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

Jiankai Sun, Yan Xu, Mingyu Ding, Hongwei Yi, Chen Wang, Jingdong Wang, *Fellow, IEEE*, Liangjun Zhang, and Mac Schwager, *Member, IEEE*

***Abstract*—Neural Radiance Fields (NeRFs) have become a widely-applied scene representation technique in recent years, showing advantages for robot navigation and manipulation tasks. To further advance the utility of NeRFs for robotics, 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, oriented 3D bounding boxes of objects as output. Using current NeRF training tools, a robot can train a NeRF environment model in real-time and, using our algorithm, identify 3D bounding boxes of objects of interest within the NeRF for downstream navigation or manipulation tasks. Concretely, we design a pair of paralleled 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 conventional RGB(-D) based methods that take rendered RGB images and depths from NeRFs as inputs. Our method is better than the baselines.**

***Index Terms*—Object localization, object detection, neural radiance field (NeRF).**

I. Introduction

WHEN a robot enters a novel environment, it needs to first perceive the surrounding objects and understand their spatial relationships, so that it can navigate toward objects of interest or avoid objects that may present a threat. Similarly, for manipulation tasks, a robot must first detect the object it intends to manipulate, and determine its pose relative to the robot. Accurate object localization in 3D space thus becomes a fundamental problem in robotics. The choice of object localization strategy depends on the underlying map representation [1], [2]. Compared with representing the scene with a set of 2D images, 3D maps maintain 3D topology and richer geometric information. Moving further towards high-fidelity 3D environment representations, in this letter, we propose a method to detect 3D object bounding boxes directly within a NeRF environment model.

Point cloud or voxel-based map representations have significant limitations. These classic 3D representations generally lack radiance (i.e., color and lighting) details, which are useful 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) [3], and successfully deploy it in real-time robot SLAM [4], [5] systems. With modern NeRF training tools, a robot can build a high-fidelity NeRF model of its environment, and use this as a map representation for downstream navigation and manipulation tasks. NeRF is a continuous scene representation where the 3D scene is compressed in feed-forward neural networks that memorize the 3D spatial density and radiance fields. Compared

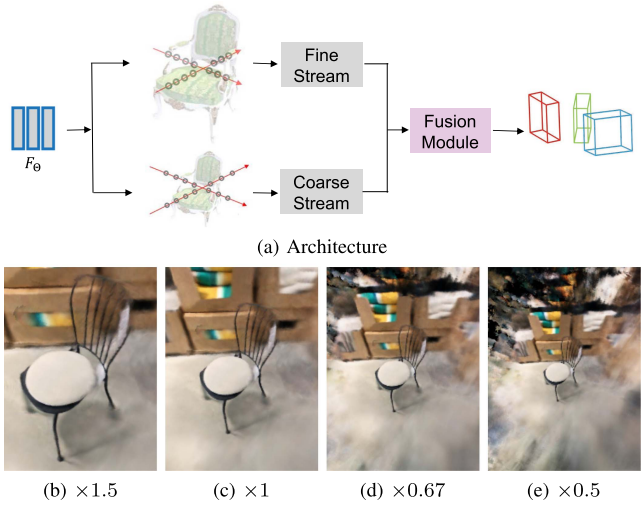


Fig. 1. (a) NeRF-Loc networks process information at multiple scales. The coarse stream learns to sample the most informative positions in the overview image, while the fine stream processes the near-field image to extract fine-grained information. These two streams are connected by an attention fusion 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.

Manuscript received 10 March 2023; accepted 19 June 2023. Date of publication 7 July 2023; date of current version 14 July 2023. This letter was recommended for publication by Associate Editor E. Kayacan and Editor A. Faust upon evaluation of the reviewers' comments. This work was supported by ONR under Grant N00014-18-1-2830. *(Jiankai Sun and Yan Xu contributed equally to this work.) (Corresponding author: Jiankai Sun.)*

Jiankai Sun, Chen Wang, and Mac Schwager are with the School of Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: jksun@stanford.edu; chenwj@stanford.edu; schwager@stanford.edu).

Yan Xu is with the Chinese University of Hong Kong, Hong Kong (e-mail: yanxu@link.cuhk.edu.hk).

Mingyu Ding is with the UC Berkeley, Berkeley, CA 94720 USA (e-mail: myding@berkeley.edu).

Hongwei Yi is with the Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany (e-mail: hongwei.yi@tuebingen.mpg.de).

Jingdong Wang is with the Baidu Inc., Sunnyvale, CA 94089 USA (e-mail: wangjingdong@outlook.com).

Liangjun Zhang is with the Robotics and Autonomous Driving Lab, Baidu Research, Sunnyvale, CA 94089 USA (e-mail: liangjun.zhang@gmail.com).

Digital Object Identifier 10.1109/LRA.2023.3293308

with classic point cloud and voxel representations, NeRF is more compact and contains more photo-realistic elements. Other motivations for utilizing NeRF in the field of robotics include the potential benefits it offers in terms of enabling end-to-end robot learning, acting as a robust simulation platform, and enhancing robot performance by training in simulated NeRF environments. Nevertheless, conventional object localization methods cannot be applied directly to NeRFs, and object localization in NeRFs has not been well-studied in the literature. Object detection is crucial for specifying objectives for robotic motion planning [6] and manipulation [7], which have already been demonstrated with NeRFs. Hence, object detection in NeRFs has the potential to benefit robot autonomy stacks based on NeRF environment representations. Our approach hopes to bridge this gap.

Specifically, we study *3D object localization in neural fields*, 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 a 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 (Fig. 1). 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. Our model takes in a previously unseen pre-trained NeRF model and camera view, and outputs labeled 3D bounding boxes for the objects in that NeRF.

Currently, there is a scarcity of datasets for our task, which needs to have both NeRF representations and 3D bounding box annotations. Objectron [8] 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 benchmark `NeRFLocBench`based on Objectron [8] and evaluate our method with it. We show that our method outperforms 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 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 [3] 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 [9] extends NeRF to jointly encode 2D semantics with appearance and geometry. There are also some few-shot NeRFs [10] 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 [11] incorporates a compositional 3D scene representation into the generative model which leads to more controllable image synthesis. STaR [12] 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 [13] 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 [6], [7], [14]. Following these works, we discuss object localization within the NeRF scenes in this letter, in the hope of facilitating 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 [15], [16], [17], [18], [19], [20], [21], [22], [23]

Algorithm 1: 3D Object Localization on Neural Fields (Training Process).

Input: NeRF function F_Θ, camera pose $\mathbf{P}$, coarse intrinsics K^C and fine intrinsics K^F, ground truth $\psi = \{B, p\}$, where B represents the bounding box instances and p represents the classes.

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.

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

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 Time of Flight (ToF) sensors. 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. The monocular methods [25], [26], [27], [28] become popular for object localization in the community given the portable and low-cost nature. However, our approach is specialized for NeRF-based scene representation.

Implicit Representations for Robotics: Adamkiewicz et al. propose a trajectory planning method that plans full, dynamically feasible trajectories to avoid collisions with a NeRF environment [6]. Li et al. [14] combine NeRF and time contrastive learning to learn viewpoint-invariant 3D-aware scene representations, which enables visuomotor control for challenging manipulation tasks. Dex-NeRF [7] leverages NeRF's view-independent learned density, and performs a transparency-aware depth-rendering to grasp transparent objects. Lin et al. [29] propose to learn dense object descriptors from NeRFs and use an optimized NeRF to extract dense correspondences between multiple views of an object. LENS [30], NeRF-Pose [31], Loc-NeRF [32], and iNeRF [33] proposed camera localization algorithms based on NeRF representation. Their ideas have the potential to be applied to object pose estimation but may be prone to errors given complex environments and multi-object cases. 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. In particular, unlike LENS [30], our detection framework can do multi-object detection.

III. Method

A. Preliminary

NeRF [3] is a differentiable implicit function that represents a continuous 3D scene. The implicit function is usually implemented with Multi-Layer Perceptrons (MLPs) $F_\Theta : (\boldsymbol{x}, \boldsymbol{d}) \rightarrow (\boldsymbol{c}, \sigma)$, which maps the 3D location $\boldsymbol{x} = (x, y, z)$ and 2D viewing direction $\boldsymbol{d} = (\theta, \phi)$ to an emitted color value $\boldsymbol{c}$ and a volume density value σ. Based on this representation, the pixel color can be obtained via volume rendering [34]:

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

where $T_k = \exp(-\sum_{k'<k} \sigma_{k'}(t_{k'+1} - t_{k'}))$, $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ denotes a ray cast from the camera center $\boldsymbol{o}$ along the direction $\boldsymbol{d}$ passing through the rendering pixel. T_k here can be interpreted as the probability that the ray is not interrupted before and successfully transmits to point $\boldsymbol{r}(t_k)$. Similarly, the expected depth $\hat{D}(\boldsymbol{r})$ where the camera ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ terminates can be calculated by replacing the color value $\boldsymbol{c}_k$ with the sampling distances t_k:

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

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:

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

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 [35] and the outstanding performance of transformer-based methods in localization [36], [37], 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 rays 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 e.g., scenes, or information around objects. 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 $\boldsymbol{d}$ is related to the focal length f and the intersection points (x, y) on the image plane:

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

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 (4). The coarse stream essentially has a larger field

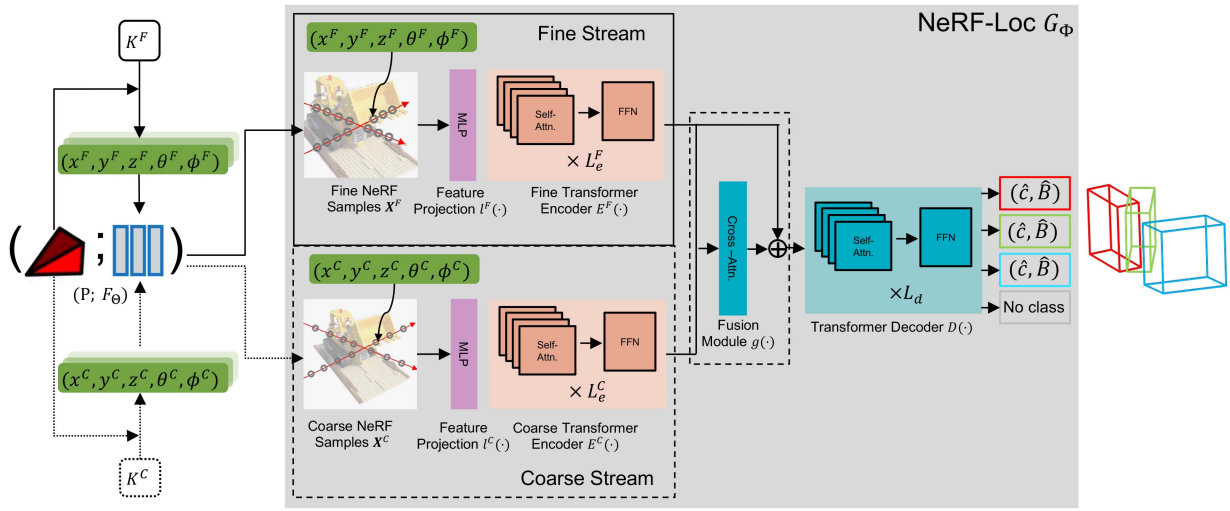


Fig. 2. Framework. Given the observation camera pose $\mathbf{P} \in SE(3)$, we sample a set of field values along view rays ((x, y, z)'s) and aim to localize the objects based on these samples. Our framework contains a fine stream and a coarse stream. The two streams share similar transformer architectures but are dedicated to processing NeRF samples of different fields of view (controlled by intrinsics K^C and K^F). The samples from two streams are encoded as high-dimensional embeddings separately before being fused together by the cross-attention-based Fusion Module. The fused features are decoded by the Transformer Decoder. Finally, the 3D bounding boxes, and the object categories are predicted by MLP heads.

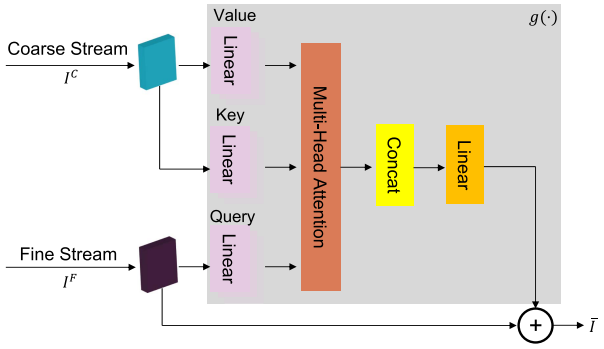


Fig. 3. Cross-attention Fusion Module fuses abstraction levels of coarse and fine context.

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 $(\mathbf{c}, \sigma)$ on these points. The sampled coarse point sets $\hat{X}^C$ and fine point sets $\hat{X}^F$ 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:

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

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

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 connection between the coarse stream and fine stream, to fuse the embeddings from the fine stream and the coarse stream (Fig. 3). 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 (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.

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

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 (8).

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

q is the learnable query in the shape of J which is randomly initialized. Please refer to [38] for an explanation of the usage of q, which is a commonly used design in transformers.

D. Loss Functions

From the embeddings output from the decoder, 3D bounding box 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 $\sigma*$ among the set of all permutations, that has the lowest matching cost $\mathcal{L}_{match}$.

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

This cost is computed efficiently via the Hungarian algorithm. $\mathcal{L}_{box}$ is a weighted combination between the IoU loss $\mathcal{L}_{iou}$ [39] and ℓ_1 loss, weighted by scalar hyperparameters λ_{iou} and λ_{ℓ_1}

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

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 $\sigma*$ 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:

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

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

IV. Experimental Results

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?

A. Dataset

To train and test our proposed model, we build a benchmark `NeRFLocBench`based on Objectron [8]. 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`networks. The `NeRFLocBench`consists of coarse and fine-grained NeRF samples $\hat{X}$ trained using NeRF [3], rendered color image $\hat{C}(\boldsymbol{r})$, rendered depth image $\hat{D}(\boldsymbol{r})$ and corresponding 3D bounding box annotations.

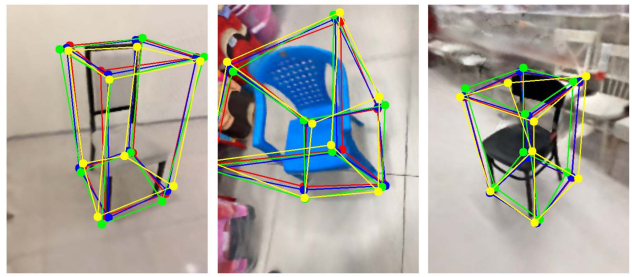
B. Baselines

Many previous 3D object localization methods rely on CAD models, which are not required by us. We evaluate the capacity of our model in 3D object localization in neural fields and compare it with the multiple baselines and ablations: Among them, 3DETR [36] is a transformer-based object detection model. CDPN [26] estimates 6-DoF object pose estimation from RGB images. DETR3D [40] performs 3D object detection from multi-view images.

To enable fair comparison, we modify the above baselines to take the NeRF samples as input and predict the 3D bounding boxes as output.

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 a warm-up learning rate of $1e-6$. 240×180 directions are sampled and 64 points along a pixel ray are used for forming an X for both training and inference. We train our model with a machine using NVIDIA TITAN Xp GPU and Intel Xeon CPU.

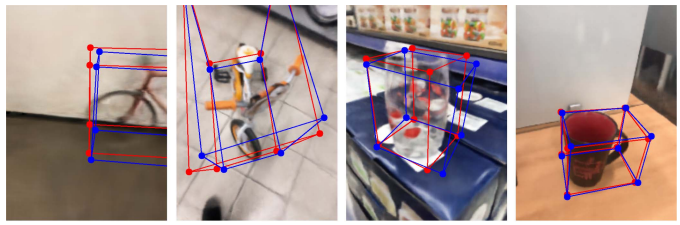


(a) (b) (c)

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

(a) (b) (c)

Fig. 5. Failure cases. (a) Prediction error is caused by severely blurred rendering. (b) The object occlusion leads to a failure case. (c) The color similarity between the foreground and background poses difficulties for object localization.



(a) (b) (c) (d)

Fig. 6. Qualitative results of unseen scenes and novel views. (a) (b) are unseen scenes and (c) (d) are unseen views.

D. Evaluating `NeRF-Loc`

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 Figs. 4, 5, and 6, 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 recent methods on the 3D object localization task in Table I . For this evaluation, we report the performance (mAP) at

TABLE I
PERFORMANCE COMPARISON OF OUR APPROACH AND BASELINES

Algorithm Category	Method	mAP@IoU (%)			Average
		0.1	0.5	0.9	
Single-view	3DETR [36]	78.15	56.96	0.11	50.20
	CDPN [26]	95.01	83.39	0.49	66.28
	NeRF-Loc (Fine-only)	98.24	**91.49**	0.55	70.78
	NeRF-Loc (Coarse-only)	95.50	88.29	0.25	67.67
Multi-view	3DETR [36]	80.21	59.12	0.23	52.25
	DETR3D [40]	97.42	86.28	0.21	67.84
	NeRF-Loc	**99.22**	87.92	**1.70**	**72.02**

For single-view, the fine view is used for baselines. For multi-view, both fine and coarse information is used for baselines.

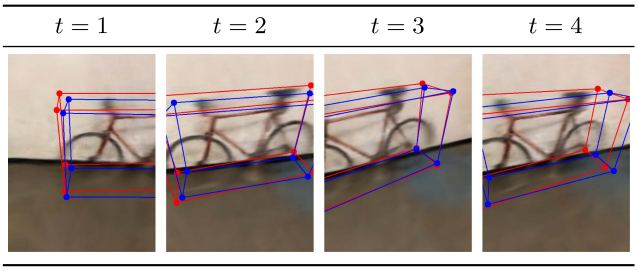


Fig. 7. Qualitative results of object tracking in NeRF representations without the need for removing the background.

different IoUs. The comparisons in Table I show that our approach performs better than the previous methods and multi-view input has higher performance than single-view input. Our approach is more effective than other methods when the mAP is at high IoU. Sometimes NeRF is not trained well enough (e.g., it is rendered blurry in Fig. 5), which can lead to poor detection. Our choice of coarse- and fine-grained scale (e.g., the field of view does not contain sufficient information about the object) can also affect performance. Although our model outperforms the multi-view baselines using only the fine view, the proposed Coarse-Fine architecture further improves it by a large margin (from 70.78% to 72.72%).

E. Tracking With NeRF Representation

We show the qualitative results of pose tracking in Fig. 7. Once we got the pre-trained NeRF, at each time step t, NeRF-Locestimates the object's bounding box, without the need to remove the background.

F. Ablation Study

1) Modality: We compare multiple modality combinations: (i) NeRF samples $\hat{\boldsymbol{X}}$, (ii) rendered color image $\hat{C}(\boldsymbol{r})$, and (iii) rendered depth image $\hat{D}(\boldsymbol{r})$, 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 Table II). This indicates that NeRF samples are more effective than the other two representations and facilitate faster and better localization. The average mAP of $\hat{\boldsymbol{X}} + \hat{C}(\boldsymbol{r})$ is slightly better than using $\hat{\boldsymbol{X}}$ 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.

TABLE II
ABLATION STUDY OF DIFFERENT MODALITIES

Modality IoU	mAP@ (%)			Average
	0.1	0.5	0.9	
$\hat{C}(\boldsymbol{r})$	81.92	49.07	0.17	43.01
$\hat{D}(\boldsymbol{r})$	78.51	33.74	0.00	35.28
$\hat{\boldsymbol{X}}$	99.22	87.92	1.70	72.02
$\hat{\boldsymbol{X}} + \hat{C}(\boldsymbol{r})$	98.29	93.11	0.46	**72.78**
$\hat{\boldsymbol{X}} + \hat{D}(\boldsymbol{r})$	97.81	84.97	1.05	68.43
$\hat{C}(\boldsymbol{r}) + \hat{D}(\boldsymbol{r})$	**99.34**	**94.02**	0.00	67.92
$\hat{\boldsymbol{X}} + \hat{C}(\boldsymbol{r}) + \hat{D}(\boldsymbol{r})$	97.91	86.05	**1.84**	69.80

Both fine and coarse information is used. $\hat{D}(\boldsymbol{r})$: rendered depth image, $\hat{C}(\boldsymbol{r})$: rendered color image, $\hat{\boldsymbol{X}}$:NeRF raw samples.

TABLE III
ABLATION STUDY OF DIFFERENT FUSION MODULES

Method IoU	mAP@ (%)			Average
	0.1	0.5	0.9	
MLP	98.95	**88.08**	1.45	67.60
Attention	**99.22**	87.92	**1.70**	**72.02**

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 $\bar{I}$. As Table 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. DISCUSSION, LIMITATIONS, AND CONCLUSIONS

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. One of the limitations of our work is that we assume that the NeRF model is already pre-trained. The NeRF model usually takes time for training and has difficulties being directly applied to real-time systems. With the progress of technology, the training time of NeRF has been greatly reduced. 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.

REFERENCES

[1] K. P. Singh, S. Bhambri, B. Kim, R. Mottaghi, and J. Choi, "Moca: A modular object-centric approach for interactive instruction following," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2020, pp. 1888–1897.
[2] J. Sun, D.-A. Huang, B. Lu, Y.-H. Liu, B. Zhou, and A. Garg, "PlaTe: Visually-grounded planning with transformers in procedural tasks," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4924–4930, Apr. 2022.
[3] 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 *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

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

[5] Z. Zhu et al., "Nice-slam: Neural implicit scalable encoding for slam," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12786–12796.

[6] M. Adamkiewicz et al., "Vision-only robot navigation in a neural radiance world," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4606–4613, Apr. 2022.

[7] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-NeRF: Using a neural radiance field to grasp transparent objects," in *Proc. Conf. Robot Learn.*, 2020.

[8] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, "Objectron: A large scale dataset of object-centric videos in the wild with pose annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7822–7831.

[9] S. Zhi, T. Laidlow, S. Leutenegger, and A. Davison, "In-place scene labelling and understanding with implicit scene representation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15838–15847.

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

[11] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11453–11464.

[12] W. Yuan, Z. Lv, T. Schmidt, and S. Lovegrove, "Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13144–13152.

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

[14] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3D neural scene representations for visuomotor control," in *Proc. Conf. Robot Learn.*, 2022, pp. 112–123.

[15] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1907–1915.

[16] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.

[17] S. Shi, X. Wang, and H. Li, "PointrCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.

[18] H. Yi et al., "SegVoxelNet: Exploring semantic context and depth-aware features for 3D vehicle detection from point cloud," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2274–2280.

[19] C. Wang et al., "Densefusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3343–3352.

[20] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3D keypoints voting network for 6dof pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11632–11641.

[21] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "FFB6D: A full flow bidirectional fusion network for 6D pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3003–3013.

[22] Y. Di et al., "GPV-Pose: Category-level object pose estimation via geometry-guided point-wise voting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6781–6791.

[23] W. Peng, J. Yan, H. Wen, and Y. Sun, "Self-supervised category-level 6D object pose estimation with deep implicit shape representation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 2082–2090.

[24] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals using stereo imagery for accurate object class detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1259–1272, May 2018.

[25] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. Robot.: Sci. Syst.*, 2018.

[26] Z. Li, G. Wang, and X. Ji, "Cdpn: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7677–7686.

[27] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "GS3D: An efficient 3D object detection framework for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1019–1028.

[28] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "CamNet: Coarse-to-fine retrieval for camera re-localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2871–2880.

[29] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "NeRF-Supervision: Learning dense object descriptors from neural radiance fields," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 6496–6503.

[30] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "LENS: Localization enhanced by NeRF synthesis," in *Proc. Conf. Robot Learn.*, 2021, pp. 1347–1356.

[31] F. Li, H. Yu, I. Shugurov, B. Busam, S. Yang, and S. Ilic, "NeRF-pose: A first-reconstruct-then-regress approach for weakly-supervised 6D object pose estimation, 2022, *arXiv:2203.04802*.

[32] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-NeRF: Monte Carlo localization using neural radiance fields," in *Proc. IEEE Int. Conf. Robot. Automat.*, London, U.K., 2023, pp. 4018–4025, doi: 10.1109/ICRA48891.2023.10160782.

[33] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1323–1330.

[34] M. Levoy, "Efficient ray tracing of volume data," *ACM Trans. Graph.*, vol. 9, no. 3, pp. 245–261, 1990.

[35] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4867–4873, Jul. 2020.

[36] I. Misra, R. Girdhar, and A. Joulin, "An End-to-End Transformer Model for 3D Object Detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2906–2917.

[37] J. Sun, B. Zhou, M. J. Black, and A. Chandrasekaran, "Locate: End-to-end localization of actions in 3D with transformers, 2022, *arXiv:2203.10719*.

[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2020, pp. 213–229.

[39] G. I. O. Union, "A metric and a loss for bounding box regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 658–666.

[40] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "DETR3D: 3D object detection from multi-view images via 3D-to-2D queries," in *Proc. Conf. Robot Learn.*, 2022, pp. 180–191.