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

ActNeRF: Uncertainty-aware Active Learning of NeRF-based Object Models for Robot Manipulators using Visual and Re-orientation Actions

Saptarshi Dasgupta*, Akshat Gupta*, Shreshth Tuli and Rohan Paul
Indian Institute of Technology Delhi, India {*} denotes equal contribution

***Abstract*— Manipulating unseen objects is challenging without a 3D representation, as objects generally have occluded surfaces. This requires physical interaction with objects to build their internal representations. This paper presents an approach that enables a robot to rapidly learn the complete 3D model of a given object for manipulation in unfamiliar orientations. We use an ensemble of partially constructed NeRF models to quantify model uncertainty to determine the next action (a visual or re-orientation action) by optimizing informativeness and feasibility. Further, our approach determines *when* and *how* to grasp and re-orient an object given its partial NeRF model and re-estimates the object pose to rectify misalignments introduced during the interaction. Experiments with a simulated Franka Emika Robot Manipulator operating in a tabletop environment with benchmark objects demonstrate an improvement of (i) 14% in visual reconstruction quality (PSNR), (ii) 20% in the geometric/depth reconstruction of the object surface (F-score) and (iii) 71% in the task success rate of manipulating objects *a-priori* unseen orientations/stable configurations in the scene; over current methods. The project page can be found at https://actnerf.github.io/**

I. Introduction

We consider the problem of acquiring a 3D visual and geometric representation of an object for sequential robot manipulation tasks. In recent years, Neural Radiance Fields (NeRF) has emerged as a useful implicit representation that allows synthesis of novel views aiding in downstream planning, manipulation, and pose estimation tasks. Such a representation is acquired by collecting a set of views from known poses in the environment. The process for collecting such views is either in (i) batch mode [1]–[3] by exhaustively collecting observations covering a region or (ii) actively by determining a set of informative views [4], [5]. While effective in quickly constructing object models, these approaches can only reconstruct visible regions, leaving obscured parts like the base, internal contents, and other occluded areas unmodelled. The inability to accurately model the object owing to occlusions in the scene translates to poor manipulation ability for subsequent manipulation tasks.

This work considers the possibility of *directly interacting* via grasping, re-orientation, and releasing the object to expose previously *unexposed* regions. Fig. 1 presents an overview of our model acquisition technique. Introducing physical interaction during model acquisition poses two key challenges. First, finding stable grasping points using a *partially* built model is challenging due to depth uncertainty in unobserved or poorly observed regions. Second, re-orientation introduces uncertainty in the object's pose, affecting the fusion of the radiance field arising from new

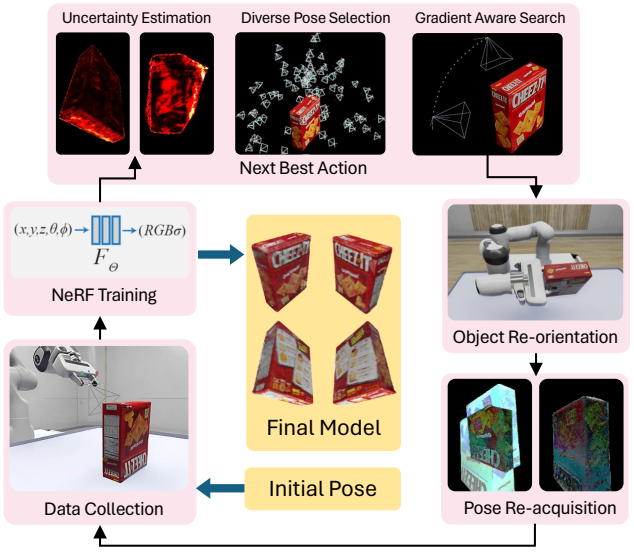


Fig. 1: **Overview.** We present an active learning approach for building a NeRF model of an object that allows the robot to re-orient the object while collecting visual observations. At each iteration, we determine the next best action, perform the action on the object, apply pose-reacquisition to correct for any stochastic results, capture images and train NeRF model which is again used in the next iteration.

observations. Further, as opposed to scene-based representations, we seek the ability to acquire object-centric radiance fields to support semantic tasks that may require sequential manipulation actions (e.g., clearing objects from a region).

Overall, this paper makes the following contributions:

- Leveraging vision foundation models to isolate the object of interest to disentangle its uncertainty from that of other background objects in the scene.
- A search procedure that estimates the next most informative action (visual or re-orientation). The procedure relies on a coarse-to-fine optimization of the continuous viewing space incorporating (i) model uncertainty in the partially-built model (adapting [5]), (ii) motion costs, and (iii) kinematic constraints.
- An approach for grasping while accounting for the uncertainty in the partially constructed model and re-estimating the pose of the object after interaction for fusing the incrementally acquired model.

Extensive evaluation with a simulated robot manipulator with benchmark objects shows improvements in the coverage and visual/geometric quality of the acquired model. Overall, this work takes a step in the direction of acquiring a rich NeRF model of an object to support future robot manipulation tasks such as pick/place from arbitrary object configurations.

979-8-3503-7769-9/24/$31.00 ©2024 IEEE

II. Related Work

NeRF-based [1] representations have been used in many robotics problems. DexNerf [6], and EvoNeRF [7] use NeRFs for modeling transparent objects that are difficult to represent with voxel-based methods. Adamkiewicz et al. [8] uses NeRFs to model the environment and synthesize trajectories for a quadrotor, while Driess et al. [9] use NeRF for representing multi-object scenes and train graph neural networks to learn dynamics models. While the aforementioned approaches utilize NeRF models for robotic tasks, they do not directly address the problem of determining the optimal viewpoints for constructing said NeRF models.

The concept of actively constructing a NeRF model has garnered attention in existing literature, closely intertwined with the next-best-view (NBV) problem, which entails identifying the optimal sensor location to maximize information acquisition about a given object or scene. Traditional approaches for tackling the NBV problem include [10]–[12], who build volumetric 3D models through active learning. More recently, Lee et al. [4], and NeU-NBV [13] have delved into constructing implicit neural models by addressing the NBV problem within a robotic framework. Additionally, ActiveNeRF[14] and Lin et al. [5] have approached the NBV problem purely from a visual perspective, without a robot manipulator. Central to these NBV techniques is characterizing *model uncertainty* or the internal uncertainty estimates of the robot's own model. Several approaches have been proposed to quantify the uncertainty in NeRF models. S-NeRF [15], ProbNeRF [16], and ActiveNeRF [14] integrate uncertainty prediction directly into the NeRF architecture. Lee et al. [4] models uncertainty as the entropy of the weight distribution along camera rays. Lin et al. [5] leverage variance in NeRF ensemble renderings for uncertainty quantification, while Sunderhauf et al. [17] employ a combination of ensemble variance and termination probabilities along rays.

Our work differs from the NBV approaches discussed above in two key aspects: firstly, by incorporating costs associated with each action and the robot's kinematics constraints, and secondly, by addressing the challenge of finding the next-best-view in the continuous SE(3) space while also permitting discrete actions through robot interactions, rather than focusing solely on selecting the best k images from a discrete set of (image, camera-pose) pairs.

III. Background and Problem Setup

A. NeRF-based Object Models

Over the recent years, Neural Radiance Fields (NeRF) [1] have gained prominence as an effective implicit neural representation technique for synthesizing novel views of a scene from a set of N RGB images and their associated camera poses. NeRF employs a neural network to represent each scene, predicting both the volumetric density and view-dependent color for any given point within the scene. Specifically, the volumetric density σ and RGB color $\mathbf{c}$ for each scene point are computed based on the parameters Θ of a Multilayer Perceptron (MLP), denoted by F. This MLP, is characterized by its input comprising the 3D position $\mathbf{x} = (x, y, z)$ and the viewing direction $\mathbf{d} = (d_x, d_y, d_z)$, outputs the ordered pair $(\sigma, \mathbf{c})$, collectively defining the scene's *radiance field*.

To render a novel view, NeRF traces camera rays for each pixel on the image plane, parameterized as $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where $t \geq 0$, $\mathbf{o}$ represents the camera origin, and $\mathbf{d}$ is the unit vector in the direction of the ray. For each ray, N points $\{\mathbf{r}_i = \mathbf{o} + t_i\mathbf{d}\}_{i=1}^{N}$ are sampled and processed by the MLP to obtain densities and colors. These are then integrated using volume rendering techniques (for further details, refer to [1]) to approximate the color $\hat{\boldsymbol{C}}(\mathbf{r})$, depth $\hat{D}(\mathbf{r})$, and opacity $\hat{O}(\mathbf{r})$ of each pixel. The NeRF model approximates these quantities using the Quadrature Rule [18], expressed as follows:

$$\hat{\boldsymbol{C}}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i \mathbf{c}_i, \hat{D}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i t_i, \hat{O}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i, \quad (1)$$

$$\alpha_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)\left(1 - \exp(-\sigma_i \delta_i)\right), \quad (2)$$

where σ_i and $\mathbf{c}_i$ denote the density and color predicted by the model at point $\mathbf{r}_i$ along ray $\mathbf{r}$, respectively, and $\delta_i = t_{i+1} - t_i$ represents the distance between adjacent samples along the ray.

B. Learning NeRF-based Object Models

Our problem concerns a robot manipulator in a tabletop environment and an object placed near the table's center. The robot is tasked to acquire a 3D representation of the object, which can be subsequently leveraged to manipulate the object in any position and orientation. Let A denote the robot's actions which include: `Move(`p_i`)` which position the robot arm to $SE(3)$ pose p_i, `Flip()`, which allows the robot to flip an object within its grasp using its object model, and `Capture()`, where the robot acquires an image from the camera attached to the robot arm. Further, let $\Gamma(a)$ denote the cost of an action $a \in A$.

The robot is required to execute a sequence of actions $A^* = (a_1, a_2, \ldots, a_n)$, where each a_i represents a specific combination of actions from A. After executing each action a_i, the robot applies a capture function $Capture()$ action to obtain an image. The collected images $I^* = (I_1, I_2, \ldots, I_n)$ are then used to train a NeRF model F_θ. Given a partially trained model $F_{\Theta_{k-1}}$, based on images $i_1, i_2, \ldots, i_{k-1}$, the goal is to identify the next action a_k that enables the robot to capture an image from a viewpoint where the model exhibits the highest uncertainty, while also minimizing the associated action cost $\Gamma(a_k)$.

IV. Technical Approach

Our approach for active learning of NeRF-based object models consists of (i) estimating model uncertainty for a partially-built model, (ii) determining the next informative and feasible action and (iii) incorporating object re-orientation and pose re-acquistion. These modules are detailed in this section (see Fig. 2). Formally, we express the

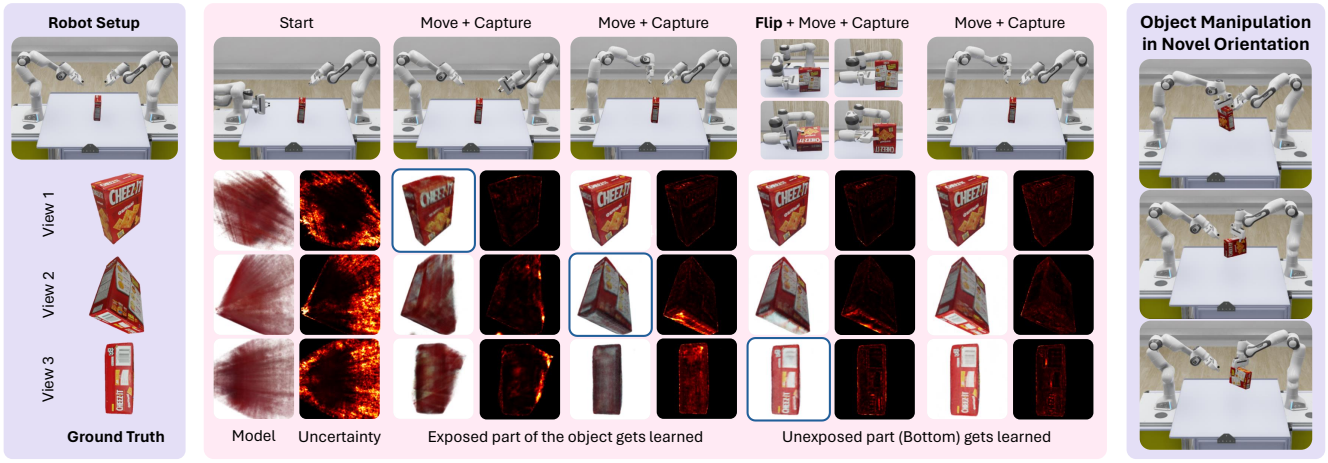


Fig. 2: **Active Learning in Action.** We show the RGB images and uncertainty maps rendered from trained models during our active learning process. The GT images are shown for reference. We note from the figure that before flipping, the bottom surface of the object has high uncertainty, which only diminishes once we perform the flip and acquire information about the bottom surface. The robot then uses the acquired object model to manipulate the object in any orientation.

aforementioned objective as optimizing the following:

$$a_k = \arg\max_a \left[U(F_{\Theta_{k-1}}, p) - \lambda \Gamma(a) \right], \tag{3}$$

where, p represents the 6 degrees of freedom (DoF) pose achieved by the robotic arm upon executing action a, and $U(F_{\Theta_{k-1}}, p)$ quantifies the uncertainty in the model from pose p. The objective can be equivalently expressed as minimizing the loss function $L(a)$, defined as:

$$L(a) = \lambda \Gamma(a) - U(F_{\Theta_{k-1}}, p) \tag{4}$$

A. Estimating Model Uncertainty

As discussed in Section III, quantifying the uncertainty present in a partial NeRF model from a given pose is crucial for our approach. Following the methodology of Lin et al. [5], we employ an ensemble-based strategy to measure this uncertainty. Specifically, we train M NeRF models using the same set of images but initialize each model with distinct weights sampled from a Xavier uniform distribution. By rendering images from these M models for any selected camera pose, we calculate the total variance across the RGB color channels and produce an uncertainty heatmap (see Fig. 3).

The overall uncertainty for a given pose is determined by aggregating the uncertainties of individual pixels within the rendered image. Therefore, the uncertainty associated with a pixel corresponding to ray $\mathbf{r}$ is defined as the variance of the estimated colors $\hat{\boldsymbol{C}}_i(\mathbf{r})$, calculated as follows:

$$\sigma^2(\mathbf{r}) = \frac{1}{M} \sum_{k=1}^{M} \|\boldsymbol{\mu}(\mathbf{r}) - \hat{\boldsymbol{C}}_k(\mathbf{r})\|^2, \text{where} \tag{5}$$

$$\boldsymbol{\mu}(\mathbf{r}) = \frac{1}{M} \sum_{k=1}^{M} \hat{\boldsymbol{C}}_k(\mathbf{r}), \tag{6}$$

and $\boldsymbol{\mu}(\mathbf{r})$ and $\hat{\boldsymbol{C}}_i(\mathbf{r})$ are vectors representing the RGB color channels. Here, M denotes the total number of models in the NeRF ensemble. Using the expression for $\sigma^2(\mathbf{r})$, the uncertainty for a pose p can be quantified as the sum of the uncertainties for all rays emanating from p:

$$U(F_{\Theta_{k-1}}, p) = \sum_{\mathbf{r} \in \text{Rays}(p)} \sigma^2(\mathbf{r}). \tag{7}$$

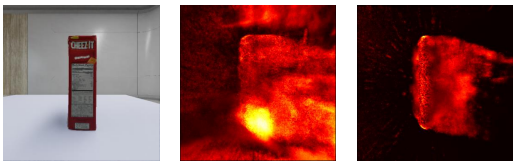


Fig. 3: **Effect of background on uncertainty estimation.** Uncertainty heatmaps of NeRF models trained on segmented v/s original images. Segmentation of the object removes artefacts arising from modeling background visual appearance.

Note that creating a 3D representation encompassing the entire scene results in an estimated uncertainty that reflects both the object of interest and the surrounding environment. Consequently, employing an uncertainty-based next-best-view (NBV) strategy under such conditions inadvertently optimizes for the reduction of background uncertainty as well, which diverges from our primary objective. Moreover, this method proves ineffective in cluttered scenes populated with multiple objects. Our aim is to isolate and enhance the uncertainty associated exclusively with the object of interest. To this end, we employ Grounded-SAM [19], a technique that utilizes textual prompts to generate object masks through the integration of Grounding DINO [20] and SAM [21], facilitating the training of NeRF models on segmented images. This approach provides a more accurate assessment of model uncertainty from the perspective of the object of interest (refer to Fig. 3).

Sünderhauf et al. [17] argue that RGB uncertainty does not adequately represent the model's epistemic uncertainty, particularly in relation to scene elements that remain unobserved during training. They propose quantifying epistemic uncertainty via the aggregation of termination probabilities

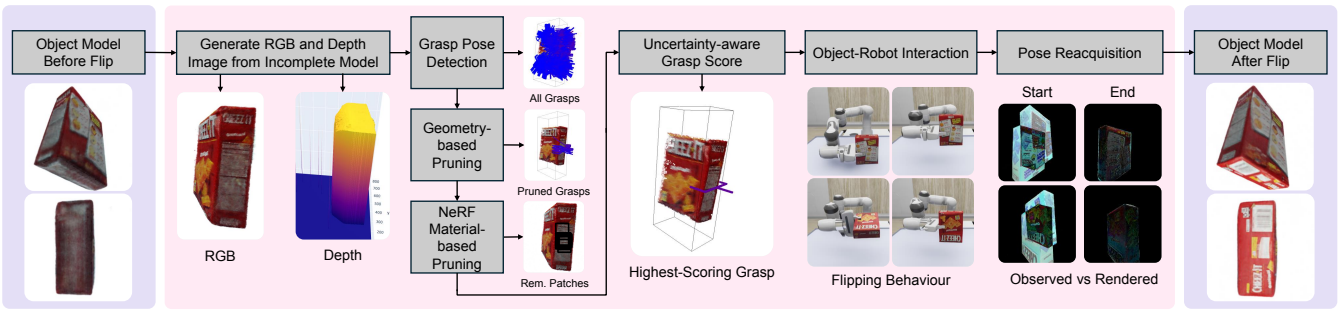


Fig. 4: **Object Re-orientation Approach.** First, the RGB and Depth images are rendered from the object's current NeRF model. Using these, AnyGrasp detects potential grasps, which are then pruned based on the geometry of the generated point cloud and NeRF's material density on grasp patches. The best grasp is selected from the remaining using our uncertainty-aware grasp score. The robot executes the chosen grasp to re-orient the object, and the modified iNeRF is employed to re-acquire the object's pose in its new orientation. We show the quality of the object models before and after the flip. The post-flip model is obtained by capturing images in a re-oriented position and adding them to the training dataset.

for points sampled along each ray, noting that uncertainty peaks when rays fail to intersect with the scene. However, this method is not applicable to our scenario, as our focus lies on quantifying object-specific uncertainty rather than that of the entire scene. Rays that do not intersect with the object contribute to an increased epistemic uncertainty, particularly for camera views distant from or oriented away from the object. Identifying and filtering out rays that do not intersect with the object of interest is a much harder problem with an apriori unknown object model. To circumvent this, we use RGB uncertainty as a proxy metric that effectively indicates heightened uncertainty in views of the object that have not been previously observed. Additionally, we conduct ablation studies comparing our approach with a modified version of their uncertainty measure, as detailed in Section VI-D to further highlight that RGB uncertainty is more amenable for robotic manipulation scenarios.

B. Uncertainty-guided Next Action Selection

We now tackle the challenge of identifying the next best action within the context of our active learning framework, given the current training dataset of the NeRF model and the robot's present pose. This task is formalized as minimizing the objective function $L(a)$, as defined in (4), where $U(F_{\Theta_{k-1}}, p)$ is articulated in (7). A notable issue arises due to the significant variance in uncertainty values across different NeRF models, even when trained on disparate images of the same object. To address this and standardize the selection of the λ parameter across all models, we normalize the uncertainty derived from (7) by the model's mean uncertainty, calculated over a set of poses randomly sampled from a uniform spherical distribution, ensuring the uncertainty prediction is *model-agnostic*.

Initially, we consider a simplified scenario where only `Move()` actions are permissible. This is because the model is not exposed to enough images to build a reasonable 3D representation required for grasping and consequently flipping. In this case, the minimization variable a in (4) is substituted with $p \in SE(3)$, representing the 6-DoF pose of the camera affixed to the robot's end-effector. The designated action for a pose p corresponds to maneuvering the end-effector to position the camera at p. To circumvent the limitations of naive gradient descent approaches, which falter due to the presence of numerous local minima within the objective function, we propose a bi-level optimization strategy. The primary level involves selecting a *sparse* and *diverse* subset of k candidate poses from n randomly sampled poses, all oriented towards the workspace's center. Subsequently, at the secondary level, we execute a gradient descent search from each candidate pose, mitigating the risk of converging to suboptimal local minima. The final solution is determined by selecting the candidate pose from the second level that yields the lowest objective function value.

Subsequently, in scenarios where `Flip()` actions are also considered, the problem is decomposed into two subproblems: 1) Identifying the optimal action assuming no flip action is permitted, and 2) Adjusting the coordinate axes to simulate a flip action and determining the optimal subsequent action. The cost associated with `Flip()` is accounted for exclusively in the second subproblem. The ultimate optimal action is selected based on the lower value of $L(a)$ obtained from these subproblems.

Our methodology accommodates any form of action cost $\Gamma(a)$ specified in (4). For our experiments, we define $\Gamma(a)$ for a `Move()` action, which transitions the end-effector from (r_1, q_1) to (r_2, q_2) (with r indicating position and q representing rotation in quaternion form), as follows:

$$\Gamma(a) = \beta_1(1 - d(q_1, q_2)) + \beta_2 d(r_1, r_2), \tag{8}$$

whereas, for a `Flip()` action, we set $\Gamma(a) = \beta_3$. The cumulative cost Γ for a sequence of actions is the sum of the costs for individual actions, where β_i are adjustable based on the relative importance of each action cost component.

C. Object Re-orientation during Model Acquisition

In the subsequent phase of our methodology, we delve into the estimation of the grasp pose (see Fig. 4). To compute the optimal lateral grasp pose based on the currently available partial NeRF model, we employ AnyGrasp [22]. The quality of the selected grasp pose significantly influences the decision-making process of our next-best-action algorithm, particularly in determining the possibility of a flip action in the ensuing iteration. AnyGrasp operates by processing depth images, from which it generates a collection of grasp pose and grasp confidence pairings. However, our empirical

TABLE I: Comparison of our method with ActiveNeRF and other baselines. All the baselines include `Flip()` action as detailed in section V-B and are trained with segmented images. F-score* represents the F-score values multiplied by 10. As evident from the tables, our approach outperforms other methods by a significant margin.

Method	Basket		Cheezit Box		Mug		Rubik's Cube		Spam Can		**Total**	
	PSNR↑	F-score*↑	PSNR↑	F-score*↑	PSNR↑	F-score*↑	PSNR↑	F-score*↑	PSNR↑	F-score*↑	PSNR↑	F-score*↑
Model quality after 20 iterations **without** grasping												
Ours	**17.2** ± **0.1**	**4.2** ± **0.5**	**21.8** ± **0.2**	**4.3** ± **0.2**	26.3 ± 0.1	6.5 ± 0.2	**30.3** ± **0.3**	3.8 ± 0.1	23.9 ± 0.6	4.1 ± 0.3	**23.9** ± **0.1**	**4.6** ± **0.1**
Random	17.0 ± 0.2	4.0 ± 0.6	21.4 ± 0.4	4.0 ± 0.4	**26.5** ± **0.2**	**6.6** ± **0.1**	29.5 ± 0.2	**4.1** ± **0.3**	**24.7** ± **0.2**	**4.4** ± **0.0**	23.8 ± 0.1	4.6 ± 0.2
Furthest	17.1 ± 0.1	3.0 ± 0.2	21.2 ± 0.1	4.2 ± 0.2	26.5 ± 0.3	5.8 ± 0.2	28.6 ± 0.5	3.6 ± 0.2	23.9 ± 0.3	4.0 ± 0.2	23.5 ± 0.1	4.1 ± 0.1
Active [14]	15.5 ± 0.5	2.6 ± 0.2	19.0 ± 0.1	2.9 ± 0.3	24.8 ± 0.9	4.4 ± 0.1	27.0 ± 0.6	3.4 ± 0.3	19.7 ± 1.3	3.1 ± 0.1	21.2 ± 0.4	3.3 ± 0.1
Model quality after 20 iterations **with** grasping												
Ours	**16.9** ± **0.1**	3.4 ± 0.3	**21.3** ± **0.3**	**4.0** ± **0.2**	**26.9** ± **0.2**	**5.9** ± **0.1**	**31.6** ± **0.5**	**4.0** ± **0.3**	**23.4** ± **0.5**	**3.8** ± **0.1**	**24.0** ± **0.2**	**4.2** ± **0.1**
Random	16.0 ± 0.6	**3.5** ± **0.8**	20.9 ± 1.0	3.9 ± 0.4	22.2 ± 0.8	4.6 ± 0.2	29.1 ± 0.9	3.5 ± 0.5	22.8 ± 1.8	3.4 ± 0.4	22.2 ± 0.5	3.8 ± 0.2
Furthest	16.5 ± 0.2	3.1 ± 0.2	19.8 ± 0.7	3.7 ± 0.4	24.1 ± 0.3	4.8 ± 0.7	27.8 ± 0.9	3.8 ± 0.4	21.1 ± 1.5	3.3 ± 0.3	21.9 ± 0.4	3.7 ± 0.2
Active [14]	16.1 ± 0.2	2.6 ± 0.2	19.2 ± 0.2	2.5 ± 0.9	23.6 ± 3.2	5.5 ± 0.3	27.6 ± 0.8	3.9 ± 0.6	22.1 ± 0.2	3.2 ± 0.1	21.7 ± 0.7	3.5 ± 0.2
Model quality attained given a cost budget of 2 **without** grasping												
Ours	**17.1** ± **0.1**	**3.9** ± **0.3**	**21.4** ± **0.2**	**4.3** ± **0.1**	**26.0** ± **0.3**	**5.8** ± **0.5**	**29.8** ± **0.7**	**4.5** ± **0.2**	**23.7** ± **0.4**	**4.1** ± **0.2**	**23.6** ± **0.2**	**4.5** ± **0.1**
Random	17.0 ± 0.2	3.8 ± 0.4	19.3 ± 1.1	3.7 ± 0.4	25.2 ± 0.9	5.7 ± 0.7	28.6 ± 0.6	3.8 ± 0.1	23.3 ± 1.4	3.6 ± 0.3	22.7 ± 0.4	4.1 ± 0.2
Furthest	16.4 ± 0.4	2.6 ± 0.1	19.2 ± 0.2	3.5 ± 0.1	24.7 ± 1.1	4.4 ± 0.1	26.6 ± 1.2	4.1 ± 0.1	22.6 ± 0.4	3.9 ± 0.1	21.9 ± 0.3	3.7 ± 0.0
Active [14]	15.4 ± 0.4	3.1 ± 0.1	18.4 ± 0.2	3.1 ± 0.1	24.1 ± 0.4	4.1 ± 0.2	26.3 ± 0.6	3.4 ± 0.3	19.5 ± 1.7	3.6 ± 0.1	20.7 ± 0.4	3.5 ± 0.1
Model quality attained given a cost budget of 2 **with** grasping												
Ours	**16.9** ± **0.1**	**3.3** ± **0.4**	**21.2** ± **0.2**	**3.9** ± **0.2**	**26.8** ± **0.4**	**5.8** ± **0.3**	**30.9** ± **0.7**	**4.0** ± **0.3**	**23.7** ± **0.6**	**3.8** ± **0.2**	**23.9** ± **0.2**	**4.2** ± **0.1**
Random	16.0 ± 0.3	3.0 ± 0.3	20.9 ± 1.0	3.8 ± 0.2	22.3 ± 0.8	4.4 ± 0.2	28.5 ± 1.3	3.6 ± 0.3	22.8 ± 1.9	3.4 ± 0.4	22.1 ± 0.5	3.6 ± 0.1
Furthest	16.5 ± 0.2	1.5 ± 0.8	19.9 ± 0.2	3.6 ± 0.2	23.5 ± 0.4	4.3 ± 0.3	27.7 ± 0.9	3.5 ± 0.4	22.1 ± 1.7	3.3 ± 0.3	21.9 ± 0.4	3.2 ± 0.2
Active [14]	15.8 ± 0.2	3.3 ± 0.2	19.2 ± 0.3	3.2 ± 0.1	21.9 ± 2.0	5.1 ± 0.2	26.4 ± 0.8	2.7 ± 1.0	21.6 ± 0.5	3.3 ± 0.2	21.0 ± 0.8	3.5 ± 0.2

observations reveal that the confidence scores produced by AnyGrasp are not directly applicable for selecting grasp poses as it is trained on RGBD images obtained from depth cameras, which contrasts with our utilization of partial object models that may include extraneous geometric features. We use our partially trained NeRF model to generate a depth image for a horizontal grasp. We then generate all candidate grasp poses using AnyGrasp. We prune grasp poses based on the following criteria: (i) Distance from center of the point cloud, (ii) Grasp angle w.r.t. surface normal, and (iii) Average opacity $\hat{O}(\mathbf{r})$ (1) of the grasp patch. We then assign a score to each grasp pose, to select the most suitable grasp. Our grasp score is defined as follows

$$G_s = \frac{1-\theta}{U_d} \tag{9}$$

where θ is the angle between the grasp pose and surface normal. θ should be minimized as grasping from a non-lateral grasp increases the probability of the object toppling during or after the flip. U_d is the variance of rendered depths summed over the rays of the grasp patch. Its computation is similar to that of (5), and (7) with predicted depth $\hat{D}(\mathbf{r})$ (1) being used instead of color. We minimize U_d to be certain about the location of the object surface in 3D space near the grasp pose. This approach ensures that the grasp poses chosen are not only theoretically viable according to AnyGrasp's criteria but also practically applicable within the constraints and current state of our partial object models.

D. Pose Re-acquisition for Model Unification

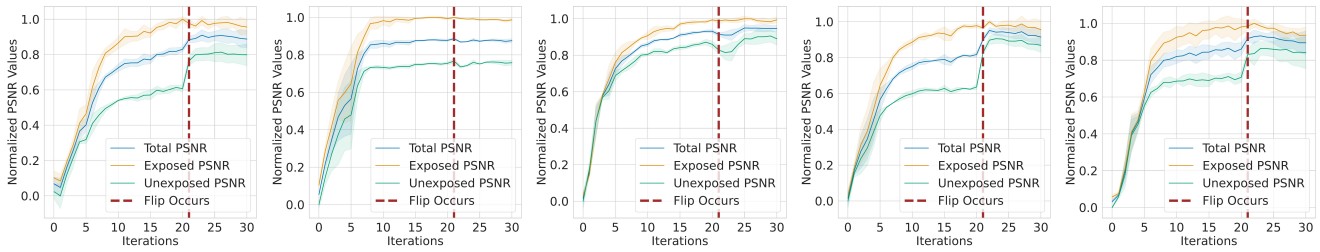
The stochastic nature of robotic actions necessitates the recovery of an object's pose after the execution of a `Flip()` action. To address this, we employ a methodology inspired by iNeRF [23]. Subsequent to the interaction, the robot captures an RGB image, the pose of which is ascertainable through the robot's forward kinematics. The alteration in the object's pose due to the interaction, however, results in discrepancies between the newly captured image and what NeRF would render from the same camera position. To reconcile these differences, we optimize for the camera pose that minimizes the Sum of Squared Differences (**SSD**) between the captured image and NeRF's predicted image, thereby enabling an estimation of the object's post-interaction pose.

Our experimentation reveals that the precision of pose estimation via the original iNeRF framework does not meet the requisite standards for eliminating the discrepancies in the data collected before and after re-orientation. Consequently, we introduce two significant enhancements to the conventional iNeRF approach. Firstly, diverging from iNeRF's gradient-based search methodology, we adopt non-gradient-based optimization techniques, which have demonstrated superior performance in accurately recovering object poses. Specifically, we combine three distinct optimization strategies: Nelder-Mead [24], COBYLA [25], [26], and Powell's method [27]. The most accurate pose estimation from among these methods is selected based on the lowest **SSD** score. Secondly, in lieu of relying on a single image for pose estimation, we utilize multiple images to enhance the robustness. The optimization process is thus aimed at minimizing the cumulative **SSD** across all pairs of captured and NeRF-predicted images. This multi-image strategy bolsters the accuracy of our pose estimation, ensuring a more reliable recovery of the object's pose post-interaction.

V. Evaluation Setup

A. Simulation Environment and Dataset

Our experiments consider a table as a workspace with two Franka Emika robotic arms, situated at opposite ends.



(a) Cheezit Box (b) Basket (c) Mug (d) Rubik's Cube (e) Spam Can

Fig. 5: **Effect of Flip on Model Quality**.We demonstrate the impact of flipping objects on model quality (PSNR). The dashed line indicates the iteration at which the object is flipped. The PSNR is shown for the **Exposed** and **Unexposed** subsets of the validation set, representing camera poses above and below the object center, respectively. In most cases, the Exposed PSNR remains almost constant, while the Unexposed PSNR shows a significant increase, leading to an overall improvement in total PSNR. PSNR values are min-max normalized in each plot.

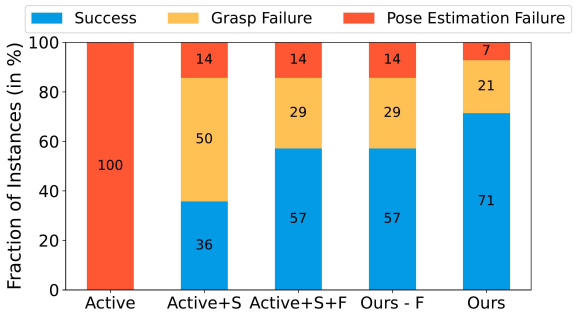


Fig. 6: **Grasping Performance Analysis.** Analysis of grasp task success rate and failure scenarios. **Active** denotes vanilla ActiveNeRF trained on captured images without segmentation and with no `Flip()` action. **S** denotes *object segmentation*, **F** denotes possibility of `Flip()` action.

This dual-arm setup is necessitated by the limitations of a single arm's reach and inability to capture images covering the entirety of an object's surface, especially areas directly opposite the arm. To simulate a realistic environment conducive to our active learning endeavors, we employ Nvidia's *Isaac Sim* simulator. We curate object models from the YCB dataset [28], focusing on objects amenable to lateral grasping and flipping. The dataset consists of five objects as shown in Fig 7.

B. Baselines and Metrics

Our evaluation framework benchmarks the proposed active learning strategy against three baselines to ensure a rigorous comparison: (i) *Random View*, where subsequent views are randomly selected, (ii) *Next Furthest View*, selecting the next view to maximize the cumulative distance from existing training views, and (iii) *ActiveNeRF* [14], implemented via the Kaolin-Wisp [29] framework. We integrated a `Flip()` action in these baselines to align them with our framework. Specifically, for (i) and (ii), a `Flip()` is executed at the first iteration that meets a predefined grasp score threshold (9). This approach is infeasible for (iii) due to its generation of partial models lacking precise surfaces, which complicates grasp pose determination. Consequently, in the case of (iii), we resort to a predetermined flip via an external manipulator at a specific iteration, assuming accurate post-flip object pose knowledge to ensure that the generated models are of the highest quality.

Evaluation metrics employed include 1) **PSNR** (Peak Signal-to-Noise Ratio) for assessing visual fidelity, with validation sets of 64 images per object, and 2) **F-score** [30] for measuring geometric accuracy, using point clouds derived from the trained NeRF models via Marching Cubes [31].

C. Other Implementation Details

In our experimental setup, the cost function parameters, including λ and β_i, play a pivotal role (see 4 and 8). For our purposes, λ is fixed at 1, and the β values are determined based on the relative average durations of their corresponding actions executed by the robot, reflecting a practical consideration of action cost in terms of time. However, the observations translate to other values of hyper-parameters as well.

Our methodology is implemented on the Kaolin-Wisp framework [29], utilizing the InstantNGP model [32]. We train an ensemble of five models on a single NVIDIA A40 GPU. For a given pose, we consider the prediction of the ensemble as the mean of the predictions of individual models. On average, training a single NeRF model takes approximately 36 seconds, while determining the next best action requires about 3 minutes. These processes are amenable to parallelization, potentially reducing computation times significantly. The models are trained with images of 800×800 resolution, and PSNR evaluations are conducted using images at their full resolutions.

VI. Results

A. Evaluation of Model Quality

We evaluate the quality of the model acquired using the proposed approach in relation to the baseline models. This section presents our findings under two distinct conditions: (i) `Flip()` actions prohibited, and (ii) `Flip()` actions permitted. First, to ensure an equitable comparison, particularly with ActiveNeRF, which does not focus on minimizing cumulative action costs, we conduct our experiments and those of the baselines across a uniform number of iterations, set at 20. This fixed iteration count is selected to afford ample iterations for all methods to converge to reasonable NeRF models for manipulation.

Subsequently, we evaluate the quality of the acquired models within a fixed budget on the total action cost, aligning with our original research objective. The allocated cost budget is carefully chosen to be sufficiently generous, enabling the active learning frameworks to develop robust

TABLE II: Ablation results for uncertainty estimation technique. Epi, Total stands for epistemic and overall uncertainty as proposed by [17] (with modifications stated in section VI-D). F-score* represents the F-score values multiplied by 10

Uncertainty	Basket		Cheezit Box		Mug		Rubik's Cube		Spam Can		**Total**	
	PSNR↑	F-score*↑	PSNR↑	F-score*↑	PSNR↑	F-score*↑	PSNR↑	F-score*↑	PSNR↑	F-score*↑	PSNR↑	F-score*↑
Model quality after 20 iterations **without** grasping												
Ours	**17.2** ± **0.1**	**4.2** ± **0.5**	**21.8** ± **0.2**	**4.3** ± **0.2**	**26.3** ± **0.1**	**6.5** ± **0.2**	**30.3** ± **0.3**	3.8 ± 0.1	**23.9** ± **0.6**	**4.1** ± **0.3**	**23.9** ± **0.1**	**4.6** ± **0.1**
Entropy [4]	14.2 ± 0.3	3.0 ± 0.2	19.8 ± 0.2	3.4 ± 0.1	25.0 ± 0.6	5.6 ± 0.2	27.7 ± 0.6	4.0 ± 0.4	20.3 ± 0.3	4.0 ± 0.2	21.4 ± 0.2	4.0 ± 0.1
Epi [17]	15.5 ± 0.6	3.2 ± 0.3	20.4 ± 0.3	3.7 ± 0.1	25.8 ± 0.3	6.0 ± 0.4	29.0 ± 0.1	**4.5** ± **0.7**	21.0 ± 0.4	3.8 ± 0.1	22.3 ± 0.2	4.2 ± 0.2
Total [17]	16.6 ± 0.4	3.5 ± 0.2	19.7 ± 0.6	3.8 ± 0.3	21.2 ± 2.2	4.1 ± 0.7	25.6 ± 0.5	3.0 ± 0.2	22.1 ± 1.0	3.9 ± 0.1	21.0 ± 0.5	3.7 ± 0.2
Model quality attained given a cost budget of 2 **without** grasping												
Ours	**17.1** ± **0.1**	**3.9** ± **0.3**	**21.4** ± **0.2**	**4.3** ± **0.1**	**26.0** ± **0.3**	**5.8** ± **0.5**	**29.8** ± **0.7**	**4.5** ± **0.2**	**23.7** ± **0.4**	**4.1** ± **0.2**	**23.6** ± **0.2**	**4.5** ± **0.1**
Entropy [4]	14.8 ± 1.0	3.0 ± 0.2	19.3 ± 0.5	3.6 ± 0.1	24.9 ± 0.5	5.5 ± 0.1	27.7 ± 0.6	4.0 ± 0.4	20.3 ± 0.3	4.0 ± 0.2	21.4 ± 0.3	4.0 ± 0.1
Epi [17]	15.2 ± 0.2	3.1 ± 0.2	19.3 ± 0.2	3.4 ± 0.1	25.2 ± 0.3	5.3 ± 0.3	28.4 ± 0.7	4.5 ± 0.4	20.7 ± 0.2	4.0 ± 0.0	21.8 ± 0.2	4.1 ± 0.1
Total [17]	16.6 ± 0.4	3.5 ± 0.1	19.7 ± 0.6	3.8 ± 0.3	21.2 ± 2.2	4.1 ± 0.7	25.6 ± 0.5	3.0 ± 0.2	22.1 ± 1.0	3.9 ± 0.1	21.0 ± 0.5	3.7 ± 0.2

TABLE III: Fraction of Correct Pose Estimation Instances within Bounds
Tight Bound := (Rotation Error $< 2°$ AND Translation Error < 0.5 cm)
Loose Bound:= (Rotation Error $< 5°$ AND Translation Error < 1.0 cm)
Loss- *Single*: SSD of a single image, *Multi*: summed SSD of 4 images

Optimization Method	Tight Bound (in %)		Loose Bound (in %)	
	Single	Multi	Single	Multi
Nelder-Mead [24]	17.9	33.3	30.8	59.0
COBYLA [25] [26]	5.1	23.1	7.7	51.3
Powell [27]	51.3	61.5	56.4	64.1
Ours (Combined)	**51.3**	**69.2**	**61.5**	**82.1**

models, yet not so ample as to lead to quality saturation. Specifically, the cost budget is set to 3 for scenarios allowing the `Flip()` action and to 2 for those that do not, with the difference equal to the cost associated with a flip action.

The results are summarized in Table I. These results are derived from training both our model and the baselines on segmented images, ensuring a consistent basis for comparison. The proposed method improves PSNR (by 14%) and F-Score (by 20%) compared to ActiveNeRF.

B. Benefit of Object Re-orientation

Fig. 5 shows that the model quality improves significantly after flipping and exposing previously unseen surfaces. We also show images rendered from models (trained for 20 iterations each) without `Flip()` and compare it against our best model (Fig. 7). We conclude from the figure that the bottom surface of the object can only be learned by re-orienting the object.

C. Grasping Performance Evaluation

To evaluate the practical utility of the NeRF models, we construct a dataset comprising objects placed in random positions and orientations on a tabletop setup. The robot is tasked to estimate the object pose using a trained NeRF model, followed by an attempt to execute a grasp. The effectiveness of the NeRF models is quantified using GSR (Grasp Success Rate). We conduct a comparative analysis of the performance of NeRF models developed with our active learning methodology and with ActiveNeRF, both with and without the inclusion of the `Flip()` action. The outcomes of this comparison, including the GSR and a breakdown of failure modes for all model variants, are depicted in Fig. 6. We note that ActiveNeRF, is unable to grasp any object, whereas the GSR for our proposed approach is 71%.

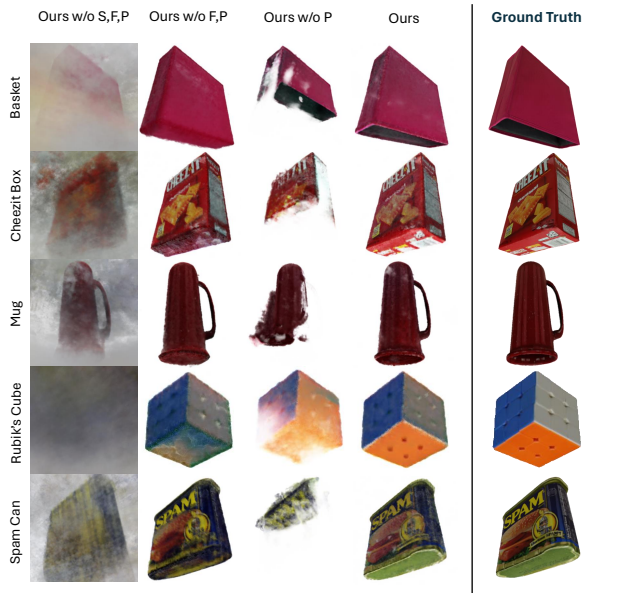


Fig. 7: **Model Quality Comparison under Pipeline Modifications.** Comparison of learned model quality with ground truth meshes. The different models are obtained after removing certain components from our pipeline. **S** denotes *object segmentation*, **F** denotes *flip*, and **P** denotes *pose re-acquisition* after the flip has been executed

D. Ablations

First, to show the necessity of each component of our pipeline, we remove them step-by-step and run for 20 iterations each. The qualitative results are shown in Fig. 7. Next, we delve into the effectiveness of different optimization techniques for pose re-acquisition, including Nelder-Mead [24], COBYLA [26], and Powell's method [27], alongside our approach of selecting the minimum loss among these. The comparative analysis extends to single versus multi-image optimization strategies, as elucidated in Section IV-D, with outcomes presented in Table III. Notably, Fig. 7 demonstrates the crucial role of pose re-acquisition, highlighting that its absence results in significantly degraded NeRF models, rendering them impractical for robotic applications.

Finally, we conduct ablation studies on various uncertainty estimation methodologies. We assess the entropy-based uncertainty metric introduced by Lee et al. [4] and the epistemic and total uncertainties delineated by Sünderhauf et al. [17]. As discussed in IV-A, their epistemic uncertainty takes on

the maximum possible value of 1 for the pixels lying outside the segmented image of the object. Since these pixels should not contribute to object uncertainty, we assign them a value of 0. The results are shown in Table II.

VII. Limitations

A primary limitation of our approach is that we are limited by the quality of grasps generated by Anygrasp from our rendered depth images, so this technique does not work with transparent objects, for which generating depth images are difficult. Another limitation is that we have only considered occlusions of the bottom surface of an object when it is placed on a surface, however it is not difficult to incorporate other kinds of occlusions in our approach by modifying the robot interactions accordingly. Further, our pose re-acquisition strategy, despite its general efficacy, occasionally fails to accurately determine the object's pose, as indicated in Table III. Lastly, our approach is computationally intensive primarily due to the necessity of ensemble training. Although this process benefits from parallelization, it necessitates multiple GPUs for training with high-resolution images.

VIII. Conclusion

This paper introduces an active learning approach to acquire a NeRF model using both visual observation and re-orientation actions. This facilitates modeling of previously occluded surfaces allowing future pick/place interactions when the object may be positioned arbitrarily in any stable mode. Our approach estimates model uncertainty using an ensemble of models and uses that to estimate the next action (visual or re-orientation) to improve the model. Future work will explore improving the timing efficiency of ensembling and incorporate multi-step interactions, extensions to articulated objects, and experiments on real platforms.

References

[1] 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," in *ECCV*, 2020.

[2] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural radiance fields from one or few images," in *CVPR*, 2021.

[3] M. M. Johari, Y. Lepoittevin, and F. Fleuret, "Geonerf: Generalizing nerf with geometry priors," *IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[4] S. Lee, L. Chen, J. Wang, A. Liniger, S. Kumar, and F. Yu, "Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 070–12 077, 2022.

[5] K. Lin and B. Yi, "Active view planning for radiance fields," in *Robotics Science and Systems*, 2022.

[6] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," *ArXiv*, vol. abs/2110.14217, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:239998474.

[7] J. Kerr, L. Fu, H. Huang, *et al.*, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *6th Annual Conference on Robot Learning*, 2022.

[8] M. Adamkiewicz, T. Chen, A. Caccavale, *et al.*, "Vision-Only Robot Navigation in a Neural Radiance World," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 4606–4613, Apr. 2022, website: https://mikh3x4.github.io/nerf-navigation/.

[9] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint, "Learning multi-object dynamics with compositional neural radiance fields," in *Conf. on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 205, 2023, pp. 1755–1768.

[10] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3d object models using next best view manipulation planning," in *2011 IEEE international conference on robotics and automation*, IEEE, 2011, pp. 5031–5037.

[11] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3d reconstruction," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 3477–3484.

[12] J. Daudelin and M. Campbell, "An adaptable, probabilistic, next-best view algorithm for reconstruction of unknown 3-d objects," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1540–1547, 2017.

[13] L. Jin, X. Chen, J. Rückin, and M. Popović, "Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 11 305–11 312.

[14] X. Pan, Z. Lai, S. Song, and G. Huang, "Activenerf: Learning where to see with uncertainty estimation," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, Springer, 2022, pp. 230–246.

[15] 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.

[16] M. D. Hoffman, T. A. Le, P. Sountsov, *et al.*, "Probnerf: Uncertainty-aware inference of 3d shapes from 2d images," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2023, pp. 10 425–10 444.

[17] N. Sünderhauf, J. Abou-Chakra, and D. Miller, "Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 9370–9376.

[18] N. Max, "Optical models for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.

[19] T. Ren, S. Liu, A. Zeng, *et al.*, *Grounded sam: Assembling open-world models for diverse visual tasks*, 2024. arXiv: 2401.14159 [cs.CV].

[20] S. Liu, Z. Zeng, T. Ren, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[21] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," *arXiv:2304.02643*, 2023.

[22] H.-S. Fang, C. Wang, H. Fang, *et al.*, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, 2023.

[23] 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.

[24] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[25] M. J. Powell, *A direct search optimization method that models the objective and constraint functions by linear interpolation*. Springer, 1994.

[26] M. J. Powell, "Direct search algorithms for optimization calculations," *Acta numerica*, vol. 7, pp. 287–336, 1998.

[27] M. J. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The computer journal*, vol. 7, no. 2, pp. 155–162, 1964.

[28] B. Calli, A. Singh, J. Bruce, *et al.*, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.

[29] T. Takikawa, O. Perel, C. F. Tsang, *et al.*, *Kaolin wisp: A pytorch library and engine for neural fields research*, 2022.

[30] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[31] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Seminal graphics: pioneering efforts that shaped the field*, 1998, pp. 347–353.

[32] 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.