

NeRF-LoC: Transformer-Based Object Localization Within Neural Radiance Fields

Jiankai Sun^{1†*}, Yan Xu^{2*}, Mingyu Ding³, Hongwei Yi⁴, Jingdong Wang⁵, Liangjun Zhang⁶, Mac Schwager¹

Abstract—Neural Radiance Fields (NeRFs) have been successfully used for scene representation. Recent works have also developed robotic navigation and manipulation systems using NeRF-based environment representations. As object localization is the foundation for many robotic applications, to further unleash the potential of NeRFs in robotic systems, we study *object localization within a NeRF scene*. We propose a transformer-based framework NeRF-LoC to extract 3D bounding boxes of objects in NeRF scenes. NeRF-LoC takes a pre-trained NeRF model and camera view as input, and produces labeled 3D bounding boxes of objects as output. Concretely, we design a pair of parallel transformer encoder branches, namely the coarse stream and the fine stream, to encode both the context and details of target objects. The encoded features are then fused together with attention layers to alleviate ambiguities for accurate object localization. We have compared our method with the conventional transformer-based method and our method achieves better performance. In addition, we also present the first NeRF samples-based object localization benchmark NeRFLoCBench.

Index Terms—Object Localization, Neural Radiance Fields

I. INTRODUCTION

When a robot explores a novel environment, it needs to perceive the objects around it and understand the spatial relationships among them, then it can plan motions to navigate around and manipulate those objects in order to accomplish its desired goals. Accurately localizing the objects in 3D space thus becomes a problem of crucial importance for many robotic applications. The choice of object localization strategy depends on the types of the adopted perception model and the underlying environment representation. For instance, some recent works [1–4] propose to ground the instructions from the observed 2D camera images, where 2D object localization method [5] is hence applied. Wada *et al.* [6] construct 3D maps (represented by point clouds/voxels) for the purpose of object pose estimation. Then, the 3D CNNs become a natural choice to process the 3D data. Compared with representing the scene with a set of 2D images, the reconstructed 3D maps maintain 3D topology and richer geometric information. This greatly facilitates the robots to understand the scene and make better decisions.

However, point cloud/voxel-based map representations have significant limitations. These classic 3D representations generally lack radiance (i.e., color and lighting) details, which

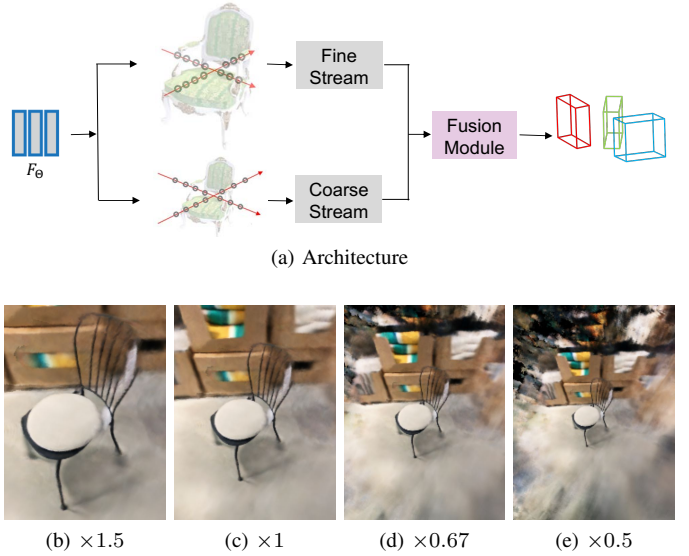


Fig. 1: (a) NeRF-LoC networks process information at multiple scales. The coarse stream learns to sample the most informative positions in the overview image, while the fine stream processes the near-field image to extract fine-grained information. These two streams are connected by an attention fusion module, after which 3D object localizations are predicted. (b)-(e) shows images rendered by NeRF from fine to coarse scales. We can see that they contain different amounts of information.

are essential for object localization, and the data from these traditional models can be sparse or incomplete. The robot that relies on point cloud or voxel-based models may thus fail to leverage realistic structures or appearances of objects as humans do. Recently, researchers have identified this issue and proposed to represent the environment with Neural Radiance Fields (NeRFs) [7], and successfully deploy it in SLAM [8, 9] systems. NeRF is used as a technique for continuous scene representation where the 3D scene is compressed in feed-forward neural networks that memorize the density and radiance values of each 3D location. Compared with the classic representation, NeRF is much more compact and contains more photo-realistic elements. Nevertheless, we notice that the conventional object localization methods have difficulties being directly applied to this scene representation and the object localization in a NeRF has not been well studied in the literature. On the other hand, from a robot planning perspective, NeRFs have been used with great success in robotic motion planning [10] and

* denotes equal contribution.

¹ Stanford University email: jksun@cs.stanford.edu. ² The Chinese University of Hong Kong email: yanxu@link.cuhk.edu.hk ³ The University of Hong Kong ⁴ Max Planck Institute for Intelligent Systems, Tübingen, Germany ⁵ Baidu Inc. ⁶ Robotics and Autonomous Driving Lab of Baidu Research, Sunnyvale, CA 94089 USA [†] work done as an intern at Baidu Research.

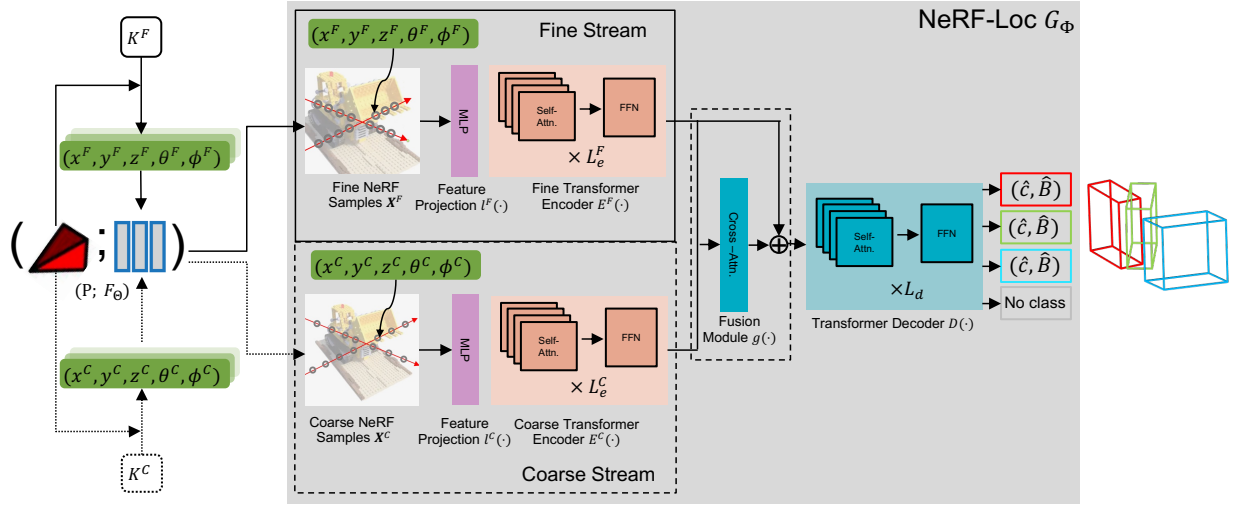


Fig. 2: **Framework.** Given the camera pose $\mathbf{P} \in SE(3)$ the robot observes with, we sample a set of query points (x, y, z) along the rays (emitted from the camera center passing through the camera plane) and then localize the objects based on the sampled field values. Our framework has two streams, the fine stream and the coarse stream. They share similar transformer-based architectures but are in charge of processing different scopes of NeRF samples. The different sampling scopes (Field of Views) are controlled by intrinsics K^C and K^F . After the feature encoding, the cross-attention-based Fusion Module fuses embeddings from the two streams. Then, the fused features are decoded by the Transformer Decoder. Finally, the 3D bounding boxes and the corresponding categories are predicted by MLP heads.

manipulation [11]. However, motion planning with implicit neural representations requires that the target location is first known, rather than randomly specified as in previous work [10]. For example, the gap between a language instruction like “go to the chair” and path planning is to know where the target “chair” is. Our approach hopes to bridge this gap.

To fully unleash the potential of NeRFs in robotic applications, we study *3D object localization in neural fields*, a non-trivial task that can be seen as a step towards bridging the gap between robotic perception and planning/control in environments represented by NeRFs. The challenge mainly lies in how to effectively exploit the geometric information contained in the NeRF representation, and in particular, to take advantage of the ease of scaling with NeRF representation. Specifically, we design a transformer-based network to directly estimate object localization within a Neural Radiance Field. To take advantage of the ease of scaling up and down with the NeRF representation, our framework includes two sub-networks of coarse and fine streams, which efficiently fuse the information from the wider horizon and the zoom-in view for comprehensive scene understanding. Intuitively, the coarse stream perceives a broader view which provides more global contexts to alleviate the ambiguities, while the fine stream helps to localize the object at a finer level.

Currently, there is a scarcity of datasets for our task, which needs to satisfy having both NeRF representations and 3D bounding box annotations. Objectron [12] is a dataset recently proposed suitable for our task. It contains consecutive video frames and corresponding camera poses, which can be used for NeRF training. We build a NeRF object localization bench-

mark **NeRF-LocBench** based on Objectron [12] and evaluate our method with it. We show that our method outperforms the previous baselines.

In summary, our main contributions are as follows:

- We introduce the problem of *object localization in a neural radiance world*, a step towards semantic robotic perception with neural scene representations, which can be used for downstream tasks such as planning and control.
- We propose **NeRF-Loc**, a framework for 3D object localization that exploits the geometric information imposed by neural representations. We also present the first NeRF samples-based object localization benchmark **NeRF-LocBench**.
- We evaluate our approach extensively and experimental results show that our approach significantly outperforms existing methods in NeRF object localization task.

II. RELATED WORK

Neural Radiance Field (NeRF). NeRF [7] utilizes an MLP network to predict the density and color of points in a scene, which allows for differentiable rendering by tracing rays through the scene and integrating them. Semantic NeRF [13] extends NeRF to jointly encode 2D semantics with appearance and geometry. There are also some few-shot NeRFs [14, 15] to perform novel view synthesis from a sparse set of views. Recently, object-centric NeRFs investigate how the synthesis process can be controlled at the object-level. GIRAFFE [16] incorporates a compositional 3D scene representation into the generative model which leads to more controllable image

Algorithm 1: 3D Object Localization on Neural Fields

Input: NeRF function F_Θ , observation pose \mathbf{P} , coarse intrinsic matrix K^C and fine intrinsic matrix K^F

Output: J object proposals $\{\hat{\psi}^j\}_{j=1}^J$, $\hat{\psi}^j = \{\hat{B}^j, \hat{c}^j\}$, $\hat{B} = \{x_i^b, y_i^b, z_i^b\}_{i=1}^{N_c}$ is the coordinates for each bounding box. N_c is the number of corner points. \hat{c} is the predicted class for each object.

- 1 $\mathbf{X}^C = F_\Theta(\mathbf{P}, K^C)$;
- 2 $\mathbf{X}^F = F_\Theta(\mathbf{P}, K^F)$;
- 3 Compute embeddings using coarse encoder and fine encoder (see Equ. (5) and Equ. (6));
- 4 Fuse embeddings using attention fusion module (see Equ. (7));
- 5 Compute output embeddings using decoder;
- 6 **for** $j \in 1, 2, \dots, J$ **do**
- 7 Predict bounding box instance $\hat{\psi}^j = \{\hat{B}^j, \hat{p}^j\}$ using prediction head;
- 8 **end**
- 9 Compute optimal matching σ^* between $\{\psi^j\}_{j=1}^J$ and $\{\hat{\psi}^j\}_{j=1}^J$ using Hungarian matcher;
- 10 Compute final loss \mathcal{L}_H between $\{\psi^j\}_{j=1}^J$ and $\{\hat{\psi}^{\sigma^*(j)}\}_{j=1}^J$;
- 11 Backpropagate \mathcal{L}_H ;

synthesis. STaR [17] jointly optimizes the parameters of two Neural Radiance Fields and a set of rigid poses to decompose a dynamic scene into two constituent parts. Impressively, Block-NeRF [18] demonstrates the possibility to scale NeRF to render city-scale scenes spanning multiple blocks. With such great advances in existing NeRF-related technologies, researchers recently have explored representing the scene with NeRFs in robotic applications [10, 11, 19]. Following these works, we discuss object localization within the NeRF scenes in this paper, in the hope of facilitating the downstream robotic applications with NeRF representations.

Object Localization. Object localization is a key component for robotics applications including autonomous driving, indoor navigation, and robot manipulation. Most previous object localization methods can be divided into three main categories according to the input modality types: point-cloud based, stereo images based, and monocular image based. The point-cloud based methods [20–23] directly acquire the coordinates of the points on the surfaces of objects in 3D space. These methods generally work on the point clouds obtained from hardware sensors, e.g., LiDAR. Despite good performance, the costly depth sensor is not always available for a robotic system. Stereo image based methods [24] leverage the geometric structures obtained from the disparities between the stereo image pair. Although stereo-based methods get rid of the costly depth sensors, a significant length of rig baseline is required for accurate depth inference, which impedes the application to compact devices. The monocular methods [25–29]

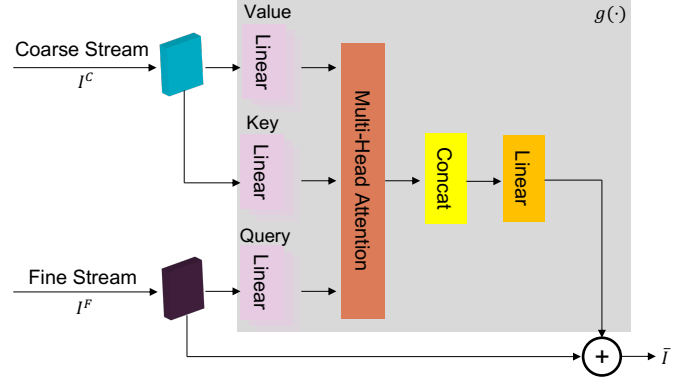


Fig. 3: Cross-attention Fusion Module fuses abstraction-levels of coarse and fine context.

thus become popular for object localization in the community given the portable and low-cost nature. However, none of these methods can be directly applied to NeRF-based scene representation.

Implicit Representations for Robotics. Adamkiewicz *et al.* [10] propose a trajectory planning method that plans full, dynamically feasible trajectories to avoid collisions with a NeRF environment. Li *et al.* [19] combine NeRF and time contrastive learning to learn viewpoint-invariant 3D-aware scene representations, which enables visuomotor control for challenging manipulation tasks. Dex-NeRF [11] leverages NeRF’s view-independent learned density, and performs a transparency-aware depth-rendering to grasp transparent objects. Lin *et al.* [30] propose to learn dense object descriptors from NeRFs and use an optimized NeRF to extract dense correspondences between multiple views of an object. In contrast to previous work, we find the compact representation of NeRF (which can be zoomed in and out freely) is particularly useful for 3D object localization, which can further aid downstream robotics applications such as navigation and manipulation.

III. METHOD

A. Preliminary

NeRF [7] is a differentiable implicit function that represents a continuous 3D scene. The implicit function is usually implemented with Multi-Layer Perceptrons (MLPs) $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$, which maps the 3D location $\mathbf{x} = (x, y, z)$ and 2D viewing direction $\mathbf{d} = (\theta, \phi)$ to an emitted color value \mathbf{c} and a volume density value σ . Based on this representation, the pixel color can be obtained via volume rendering [31]:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^K T_k (1 - \exp(-\sigma_k(t_{k+1} - t_k))) \mathbf{c}_k, \quad (1)$$

$$\text{where } T_k = \exp\left(-\sum_{k' < k} \sigma_{k'}(t_{k'+1} - t_{k'})\right),$$

where $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ denotes a ray cast from the camera center \mathbf{o} along the direction \mathbf{d} passing through the rendering pixel. T_k here can be interpreted as the probability that the ray is

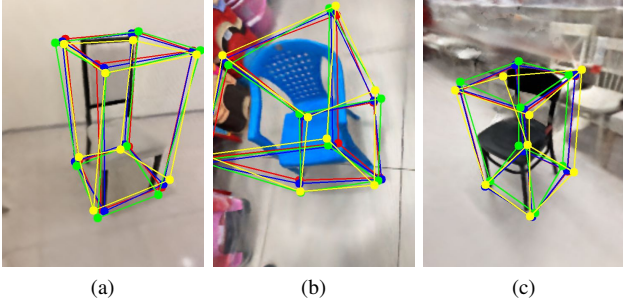


Fig. 4: Qualitative results of NeRF-Loc on the NeRFLocBench validation split. The detected objects are shown with 3D bounding boxes. Groundtruths are labeled with red while the predictions from NeRF samples $\hat{\mathbf{X}}$ are labeled with blue, the predictions from rendered color images $\hat{C}(\mathbf{r})$ are labeled with green, the predictions from rendered depth maps $\hat{D}(\mathbf{r})$ are labeled with yellow.

not interrupted by before and successfully transmits to point $\mathbf{r}(t_k)$. Similarly, the expected depth $\hat{D}(\mathbf{r})$ where the camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ terminates can be calculated by replacing the color value c_k with the sampling distances t_k :

$$\hat{D}(\mathbf{r}) = \sum_{k=1}^K T_k (1 - \exp(-\sigma_k(t_{k+1} - t_k))) t_k. \quad (2)$$

B. Problem Formulation

We define the *object localization within Neural Radiance Fields* task as follows: Given a pre-constructed NeRF environment F_Θ and an observation pose $\mathbf{P} \in SE(3)$ inside, we design a transformer network G_Φ to estimate the 3D bounding box \hat{B} and category \hat{p} of J objects in the current view:

$$\{(\hat{B}^j, \hat{p}^j)\}_{j=1}^J = G_\Phi(\mathbf{P}; F_\Theta). \quad (3)$$

Here, the bounding box \hat{B} is parameterized as its corners $(\hat{x}^b, \hat{y}^b, \hat{z}^b)$: $\hat{B} = \{(\hat{x}_i^b, \hat{y}_i^b, \hat{z}_i^b)\}_{i=1}^{N_c}$ ($N_c = 8$ in our case).

C. NeRF-Loc

Inspired by the effectiveness of multi-view information [32] and the outstanding performance of transformer-based methods in localization [33, 34], we designed a transformer-based framework to efficiently utilize information from multiple views. Fig. 2 shows an overview of the proposed framework. The pipeline consists of the following steps: 1) Given an observation pose $\mathbf{P} \in SE(3)$ and the camera intrinsic matrix \mathbf{K} , the field values (i.e. colors and densities) on the eye beams emitted from the camera center are sampled; 2) The sampled field values are then sent to a transformer-based coarse encoder and fine encoder for feature extraction. 3) These encoded features are thereafter fused with an attention fusion module to complement each other and alleviate ambiguity. 4) Finally, the fused features are sent to the transformer-based decoder to predict the bounding-box corners and categories.

1) *Fine Stream and Coarse Stream*: To localize an object in the scene, the network not only should focus on the object itself but also should leverage the helpful context information around. Following this intuition, we design two parallel branches, the fine stream and the coarse stream, to focus on the object details and the context respectively.

Given an observation pose $\mathbf{P} \in SE(3)$ and the camera intrinsics $\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$, the rays from the camera center passing through the image plane are chosen for sampling the radiance field. Specifically, each ray direction \mathbf{d} is related to the focal length f and the intersection points (x, y) on the image plane:

$$\mathbf{d}(x, y, f) = \frac{[x - p_x, y - p_y, f]^T}{\sqrt{(x - p_x)^2 + (y - p_y)^2 + f^2}}. \quad (4)$$

We apply camera intrinsic matrices with two different focal lengths, f/δ and f , where $\delta > 1$, for the coarse stream and the fine stream respectively. In our case, we set $\delta = 1.5$. In this way, different sampling scopes are adopted for different streams according to Equ. (4). The coarse stream essentially has larger field of view while the fine stream has more detailed views. Empirically, we find the two-stream encoding significantly improves the localization performance.

After having the sampling rays mentioned above, we sample N equally-spaced points on each ray and collect the field values (c, σ) on these points. The sampled coarse point sets $\hat{\mathbf{X}}^C$ and fine point sets $\hat{\mathbf{X}}^F$ are modulated by projection layers $l^C(\cdot)$ and $l^F(\cdot)$ respectively, and then sent to the coarse and fine transformer encoders $E^C(\cdot)$ and $E^F(\cdot)$ for further processing:

$$I^F = E^F(l^F(\hat{\mathbf{X}}^F)), \quad (5)$$

$$I^C = E^C(l^C(\hat{\mathbf{X}}^C)). \quad (6)$$

After the processing, the coarse embeddings I^C and fine embeddings I^F are obtained.

2) *Cross-attention Fusion*: To make better use of wide-view information and fine-grained information, we introduce Cross-attention Fusion, a lateral connections between the coarse stream and fine stream, to fuse the embeddings from the fine stream and the coarse stream. The fine stream embeddings I^F and coarse stream embeddings I^C are calculated through a lightweight cross-attention head $g(\cdot)$. Then, the selected information is fused with the original fine-grained information via skip connection as Equ. (7) shows. We hope that such a design could learn and benefit from both fine-grained contexts and coarse-grained contexts at the abstraction level.

$$\bar{I} = I^F + g(I^F, I^C). \quad (7)$$

3) *Decoder*: The fusion embeddings \bar{I} are passed through the transformer decoder network $D(\cdot)$, and J localization proposals are obtained from MLP heads, as expressed by Equ. (8).

$$\{(\hat{B}^j, \hat{p}^j)\}_{j=1}^J = D(\bar{I}, q), \quad (8)$$

TABLE I: Performance comparison of our approach and baselines. Both fine and coarse information is used for baselines.

Method IoU	mAP@ (%)									Average
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	
3DETR [33]	80.21	78.32	74.89	65.55	59.12	49.98	40.01	21.97	0.23	52.25
NeRF-Loc (Fine-only)	98.24	98.24	98.24	96.50	91.49	78.47	52.86	22.45	0.55	70.78
NeRF-Loc (Coarse-only)	95.50	95.02	94.20	92.41	88.29	75.71	47.68	20.04	0.25	67.67
NeRF-Loc	99.22	98.30	98.30	97.65	87.92	80.77	60.54	23.81	1.70	72.02

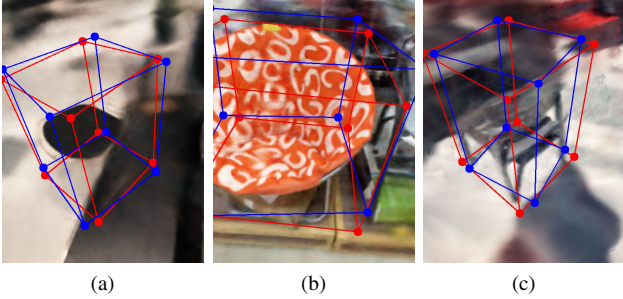


Fig. 5: Failure cases on NeRF-LocBench validation split. In a) Prediction error is caused by too vague rendering results. In b) Incomplete object leads to a failure case, c) The foreground and background look similar, which are quite difficult for 3D object localization.

q is the learnable query in the shape of J which is randomly initialized.

D. Loss Functions

From the embeddings output from the decoder, 3D bounding boxes corners are predicted by MLP regression head \hat{B} and the corresponding categories \hat{p} are predicted by the MLP classification head. The optimal match is computed between the augmented ground-truth $\psi^{(j)}$ and prediction $\hat{\psi}^{(j)}$. We search for the optimal permutation σ^* among the set of all permutations, that has the lowest matching cost \mathcal{L}_{match} .

$$\sigma^* = \arg \min_{\sigma \in \Sigma_J} \sum_{j=1}^J \mathcal{L}_{match}(\psi^j, \hat{\psi}^{\sigma(j)}). \quad (9)$$

This cost is computed efficiently via the Hungarian algorithm. \mathcal{L}_{box} is a weighted combination between the IoU loss \mathcal{L}_{iou} [35] and ℓ_1 loss, weighted by scalar hyperparameters λ_{iou} and λ_{ℓ_1}

$$\mathcal{L}_{box} = \lambda_{iou} \mathcal{L}_{iou}(B^j, \hat{B}^{\sigma^*(j)}) + \lambda_{\ell_1} \ell_1(B^j, \hat{B}^{\sigma^*(j)}). \quad (10)$$

We use cross-entropy loss \mathcal{L}_{CE} as classification objective. The Hungarian matching loss \mathcal{L}_{match} is a sum of the classification and regression loss $\mathcal{L}_{match}(\psi^n, \hat{\psi}^{\sigma^*(n)}) = \mathcal{L}_{box} + \mathcal{L}_{CE}$. After obtaining the optimal permutation σ^* based on the lowest \mathcal{L}_{match} , we compute the Hungarian loss \mathcal{L}_H for this optimal matching. \mathcal{L}_H over all the matched pairs of proposals is defined as:

$$\mathcal{L}_H = \sum_{j=1}^J \mathcal{L}_{match}(\psi^j, \hat{\psi}^{\sigma^*(j)}). \quad (11)$$

NeRF-Loc is trained end-to-end, with \mathcal{L}_H as its objective. The full NeRF-Loc learning pipeline is illustrated as Alg. 1.

IV. EXPERIMENTAL RESULTS

Imagine in the future, we first synthesize a scene with NeRF, after which we perform robot navigation and manipulation in it, requiring only NeRF representation and no dependence on other data. This is the application scenario suitable for the tasks we define. Thus, after the neural radiation world is built, we are interested in using the learned neural representations for object localization, without keeping the previous NeRF training data (RGB images and corresponding camera poses).

We aim to answer the following questions in our experiments: (i) Can we learn to predict object bounding boxes in Neural Fields? How does NeRF-Loc compare to existing methods? (ii) Is the raw representation of NeRF better than the other representations? To answer the first question, we build a challenging NeRF representation benchmark NeRF-LocBench for our task by training NeRF functions based on a real-world dataset. We show that our approach outperforms existing approaches. We answer the second question with ablation studies on different input modalities.

A. Dataset

To train and test our proposed model, we build a benchmark NeRF-LocBench based on Objectron [12], which is a large-scale dataset consisting of $\sim 15k$ continuous videos (about 4M images) with frame-wise annotations. The dataset is split as $\sim 3k$ validation and $\sim 12k$ training videos. Each video has longer clips averaging a duration of ~ 15 seconds. Such a long duration makes it a suitable dataset to train NeRF models and test NeRF-Loc. The NeRF-LocBench is used in all experiments, consisting of coarse and fine-grained NeRF samples \hat{X} trained using NeRF [7], rendered color image $\hat{C}(\mathbf{r})$, rendered depth image $\hat{D}(\mathbf{r})$ and corresponding 3D bounding box annotations. We train our model on the training split, compare it with other methods on validation split and report mean Average Precision (mAP).

B. Baselines

Many previous 3D object localization methods rely on CAD models, which are not required by us. 3DETR [33] is an end-to-end Transformer based object detection model. We modify it to take the NeRF samples as input and predict the 3D bounding boxes as output. We evaluate the capacity of our

TABLE II: Ablation study of different modalities. Both fine and coarse information is used. $\hat{D}(\mathbf{r})$: rendered depth image, $\hat{C}(\mathbf{r})$: rendered color image, $\hat{\mathbf{X}}$: NeRF raw samples.

Modality IoU	mAP@ (%)									Average
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	
$\hat{C}(\mathbf{r})$	81.92	74.57	68.50	60.21	49.07	31.53	15.62	5.55	0.17	43.01
$\hat{D}(\mathbf{r})$	78.51	69.66	63.82	49.23	33.74	16.69	5.49	0.45	0.00	35.28
$\hat{\mathbf{X}}$	99.22	98.30	98.30	97.65	87.92	80.77	60.54	23.81	1.70	72.02
$\hat{\mathbf{X}} + \hat{C}(\mathbf{r})$	98.29	98.22	97.66	96.74	93.11	81.32	63.33	25.92	0.46	72.78
$\hat{\mathbf{X}} + \hat{D}(\mathbf{r})$	97.81	96.68	95.75	93.08	84.97	75.68	51.47	19.41	1.05	68.43
$\hat{C}(\mathbf{r}) + \hat{D}(\mathbf{r})$	99.34	99.29	98.61	96.85	94.02	78.84	39.04	5.36	0.00	67.92
$\hat{\mathbf{X}} + \hat{C}(\mathbf{r}) + \hat{D}(\mathbf{r})$	97.91	97.42	94.03	90.60	86.05	78.21	57.09	25.12	1.84	69.80

TABLE III: Ablation study of different fusion modules.

Method IoU	mAP@ (%)									Average
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	
MLP	98.95	97.99	97.99	97.99	88.08	72.04	39.79	14.13	1.45	67.60
Attention	99.22	98.30	98.30	97.65	87.92	80.77	60.54	23.81	1.70	72.02

model in 3D object localization in neural fields and compare it with the baselines and ablations.

C. Implementation Details

We initialize the Coarse and Fine streams of our network from scratch and follow an end-to-end training process. We use $\delta = 1.5$, $N_c = 8$, $J = 100$ in our experiments. The number of layers of transformer encoder and decoder $L_e^F = L_e^C = L_d = 4$. For the feature projection layer $l(\cdot)$, we use 3 layers of MLPs to map the input to 256-dim. Both streams are trained together for 500 epochs with a batch size of 8 and a cosine dynamic learning rate scheduler of $1e-6$ at the start, $5e-4$ as the base learning rate, and 9 warm-up epochs with warm-up learning rate of $1e-6$. We train our model with a machine using NVIDIA TITAN Xp GPU and Intel Xeon CPU.

D. Evaluating NeRF-Loc on NeRFLocBench

First, we evaluate NeRF-Loc on NeRFLocBench by only taking NeRF samples as input to verify that we can predict object bounding boxes in neural fields.

In Fig. 4, we visualize some examples of the predicted bounding boxes results on NeRFLocBench. Our model is able to predict the bounding box of objects with the correct position. We compare the performance of the NeRF-Loc with 3DETR on the 3D object localization task in Tab. I. For this evaluation, we report the performance (mAP) at different IoUs. The comparisons in Tab. I show that our approach performs better than the previous methods. Our approach is more effective than other methods when the mAP is at high IoU. Considering the mAP of high IoUs (higher than 0.6), one can see that object localization on neural fields benefits a lot from the multi-scale feature. Although our model outperforms the dual-view baseline using only the fine view, the proposed Coarse-Fine architecture further improves it by a large margin (from 70.78% to 72.72%).

E. Ablation Study

1) *Modality*: We compare multiple modality combinations: (i) NeRF samples $\hat{\mathbf{X}}$, (ii) rendered color image $\hat{C}(\mathbf{r})$, and

(iii) rendered depth image $\hat{D}(\mathbf{r})$, and the permutation of these three modalities. Both fine and coarse views are used for all modality combinations. We find that using only the NeRF samples is better than using the rendered color image or the rendered depth image individually (see Tab. II). This indicates that using NeRF samples are more effective than the other two representations and facilitates faster and better localization. The average mAP of $\hat{\mathbf{X}} + \hat{C}(\mathbf{r})$ is slightly better than using $\hat{\mathbf{X}}$ only. Considering the time spent on rendering, the obtained NeRF samples are sufficient for downstream tasks. There is a limited necessity to spend further time on rendering color or depth maps.

2) *Fusion Type*: We have also tried MLP fusion layers: simply passing I^C and I^F through 2 layers of MLPs after stitching them in the last dimension to get \tilde{I} . As Tab. III shows, we can see that our cross-attention fusion module works much better than the MLP fusion module. The difference is more significant especially when the IoU threshold is high. This is probably due to the fact that the attention mechanism is more sensitive to fine-grained features.

V. CONCLUSION

We propose the task of object localization in a neural radiance world, which enables autonomous agents to perceive under implicit representation, understand where the goal is, and used it for downstream tasks such as planning and control. We presented NeRF-Loc, a framework for 3D object localization in neural fields, which can take advantage of the ease of scaling up and down with the NeRF representation. We introduced the Cross-attention Fusion to best combine the coarse stream with the fine stream. NeRF-Loc obtains the best known performance on NeRFLocBench. Our work can be seen as a step towards the goal of enabling autonomous agents to find the target location from NeRF world and plan for complex tasks like navigation and manipulation. In future work, we intend to further explore object localization tasks in more complex NeRF scenarios with multiple objects.

REFERENCES

- [1] Mohit Shridhar et al. “Alfred: A benchmark for interpreting grounded instructions for everyday tasks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10740–10749.
- [2] Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. “Moca: A modular object-centric approach for interactive instruction following”. In: *ICCV* (2020).
- [3] Jiankai Sun, De-An Huang, Bo Lu, Yun-Hui Liu, Bolei Zhou, and Animesh Garg. “PlaTe: Visually-grounded planning with transformers in procedural tasks”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4924–4930.
- [4] Alexander Pashevich, Cordelia Schmid, and Chen Sun. “Episodic Transformer for Vision-and-Language Navigation”. In: *ICCV* (2021).
- [5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.
- [6] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. “Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 14540–14549.
- [7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”. In: *ECCV*. 2020.
- [8] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. “iMAP: Implicit mapping and positioning in real-time”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6229–6238.
- [9] Zihan Zhu et al. “Nice-slam: Neural implicit scalable encoding for slam”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12786–12796.
- [10] Michal Adamkiewicz et al. “Vision-only robot navigation in a neural radiance world”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4606–4613.
- [11] Jeffrey Ichnowski*, Yahav Avigal*, Justin Kerr, and Ken Goldberg. “Dex-NeRF: Using a Neural Radiance field to Grasp Transparent Objects”. In: *Conference on Robot Learning (CoRL)*. 2020.
- [12] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. “Objectron: A Large Scale Dataset of Object-Centric Videos in the Wild with Pose Annotations”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2021).
- [13] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. “In-Place Scene Labelling and Understanding with Implicit Scene Representation”. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2021.
- [14] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. “pixelNeRF: Neural Radiance Fields from One or Few Images”. In: *CVPR*. 2021.
- [15] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. “Sharf: Shape-conditioned Radiance Fields from a Single View”. In: *ICML*. 2021.
- [16] Michael Niemeyer and Andreas Geiger. “GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2021.
- [17] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. “STaR: Self-supervised Tracking and Reconstruction of Rigid Objects in Motion with Neural Rendering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13144–13152.
- [18] Matthew Tancik et al. “Block-NeRF: Scalable Large Scene Neural View Synthesis”. In: *CVPR* (2022).
- [19] Yunzhu Li, Shuang Li, Vincent Sitzmann, Pulkit Agrawal, and Antonio Torralba. “3D Neural Scene Representations for Visuomotor Control”. In: *arXiv preprint arXiv:2107.04004* (2021).
- [20] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. “Multi-view 3d object detection network for autonomous driving”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 1907–1915.
- [21] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. “Frustum pointnets for 3d object detection from rgb-d data”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 918–927.
- [22] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. “Pointnet: 3d object proposal generation and detection from point cloud”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 770–779.
- [23] Hongwei Yi et al. “Segvoxelnet: Exploring semantic context and depth-aware features for 3d vehicle detection from point cloud”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 2274–2280.
- [24] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. “3d object proposals using stereo imagery for accurate object class detection”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.5 (2017), pp. 1259–1272.
- [25] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. “Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes”. In: *RSS* (2018).
- [26] Zhigang Li, Gu Wang, and Xiangyang Ji. “CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 7677–7686. DOI: 10.1109/ICCV.2019.00777.
- [27] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. “Gs3d: An efficient 3d object detection framework for autonomous driving”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1019–1028.
- [28] Chen Wang et al. “DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion”. In: *Computer Vision and Pattern Recognition (CVPR)* (2019).
- [29] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. “CamNet: Coarse-to-Fine Retrieval for Camera Re-Localization”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.
- [30] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. “NeRF-Supervision: Learning Dense Object Descriptors from Neural Radiance Fields”. In: *IEEE Conference on Robotics and Automation (ICRA)*. 2022.
- [31] Marc Levoy. “Efficient ray tracing of volume data”. In: *ACM Transactions on Graphics (TOG)* 9.3 (1990), pp. 245–261.
- [32] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. “Cross-view semantic segmentation for sensing surroundings”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4867–4873.
- [33] Ishan Misra, Rohit Girdhar, and Armand Joulin. “An End-to-End Transformer Model for 3D Object Detection”. In: *ICCV*. 2021.
- [34] Jiankai Sun, Bolei Zhou, Michael J Black, and Arjun Chandrasekaran. “LocATe: End-to-end Localization of Actions in

3D with Transformers”. In: *arXiv preprint arXiv:2203.10719* (2022).

- [35] Generalized Intersection Over Union. “A Metric and a Loss for Bounding Box Regression”. In: *Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA*. 2019, pp. 658–666.