84**	0.93	0.44	0.22	**0.19**	0.21	0.07
Mic	12.89	14.16	13.78	**14.30**	24.93	0.82	0.78	**0.82**	0.80	0.93	0.46	0.30	0.38	**0.30**	0.07
Mean	11.25	15.71	16.53	**17.56**	24.65	0.68	0.72	0.77	**0.79**	0.88	0.48	0.30	0.25	**0.22**	0.10

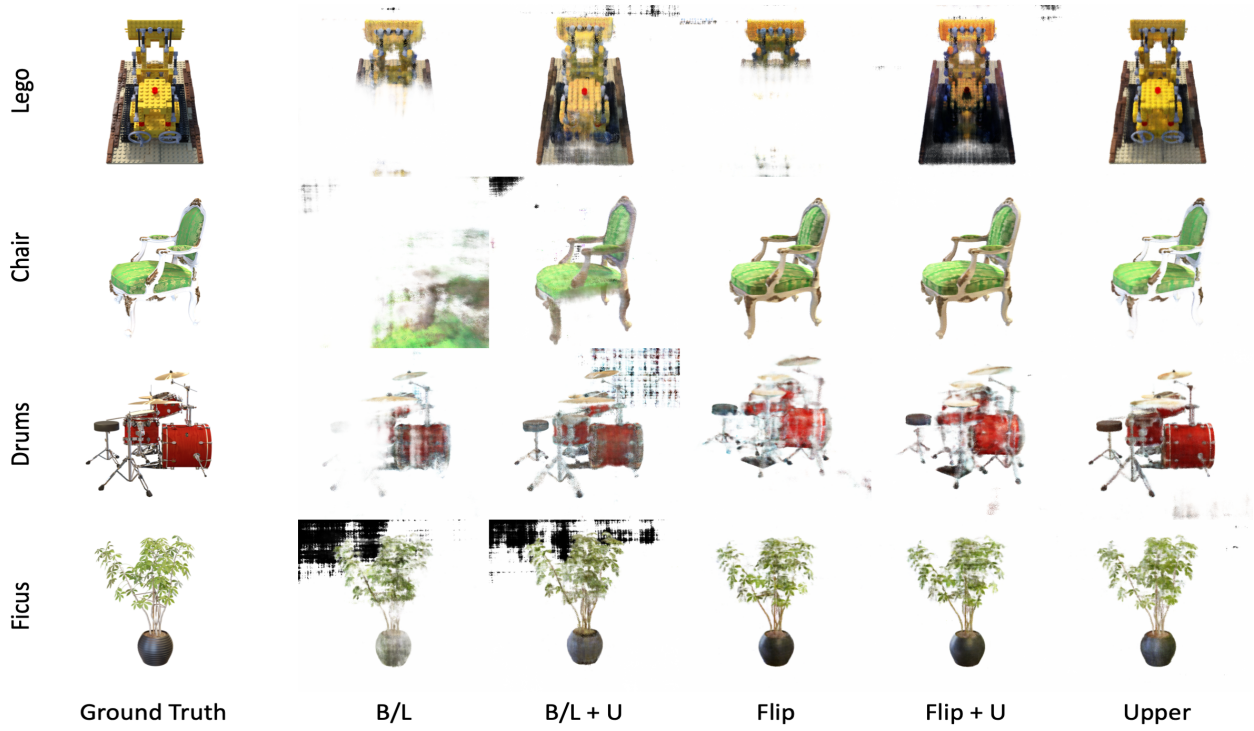

Fig. 3: Qualitative comparisons on the NeRF synthetic dataset. Our method captures the object structure of unobserved views effectively, offering a more accurate image than other methods while maintaining a noise-free output.

Predicting unobserved views by flipping images is based on utilizing the 3D structural features of objects. To demonstrate the efficacy of our approach, we additionally conducted experiments on the ModelNet dataset. Since ModelNet is a structurally featured dataset without color, flipping images and conducting bundle adjustment can be examined effectively. We arbitrary choose two instances per class, total 80 instances. We set the dataset to follow the configuration of NeRF synthetic dataset 2 and conduct experiments solely for training depth information. TABLE 4 reveals that while (a) *B/L* and (b) *B/L + U* experienced noise-induced blurring, methods (c) *Flip* and (d) *Flip + U* resolved this issue by optimizing camera poses after flipping images. Additionally, considering uncertainty has demonstrated performance improvements across all scenes, finely capturing the edges of objects. Consequently, by integrating the strengths of both (b) *B/L + U* and (c) *Flip*, our method (d) *Flip + U* significantly enhances performance, regardless of object symmetry.

VI. Conclusion

In this paper, we introduced a flipped observation generation method for NeRF to predict unobserved views. Most of the robot exploration is performed in real-time, thus it is challenging to observe all the possible viewpoints. To predict unobserved views, we propose a method that exploits the observed views by image flipping. Given that flipped images do not have camera poses, we also proposed a method for estimating the 6DOF pose of flipped images. Furthermore, we applied bundle adjustment to optimize these estimated poses in conjunction with the model parameters. Since only the observed parts should be exploited in order to have robust unobserved view prediction, for the flipped

TABLE IV: Quantitative comparisons of five methods on the Modelnet dataset [50]. We conducted our experiments using the same setup as in the NeRF synthetic dataset in 2. Our method demonstrates performance improvement in predicting unobserved views, even without utilizing areas the robot has not explored for training.

Scene	PSNR ↑					SSIM ↑					LPIPS ↓				
	B/L	*B/L+U*	*Flip*	*Flip+U*	*Upper*	*B/L*	*B/L+U*	*Flip*	*Flip+U*	*Upper*	*B/L*	*B/L+U*	*Flip*	*Flip+U*	*Upper*
Airplane	24.55	28.36	30.70	**31.46**	33.65	0.91	0.93	0.94	**0.96**	0.96	0.32	0.24	0.21	**0.17**	0.14
Bench	19.26	21.39	24.85	**26.50**	32.83	0.81	0.83	0.88	**0.90**	0.94	0.47	0.40	0.31	**0.29**	0.21
Car	23.35	25.12	29.42	**31.51**	36.88	0.90	0.91	0.94	**0.95**	0.96	0.32	0.28	0.23	**0.19**	0.13
Chair	16.72	19.30	33.34	**35.57**	36.10	0.71	0.83	0.93	**0.96**	0.97	0.51	0.40	0.18	**0.14**	0.12
Door	22.69	26.08	32.14	**35.42**	37.94	0.87	0.89	0.94	**0.95**	0.96	0.34	0.27	0.15	**0.10**	0.08
Flower	19.06	22.73	26.43	**29.79**	34.66	0.80	0.84	0.91	**0.93**	0.96	0.48	0.39	0.26	**0.13**	0.11
Guitar	19.82	23.62	24.32	**25.12**	36.71	0.87	0.89	0.93	**0.96**	0.97	0.42	0.36	0.29	**0.16**	0.07
Stair	18.216	21.67	26.75	**30.04**	35.07	0.83	0.90	0.93	**0.96**	0.97	0.40	0.33	0.24	**0.14**	0.12
Table	17.32	19.00	30.10	**32.29**	34.68	0.72	0.84	0.94	**0.95**	0.96	0.49	0.45	0.29	**0.24**	0.16
Person	18.76	25.99	32.49	**33.22**	36.47	0.73	0.89	0.94	**0.95**	0.97	0.48	0.28	0.19	**0.13**	0.10
Mean	19.98	23.33	29.06	**31.10**	35.50	0.82	0.86	0.93	**0.95**	0.97	0.42	0.34	0.24	**0.17**	0.12

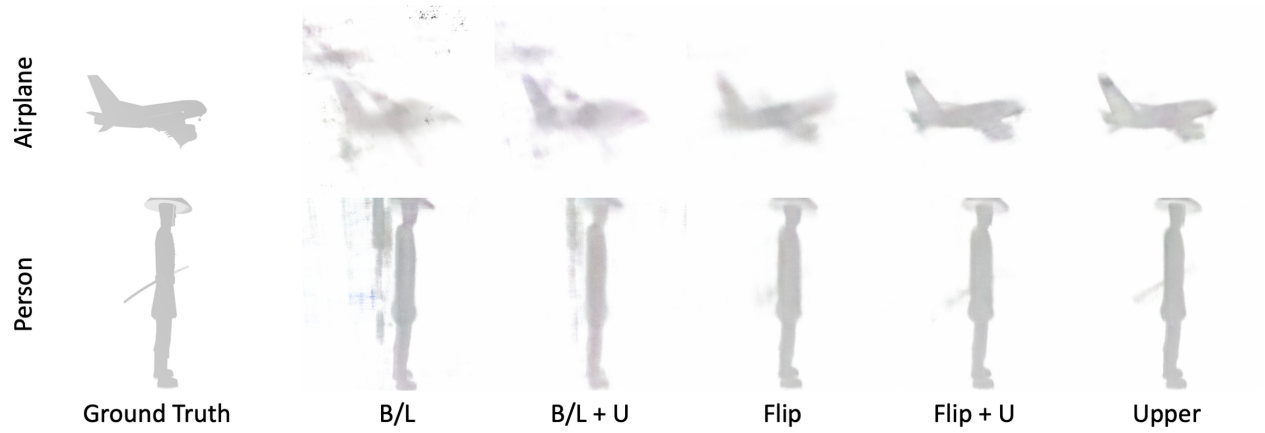

Fig. 4: Qualitative comparisons on the ModelNet dataset reveal significant performance improvement when images are flipped and camera poses optimized through bundle adjustment. The ModelNet dataset, lacking color information and consisting solely of structural data, benefits greatly from this approach, effectively addressing blurring issues and enhancing overall performance.

images, we incorporated Bayesian approach by considering the uncertainty estimation. From experimental results we show that our approach markedly improves performance in predicting scenes from unobserved viewpoints and shows competitive results compared to the traditional approaches. Our flipped observation method is suitable for robotic applications such as robot exploration and navigation where real-time is important as our approach is simple yet robust.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant (no. RS-2024-00359937).

REFERENCES

[1] M. Filipenko and I. Afanasyev, "Comparison of various slam systems for mobile robot in an indoor environment," in *2018 International Conference on Intelligent Systems (IS)*. IEEE, 2018, pp. 400–407.

[2] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[3] H. Yu, J. Moon, and B. Lee, "A variational observation model of 3d object for probabilistic semantic slam," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5866–5872.

[4] M. Mittal, R. Mohan, W. Burgard, and A. Valada, "Vision-based autonomous uav navigation and landing for urban search and rescue," in *Robotics Research: The 19th International Symposium ISRR*. Springer, 2022, pp. 575–592.

[5] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[6] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[7] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.

[8] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," *arXiv preprint arXiv:2210.13641*, 2022.

[9] 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 *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 499–507.

[10] X. Kong, S. Liu, M. Taher, and A. J. Davison, "vmap: Vectorised object mapping for neural field slam," *arXiv preprint arXiv:2302.01838*, 2023.

[11] H. Son and Y. M. Kim, "Saum: Symmetry-aware upsampling module for consistent point cloud completion," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[12] C. Ma, Y. Chen, P. Guo, J. Guo, C. Wang, and Y. Guo, "Symmetric shape-preserving autoencoder for unsupervised real scene point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 560–13 569.

[13] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3d voxel patterns for object category recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1903–1911.

[14] D. Chen, Y. Liu, L. Huang, B. Wang, and P. Pan, "Geoaug: Data augmentation for few-shot nerf with geometry constraints," in *Com-*

puter Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII. Springer, 2022, pp. 322–337.

[15] M. Bortolon, A. Del Bue, and F. Poiesi, "Data augmentation for nerf: a geometric consistent solution based on view morphing," *arXiv preprint arXiv:2210.04214*, 2022.

[16] S. Tulsiani, A. Kar, Q. Huang, J. Carreira, and J. Malik, "Shape and symmetry induction for 3d objects," *arXiv preprint arXiv:1511.07845*, 2015.

[17] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan, "Symmetry in 3d geometry: Extraction and applications," in *Computer Graphics Forum*, vol. 32, no. 6. Wiley Online Library, 2013, pp. 1–23.

[18] S. Wu, C. Rupprecht, and A. Vedaldi, "Unsupervised learning of probably symmetric deformable 3d objects from images in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1–10.

[19] X. Li, C. Hong, Y. Wang, Z. Cao, K. Xian, and G. Lin, "Symmnerf: Learning to explore symmetry prior for single-view view synthesis," in *Proceedings of the Asian conference on computer vision*, 2022, pp. 1726–1742.

[20] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1323–1330.

[21] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "Nerf–: Neural radiance fields without known camera parameters," *arXiv preprint arXiv:2102.07064*, 2021.

[22] Q. Meng, A. Chen, H. Luo, M. Wu, H. Su, L. Xu, X. He, and J. Yu, "Gnerf: Gan-based neural radiance field without posed camera," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6351–6361.

[23] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5741–5751.

[24] D. Xu, Y. Jiang, P. Wang, Z. Fan, H. Shi, and Z. Wang, "Sinnerf: Training neural radiance fields on complex scenes from a single image," in *European Conference on Computer Vision*. Springer, 2022, pp. 736–753.

[25] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, "Nope-nerf: Optimising neural radiance field with no pose prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4160–4169.

[26] G. Riegler and V. Koltun, "Free view synthesis," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*. Springer, 2020, pp. 623–640.

[27] ——, "Stable view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 216–12 225.

[28] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[29] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," in *International conference on machine learning*. PMLR, 2018, pp. 40–49.

[30] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2024–2039, 2015.

[31] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2821–2830.

[32] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 124–14 133.

[33] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5438–5448.

[34] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[35] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.

[36] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 882–12 891.

[37] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, "Citynerf: Building nerf at city scale," *arXiv preprint arXiv:2112.05504*, 2021.

[38] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.

[39] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 922–12 931.

[40] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5865–5874.

[41] G. Yang, M. Vo, N. Neverova, D. Ramanan, A. Vedaldi, and H. Joo, "Banmo: Building animatable 3d neural models from many casual videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2863–2873.

[42] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3835–3844.

[43] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 353–18 364.

[44] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. 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*, 2021, pp. 7210–7219.

[45] J. Shen, A. Ruiz, A. Agudo, and F. Moreno-Noguer, "Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 972–981.

[46] J. Shen, A. Agudo, F. Moreno-Noguer, and A. Ruiz, "Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification," in *European Conference on Computer Vision*. Springer, 2022, pp. 540–557.

[47] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[48] Y. Zhou, S. Liu, and Y. Ma, "Nerd: Neural 3d reflection symmetry detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 940–15 949.

[49] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017.

[50] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[51] J. Korhonen and J. You, "Peak signal-to-noise ratio revisited: Is simple beautiful?" in *2012 Fourth International Workshop on Quality of Multimedia Experience*. IEEE, 2012, pp. 37–38.

[52] J. C. Yue and M. K. Clayton, "A similarity measure based on species proportions," *Communications in Statistics-theory and Methods*, vol. 34, no. 11, pp. 2123–2131, 2005.

[53] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.