2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10610092

Neural Rearrangement Planning for Object Retrieval from Confined Spaces Perceivable by Robot's In-hand RGB-D Sensor

Hanwen Ren and Ahmed H. Qureshi

***Abstract*— Rearrangement planning for object retrieval tasks from confined spaces is a challenging problem, primarily due to the lack of open space for robot motion and limited perception. Several traditional methods exist to solve object retrieval tasks, but they require overhead cameras for perception and a time-consuming exhaustive search to find a solution and often make unrealistic assumptions, such as having identical, simple geometry objects in the environment. This paper presents a neural object retrieval framework that efficiently performs rearrangement planning of unknown, arbitrary objects in confined spaces to retrieve the desired object using a given robot grasp. Our method actively senses the environment with the robot's in-hand camera. It then selects and relocates the non-target objects such that they do not block the robot path homotopy to the target object, thus also aiding an underlying path planner in quickly finding robot motion sequences. Furthermore, we demonstrate our framework in challenging scenarios, including real-world cabinet-like environments with arbitrary household objects. The results show that our framework achieves the best performance among all presented methods and is, on average, two orders of magnitude computationally faster than the best-performing baselines.**

I. Introduction

Target object retrieval from unknown confined spaces is critical for robots intending to assist people in their daily lives [1]. For instance, robots aiding at hospitals will often have to retrieve the required medicines from cabinets. At factory floors, this could involve fetching tools from toolboxes. Additionally, these robot skills are also desirable for search and rescue at disaster sites to help the affected people [2]. However, object retrieval from unknown confined spaces imposes significant challenges. First, scene observation is more complex in these settings than in traditional tabletop environments due to the heavy occlusion between objects, imperfect light conditions, and limited camera view angles. Second, the robot needs to clear the pathway to the target by relocating other objects in the scene. Third, robot grasping also imposes challenges as oftentimes only limited stable grasp candidates are available. One of the challenging components of rearrangement planning for object retrieval, besides its NP-hard complexity [3], is that the algorithm needs to explicitly select path-blocking objects from a given observation and relocate them to a feasible region so that the target can be accessed [4]. The existing approaches in task and motion planning (TAMP) [5] mainly try to solve this problem using tree-search-based methods [6] until the path to the target is obtained. However, these methods are computationally expensive; therefore, they are inapplicable for real-time applications requiring fast solutions [7]. In addition, current approaches [8]–[10] tend to model the problem in the 2D plane and assume the scene is fully observed or can be captured with a single topdown viewpoint.

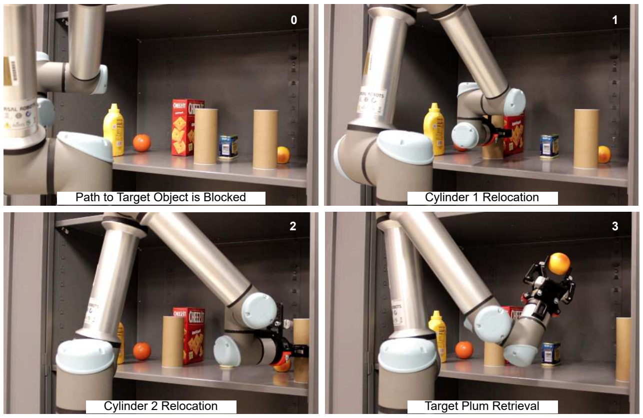

Fig. 1: Execution for retrieving the target object ("plum"): The robot's pathway to the target object is blocked in the initial setup. After executing the object manipulation plan from our method of relocating the pathway-blocking objects, in this case, the cylinders, the robot arm finally retrieves the plum from the confined cabinet environment.

Inspired by recent developments of deep learning and their application in various rearrangement planning problems [11], [12], this paper presents a neural rearrangement planning approach for object retrieval from unknown confined spaces. Our system gathers observations of the unknown, confined environment with an in-hand camera via the active sensing approach [13] which aims to perceive the underlying space using the minimal number of camera viewpoints. Once the target object is detected and is unretrievable, our framework switches to the rearrangement phase for making the pathway to reach the target object. During rearrangement, our method iteratively selects and relocates non-target objects within a confined space until the target object is kinematically reachable for the robot arm without collision. Using an in-hand camera makes it even more challenging to access confined spaces without collisions. In summary, the main contribution and salient features of our approach include the following:

- A novel object selection framework that learns the

*This work was supported by the National Science Foundation (NSF) under award no. 2204528.

Hanwen Ren and Ahmed H. Qureshi are with the Department of Computer Science, Purdue University, West Lafayette, IN, USA, 47907. Email: {ren221, ahqureshi}@purdue.edu

nature of robot movement in confined environments and chooses the object that has the highest chance of blocking the robot's way of reaching the desired target.
- A novel region proposal framework that is aware of robot path homotopy to the target object and, therefore, selects the relocation region for the given object such it will no longer block robot pathways.
- A new data generation strategy capturing the robot path homotopy to given targets for training the object selection and region proposal network.
- A unified framework for efficiently solving object retrieval tasks from unknown confined spaces with demonstrations in complex real-world cabinet-like scenarios.

II. Related Work

The first kind of object retrieval task is performed in cluttered tabletop, open scenes [9], [14], [15]. For instance, [14] presents an object searching and retrieving system that leverages dynamically controlled sampling-based algorithms and extends to different robot tasks such as grasping, relocating, and sorting. Another approach [10] brings up a system that retrieves the desired object based on natural language instructions. However, the open space and fully observed environment settings make it much easier for the robot to fulfill the objective successfully.

Many existing approaches [4], [16] also try to perform object retrieval in confined spaces. For instance, [8] uses non-prehensile actions such as pushing to clear up the space and grasp the target objects in cluttered environments. By utilizing non-prehensile grasp actions, the robot can interact with multiple objects simultaneously while eliminating the heavy computation involved in prehensile, pick-place, operations. Another approach [17] aims to minimize the number of objects to relocate by choosing modified Vector Field Histogram-plus (VFH+) [18] as a local planner to relocate non-target object at each step until a collision-free path to the target is obtained. In [19], the authors bring up a lazy object rearrangement planner that bypasses the extensive motion planning and collision-checking queries but only checks collision when a solution is found. However, all the methods mentioned above plan the object retrieval task in the fully observed 2D space consisting of identical objects.

Aside from directly solving the object retrieval tasks, other relevant works treat it as an instance in the rearrangement planning problem. For instance, several studies have been conducted in object rearrangement planning tasks with specific goal configurations in tabletop environments using tree-search [20], [21] and randomized algorithms [22], [23]. Similarly, [24] solves this problem by utilizing MCTS to avoid the expensive computation involved in traditional search-based methods. Works like [11], [12], [25] use deep neural networks to tackle different sub-tasks engaged in the process. Some of those methods also generalize to real-world scenarios with never-before-seen objects. However, these methods, due to the tabletop environment settings, cannot be deployed directly in confined spaces where object reachability, grasping, and robot motion planning induce several challenges for finding rearrangement solutions for object retrieval.

III. Proposed Method

This section formally presents our Neural Object Retrieval framework comprising the objective function, neural models, algorithm pipeline, and other implementation details. The framework structure is shown in Fig. 2.

A. Problem Definition

Let a given scene be denoted as S whose outline dimensions d_x, d_y, and d_z along x, y, and z axes, respectively, are assumed to be known. The scene S comprises the observed space denoted as S_o and an unobserved space denoted as $S_{ou} = S \backslash S_o$. A robot manipulator system situated at location $l_r \in \mathbb{R}^3$ with an RGB-D camera attached to its end-effector actively senses the given scene until the scene is fully observed, i.e., $S_o \approx S$. In the remainder of the paper, we use the notation $A_{\{B\}}$ to represent any arbitrary list A containing $B \in \mathbb{N}$ number of elements. Once the scene is observed, we extract the observed and unobserved regions in the environment, denoted as $l_{\{b\}} = \{l_0, l_1, \cdots, l_b\}$. Each region is $n \times n$ grid on the given environment's ground surface. The $l_i \in l_{\{b\}}$ is a 4D vector,(x, y, z, flag), with x,y, and z corresponding to the spatial center position of the region and the flag indicating if that region was observed (flag $= 1$) or unobserved (flag $= 0$). The robot system configuration space is denoted as $\mathcal{Q}$ with obstacle and obstacle-free space indicated as $\mathcal{Q}_{free}$ and $\mathcal{Q}_{obs} = \mathcal{Q} \backslash \mathcal{Q}_{free}$, respectively. Given a collision-free robot configuration, $q \in \mathcal{Q}_{free}$, the corresponding gripper pose is indicated as $x_g \in SE(3)$.

The observed scene contains $m+1 \in \mathbb{N}$ objects, comprising non-target objects $o_{\{m\}} = \{o_1, o_2, ..., o_m\}$ and a target object o^* that is to be retrieved by the robot system. The scene state $S_o(t)$, at time t, indicates the state of objects $o_{\{m+1\}}$. The state of each object o is a 6D vector, i.e., $(cx_o, cy_o, cz_o, dx_o, dy_o, dz_o)$, where the first three values capture the geometric center while the remainder denotes the dimension of its tightest axis-aligned bounding box. Let an indicator function $I(o^*, x_g^*, S_o(t))$ output 1 when the target object, o^*, is kinematically reachable without collision using the given grasp pose x_g^*; otherwise, 0 for the given scene state $S_o(t)$ at time t.

We assume environment settings that are confined and cluttered and require a robot to rearrange non-target objects $o_{\{m\}}$ for reaching and retrieving the target object o^* without collision. Therefore, a policy π takes the scene observation $S_o(t)$ and robot state, at time t, selects an object $o_i' \in o_{\{m\}}$, whose current location is $l_{o_i'}^s \in \mathbb{R}^3$, and proposes an alternative placement $l_{o_i'}^g \in l_{\{b\}}$ for the selected object to clear a way to retrieve the target object o^*. Let function $d(l_r, l_{o_i'}^s, l_{o_i'}^g)$ represent the total Euclidean distance from robot base position l_r to selected object location $l_{o_i'}^s$ and further to the placement region $l_{o_i'}^g$. The objective of our proposed work is to find an optimal policy

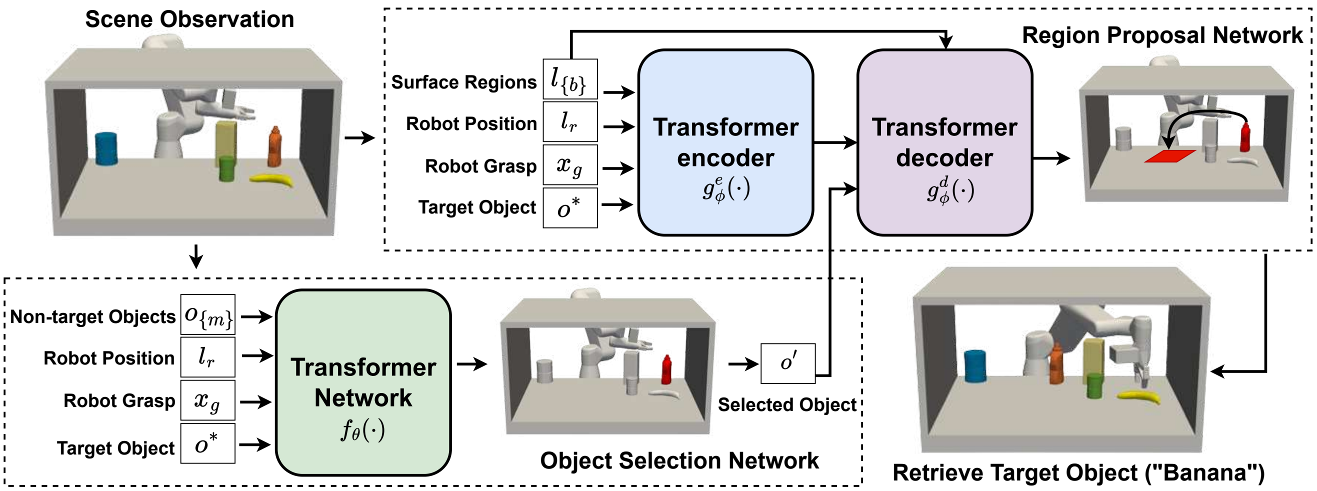
Scene Observation
Region Proposal Network
Surface Regions $l_{\{b\}}$
Robot Position l_r
Robot Grasp x_g
Target Object o^*
Transformer encoder $g^e_\phi(\cdot)$
Transformer decoder $g^d_\phi(\cdot)$
Non-target Objects $o_{\{m\}}$
Robot Position l_r
Robot Grasp x_g
Target Object o^*
Transformer Network $f_\theta(\cdot)$
o'
Selected Object
Object Selection Network
Retrieve Target Object ("Banana")

Fig. 2: Neural Object Retrieval: The task is to retrieve the yellow object (Banana). Our main modules include the object selection and region proposal network. We do not show MLPs that encode the given inputs for brevity. Given the scene observation via active sensing, the object selection network selects the non-target object for rearrangement. The chosen object o' is indicated in red in the bottom scene image. The region proposal network proposes the best placement region for the selected object to clear the pathway for target object retrieval. The proposed placement region on the environment surface is marked red on the top right scene image. Our robot moves the selected object to its new placement and retrieves the target object if possible otherwise repeats the object rearrangement process in the confined environment.

π^* that finds the shortest rearrangement sequence $\tau = \{(o'_1, l^s_{o'_1}, l^g_{o'_1}), (o'_2, l^s_{o'_2}, l^g_{o'_2}), ..., (o'_T, l^s_{o'_T}, l^g_{o'_T})\}$ to retrieve the target object o^* using the given robot gripper pose x^*_g, i.e.,

$$\pi^* = \arg\max_\pi \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}(I(o^*, x^*_g, S_o(t)) - d(l_r, l^s_{o'_t}, l^g_{o'_t}))\right] \tag{1}$$

Therefore once the above objective function is optimized, an optimal policy will enable the target object retrieval by rearranging non-target objects with a minimum total move distance d in the confined environments.

B. Scene observation

Since it is difficult to place an overhead camera in confined spaces to obtain observation, we utilize an active sensing (AS) approach [13] that uses a robot with an in-hand camera to generate the scene perception. The scene is initially unknown except for its dimensions (d_x, d_y, d_z). The AS method selects the best camera viewpoint sequences and registers all views into a unified scene observation. We run the AS module until the scene is fully observed and extract all the observed objects $o_{\{m+1\}} = \{o_1, o_2, ..., o_m, o^*\}$ using scene segmentation [26]. Each object is represented with a 6D vector comprising its geometric center and bounding box dimensions. Such 6D object representation is more compact and computationally efficient than directly processing raw object point clouds using methods like PointNet++ [27]. Finally, we also utilize the scene dimensions to determine the observed and unobserved regions $l_{\{b\}} = \{l_1, l_2, ..., l_b\}$ in the given scene.

C. Neural Object Selection

We aim for a function that at each time step t takes the scene observation $S_o(t)$, comprising objects $o_{\{m+1\}}$, and selects an object, $o' \in o_{\{m\}}$, that blocks the way for a homotopy of paths towards the target object, o^*, from the robot's current state. Directly running sampling-based motion planners (SMPs) [28] can be time-consuming due to the low sampling efficiency caused by the strict collision constraints. Therefore, checking for the path homotopy is crucial as it allows flexibility in choosing the underlying motion planner and increases its chances of successfully finding a path solution toward the given target.

Thus, we approach this problem by designing a neural network to learn the underlying nature of the homotopy paths and help the system quickly figure out the path-blocking objects. The neural network we proposed is called Object Selection Network (OSNet), which can be viewed as a function f_θ, with parameters θ, that outputs a categorical probability distribution $\hat{p}_{\{m\}}$ over the non-target objects, representing the probability of each object blocking the way for path homotopy. The input to the OSNet is the current scene information consisting of objects state $(o_{\{m\}}, o^*)$, robot location l_r, and desired gripper pose x_g to grasp the target object, i.e.,

$$\hat{p}_{\{m\}} \leftarrow f_\theta(\alpha_\theta(l_r), \gamma_\theta(x_g), \alpha_\theta(o^*), \alpha_\theta(o_{\{m\}})) \tag{2}$$

where m is the number of non-target objects, $l_r \in \mathbb{R}^3$, and $x_g \in SE(3)$. To process the given scene information, we use two different multi-layer perceptrons (MLP) [29], α_θ, γ_θ to embed robot l_r plus objects $o_{\{m+1\}}$ and the gripper x_g, resulting in $m + 3$ latent embeddings. All MLP latent embeddings form the input tokens for the multi-headed, self-attention Transformer network [30], which captures the relationship between each embedding through its self-attention mechanism and outputs new $m+3$ representations. We extract the non-target m objects' embeddings from the new representations and pass them through another MLP to predict the probability, $\hat{p} \in [0, 1]$, of selecting each object for rearrangement. In summary, the function f comprises two MLPs for encoding inputs, a Transformer Network for capturing input embedding relationships through self-attention, and a decoder MLP for predicting the selection probabilities for non-target objects.

We train function f_θ end-to-end in a supervised manner using the Mean Squared Error (MSE) between the predicted, $\hat{p}_{\{m\}}$, and ground truth, $p_{\{m\}}$, probabilities. The ground truth, $p_{\{m\}}$, is a categorical distribution that is generated by leveraging the RRTConnect motion planner [31] to select the object blocking the path homotopy. We run RRTConnect followed by path smoothing to compute multiple path to the target o^* without considering collisions with other objects. Each non-target object receives a count whenever it blocks a path and the object with the maximum blocking counts should be the one to be relocated. Finally, the ground truth categorical distribution $p_{\{m\}}$ is formed by normalizing all the objects' blocking counts. During execution, we greedily select the object, $o' \in o_{\{m\}}$, with the maximum predicted probability in $\hat{p}_{\{m\}}$ for rearrangement.

D. Neural Rearrangement Region Proposal

After selecting the object $o' \in o_{\{m\}}$ to rearrange, the next step is to decide its placement region, $l^g_{o'} \in l_{\{b\}}$, in the confined environment. The placement region for the selected object should be the nearest observed, kinematically reachable place that does not block the path homotopy towards the target. To achieve the above objectives, we propose a Region Proposal Network (RPNet), a neural function, g_ϕ, with an encoder-decoder structure, comprising parameters ϕ, that efficiently selects a feasible region for the placement of the selected object. Our architecture comprises three different MLPs, α_ϕ, γ_ϕ, and κ_ϕ, to embed the robot location l_r, the target (o^*) and selected object (o') states, the desired gripper pose x_g, and the placement regions $l_{\{b\}}$, respectively. These MLPs are shared among our encoder, g^e_ϕ, and decoder, g^d_ϕ, modules. The encoder network is a multi-headed, self-attention Transformer Network that takes the MLP embedding of the robot's current location, target object, and placement regions, and outputs the latent representations Z^e, i.e.,

$$Z^e \leftarrow g^e_\phi\big(\alpha_\phi(l_r), \gamma_\phi(x_g), \alpha_\phi(o^*), \kappa_\phi(l_{\{b\}})\big) \tag{3}$$

Therefore, the encoder encapsulates the object retrieval problem and its relation to all placement regions via self-attention. The decoder network is also a multi-headed, self-attention Transformer network, that takes the encoder network output, Z^e, and latent queries. The latent queries are formed by obtaining the MLP embeddings of the placement regions, $\kappa_\phi(l_{\{b\}})$, and adding each embedding with the MLP embedding of the selected object $\alpha_\phi(o')$. We assume the addition in latent space will enable our framework to foresee the impact of placing object o' at a certain placement region $l \in l_{\{b\}}$. The latent output representation of all regions from the decoder network is sent to a MLP to obtain the final cost values, $\hat{c}_{\{b\}}$, indicating how well placement region and selected object pairs will aid in the target object retrieval, i.e.,

$$\hat{c}_{\{b\}} \leftarrow g^d_\phi\big([\alpha_\phi(o') + \kappa_\phi(l_0)], \cdots, [\alpha_\phi(o') + \kappa_\phi(l_b)], Z^e\big) \tag{4}$$

We train RPNet in a supervised manner. During the training phase, the network's objective is to minimize the MSE loss between the ground truth costs $c_{\{b\}}$ and the predicted $\hat{c}_{\{b\}}$ for a given object o'. Following the above-mentioned criteria, the occupied, unobserved, and robot path homotopy blocking regions get the maximum cost. For others, their cost values are assigned based on the Euclidean distance to the selected object's initial location. Once g_ϕ is trained using the ground truth labels for the variety of scenarios, we use it to select the placement region with the minimum cost for placing the selected object o'.

E. Full Pipeline Algorithm

Our neural rearrangement planning for object retrieval from unknown confined spaces works as follows. The inputs to our framework include the robot's location l_r, a grasp pose x_g for retrieving the target object, target and non-target objects' states $o_{\{m+1\}}$, and environment surface regions $l_{\{b\}}$. The active sensing module takes the environment dimensions (d_x, d_y, d_z) and constructs the environment to generate the object states $o_{\{m+1\}}$ and surface regions $l_{\{b\}}$. The grasp pose is obtained using GraspNet [32], and the robot base location l_r is fixed and assumed to be known. Given the inputs, an indicator function leveraging the RRT-Connect detects if the target object is reachable by the robot without collision. If not, the algorithm enters the while loop, where it greedily selects and rearranges the non-target objects if such relocation actions are feasible until the pathway toward the target becomes clear, or the loop limit is achieved. A non-target object $o' \in o_{\{m\}}$ is selected for rearrangement using our OSNet output probabilities $p_{\{m\}}$. The argmax over $o_{\{m\}}$ returns the object with maximum selection probability. Once the object o' is selected, the RPNet proposes the region $l^g_{o'} \in l_{\{b\}}$ for its placement. The region with minimum cost is selected using argmin over $l_{\{b\}}$. If the path from the selected object's current state $l^s_{o'}$ to propose state $l^g_{o'}$ exists, the robot performs the rearrangement action. Finally, when the path toward the target object is clear, the robot retrieves the object o^* with the given grasp pose; otherwise, our method reports failure when the loop limit is reached.

IV. Results & Discussions

In this section, we present the results and analysis of the following evaluation experiments: 1) a comparison experiment evaluating the performance of the proposed neural

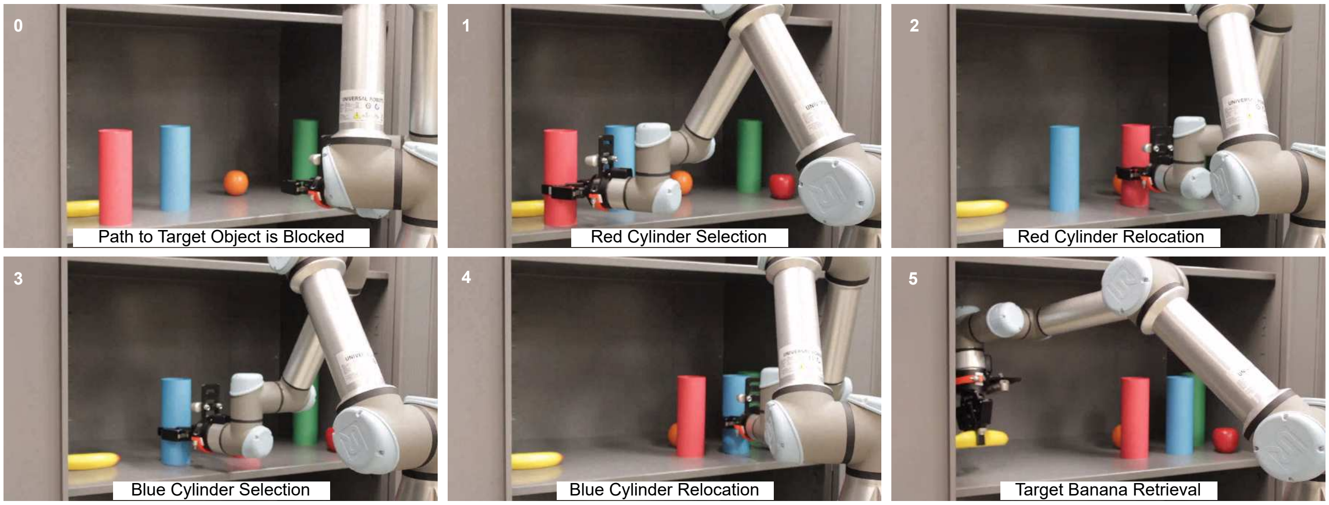

Fig. 3: Execution for retrieving the yellow target object ("banana"): In the initial setup, the target object is not retrievable as other objects block it. The robot clears the pathway by moving two cylindrical objects (frames 1-4) and then finally take the pathway going through the back of all objects to the target object. It can also be seen that confined spaces impose significant challenges in robot motion, especially when retrieving an object with a relatively lower height, such as a banana, than other objects.

	Object Retrieval Planner	**Metrics**			
		Success Rate (%) ↑	Objects Rearranged ↓	Planning Time (s) (%) ↓	Workspace Moving Distance (m) ↓
Comparison	Random Planner	45%	1.689 ± 0.962	0.001 ± 0.001	1.97 ± 1.204
	Local Planner	55 %	1.745 ± 0.976	3.307 ± 1.669	1.62 ± 0.734
	Neural Planner (Ours)	72 %	1.722 ± 0.803	0.014 ± 0.006	1.647 ± 0.856
Ablation	OSNet-only Planner	71%	1.662 ± 0.903	2.987 ± 1.636	2.074 ± 1.185
	RPNet-only Planner	47 %	1.638 ± 0.783	0.01 ± 0.005	1.808 ± 0.913

TABLE I: The table above shows the statistical comparison between various object rearrangement planners. Overall, our neural planner achieves the highest success rate while keeping the minimum planning times. The bottom table is the result of our ablation study. The higher performance of our neural framework than its ablations validates the effectiveness of our object selection (OSNet) and region proposal network (RPNet) in solving object retrieval tasks.

rearrangement planning method against the baselines in unknown confined environments; 2) an ablation study showing the effectiveness of our various neural functions involved in the system; 3) sim-to-real transfer demonstrating our approach's performance in different novel real-world cabinet settings. We use the following metrics for quantitative evaluation.

- **success rate**: It tracks the percentage of cases in the test set where the robot successfully retrieved the target object within the limit of 5 steps. An object selection and relocation is considered as 1 step, so the robot had the limit not to select and relocate more than 5 objects.
- **object rearranged**: It represents the number of objects rearranged before the target can be successfully retrieved.
- **planning time**: It stores all the time spent selecting the object to be arranged and the time consumed in determining the best region for the selected object's relocation.
- **workspace moving distance**: It shows the total distance the robot arm's end-effector moves to relocate various objects before successfully retrieving the target.

A. Baselines

We create three baseline planners to compare with our method in 100 unseen dynamically generated testing environments filled with different numbers of objects from the YCB dataset [33]. All methods start with the scene observation given by the active sensing module comprising object states and surface regions. At the beginning of each step, we run RRTConnect with a fixed time budget of one second as the initial attempt to reach the target object. If it fails, the first baseline, the random planner, randomly selects a non-target object and places it at a randomly selected valid collision-free region. The second baseline is a greedy local planner, greedily relocating all the non-target objects blocking the linear, straight-line path between the robot's current and final states for reaching the target object. The placement region is also chosen greedily based on the nearest collision-free spot next to the selected object, which does not block the linear path trajectory. The baselines and our approach share a maximum step number of 5 before declaring failure. Therefore, the results in Table I are only analyzed for successful experiments for each method based on the metrics mentioned above.

1) Success rate: Overall, the proposed neural method achieves the best success rate among all methods. For the local planner, it tries to remove the objects that block the linear interpolated robot arm path. However, when reaching the target, the arm uses RRTConnect to compute the final trajectory instead of following this linear path because of its often infeasibility in the confined environment setting. As a result, the relocated object may not be optimal for the RRTConnect path planner, which could be one of the reasons for the relatively low success rate. On the other hand, the random planner does not follow any heuristic that can guide it to clear the objects; hence, it mostly fails to retrieve the target objects.

2) Planning time: We introduced neural networks to the object retrieval problem mainly because of their generalization power and fast execution speed at evaluation time. Hence, once trained, our neural model can quickly predict the possible object rearrangement sequences with few forward propagation passes of scene information through our OSNet and RPNet. From Table I, we can see the large planning time gap between our neural method and the local planner. This is because of the potential path calculation and the robot arm trajectory mesh generation for collision checking at every possible placement region. For the random planner, randomly choosing objects and regions is quick, but the success rate is sacrificed.

3) number of object rearranged & moving distance: Due to the confined environment settings, the difference between the number of objects rearranged and the workspace move distance is not apparent. Our neural planner shared similar values with the baselines. However, since we only show the data analysis for the successful experiments, the small move distance and rearrangement steps for random and greedy planners are mainly caused by many failure cases that proposed relocating more than four objects.

B. Ablation Studies

In this section, we conduct ablation studies to show the importance and effectiveness of our OSNet and RPNet. All experiments are performed on the same 100 dynamic environments used in the previous section. The results are summarized in Table I.

1) OSNet-only Planner: In the first study, we replaced our RPNet with the analytical selection module used in the local planner. The results show that the OSNet-only planner still maintains a decent success rate. However, the planning time increases almost a hundred times. Thus, this validates that our RPNet can provide reasonable regions for relocating selected objects and saves significant computational overload when compared to classical methods.

2) RPNet-only Planner: In the second study, the OSNet is substituted by the classical object selection module from the local planner. According to the results, in the RPNet-only planner, the success rate drops to 47%. This evidence shows that our OSNet leads to a significant performance gain over a classical greedy method.

In summary, our ablation study shows that the ablated models either perform poorly in computational speed or exhibit a lower success rate than our proposed framework, validating the need for both OSNet and RPNet for solving object retrieval tasks.

C. Real World Experiments

Finally, we perform a series of real-world experiments in a confined cabinet with dimensions of (56 cm, 86 cm, 50 cm). We directly deploy our neural models trained in the simulated environments to the real robot and object retrieval setup. Two successful trials can be seen in Fig. 1 and Fig. 3. For the experiment in Fig. 3, the target is placed on the left side next to the environment boundary, far away from the initial posing of the robot gripper. We also place two cylinder blocks around the target object to ensure the robot can not directly reach it. The robot starts by performing the active sensing pipeline to understand the initially unknown environment. During the grasping phase, the robot arm first picks the red cylinder directly in front of the target (Fig. 3, frame 1) and rearranges it to an adjacent region. In the second step, the robot arm chooses the blue cylinder (Fig. 3, frame 3) that blocks the path toward the target and relocates it to a vacant area. At last, after swinging the forearm inside the cabinet, the target object is successfully retrieved by the robot. From the robot's final pose, we can clearly see the reason for choosing the two objects. On the other hand, for the experiment shown in Fig. 1, because of the limited feasible space on the left part of the scene, the planner relocates one of the blocking objects to the left while another to the right near the boundary. Eventually, the target object plum is successfully retrieved.

Note that the real experiment exhibits the generalizability of our proposed approach to real-world scenarios with direct sim2real transfer. Thus, the successful execution of our approach in these setups validates the effectiveness of our proposed framework.

V. Conclusions & Future Work

This paper presents a neural rearrangement planning method for object retrieval tasks from unknown confined spaces. Our approach can generate a sequence of object selection and their alternative placements in confined spaces to successfully retrieve the given target object. We demonstrate the generalization of the proposed approach to complex real-world scenarios without additional training. Furthermore, the results also show that our framework presents the best performance among all baselines and saves significant computation times in solving object retrieval tasks. Our method achieves such high performance by ensuring the relocation of non-target objects clears the way for the robot path homotopy to the given target object, thus significantly increasing the underlying motion planner's efficiency and chances of finding the required robot motion sequences in confined spaces.

References

[1] K. Yamazaki, R. Ueda, S. Nozawa, M. Kojima, K. Okada, K. Matsumoto, M. Ishikawa, I. Shimoyama, and M. Inaba, "Home-assistant robot for an aging society," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2429–2441, 2012.

[2] C. Schlenoff and E. Messina, "A robot ontology for urban search and rescue," in *Proceedings of the 2005 ACM workshop on Research in knowledge representation for autonomous systems*, 2005, pp. 27–34.

[3] G. Wilfong, "Motion planning in the presence of movable obstacles," in *Proceedings of the fourth annual symposium on Computational geometry*, 1988, pp. 279–288.

[4] J. Ahn, J. Lee, S. H. Cheong, C. Kim, and C. Nam, "An integrated approach for determining objects to be relocated and their goal positions inside clutter for object retrieval," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6408–6414.

[5] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.

[6] T. Ren, G. Chalvatzaki, and J. Peters, "Extended tree search for robot task and motion planning," *arXiv preprint arXiv:2103.05456*, 2021.

[7] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots," *Journal of Intelligent Manufacturing*, vol. 33, no. 2, pp. 387–424, 2022.

[8] E. R. Vieira, K. Gao, D. Nakhimovich, K. E. Bekris, and J. Yu, "Persistent homology guided monte-carlo tree search for effective nonprehensile manipulation," *arXiv preprint arXiv:2210.01283*, 2022.

[9] B. Huang, S. D. Han, J. Yu, and A. Boularias, "Visual foresight trees for object retrieval from clutter with nonprehensile rearrangement," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 231–238, 2021.

[10] T. Nguyen, N. Gopalan, R. Patel, M. Corsaro, E. Pavlick, and S. Tellex, "Robot object retrieval with contextual natural language queries," *arXiv preprint arXiv:2006.13253*, 2020.

[11] A. H. Qureshi, A. Mousavian, C. Paxton, M. C. Yip, and D. Fox, "Nerp: Neural rearrangement planning for unknown objects," *arXiv preprint arXiv:2106.01352*, 2021.

[12] A. Goyal, A. Mousavian, C. Paxton, Y.-W. Chao, B. Okorn, J. Deng, and D. Fox, "Ifor: Iterative flow minimization for robotic object rearrangement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 787–14 797.

[13] H. Ren and A. H. Qureshi, "Robot active neural sensing and planning in unknown cluttered environments," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2738–2750, 2023.

[14] K. Ren, L. E. Kavraki, and K. Hang, "Rearrangement-based manipulation via kinodynamic planning and dynamic planning horizons," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1145–1152.

[15] Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato, "Online planning for target object search in clutter under partial observability," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8241–8247.

[16] S. H. Cheong, B. Y. Cho, J. Lee, C. Kim, and C. Nam, "Where to relocate?: Object rearrangement inside cluttered and confined environments for robotic manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7791–7797.

[17] J. Lee, Y. Cho, C. Nam, J. Park, and C. Kim, "Efficient obstacle rearrangement for object manipulation tasks in cluttered environments," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 183–189.

[18] I. P. Sary, Y. P. Nugraha, M. Megayanti, E. Hidayat, and B. R. Trilaksono, "Design of obstacle avoidance system on hexacopter using vector field histogram-plus," in *2018 IEEE 8th International Conference on System Engineering and Technology (ICSET)*. IEEE, 2018, pp. 18–23.

[19] R. Wang, K. Gao, J. Yu, and K. Bekris, "Lazy rearrangement planning in confined spaces," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 32, 2022, pp. 385–393.

[20] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[21] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[22] J. Mirabel, S. Tonneau, P. Fernbach, A.-K. Seppälä, M. Campana, N. Mansard, and F. Lamiraux, "Hpp: A new software for constrained motion planning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 383–389.

[23] M. Stilman, J.-U. Schamburek, J. Kuffner, and T. Asfour, "Manipulation planning among movable obstacles," in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3327–3332.

[24] Y. Labbé, S. Zagoruyko, I. Kalevatykh, I. Laptev, J. Carpentier, M. Aubry, and J. Sivic, "Monte-carlo tree search for efficient visually guided rearrangement planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3715–3722, 2020.

[25] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee, "Transporter networks: Rearranging the visual world for robotic manipulation," *Conference on Robot Learning (CoRL)*, 2020.

[26] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[27] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[28] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *Ieee access*, vol. 2, pp. 56–77, 2014.

[29] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[31] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.

[32] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.

[33] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.