Task Planning for Object Rearrangement in Multi-Room Environments

Karan Mirakhor*, Sourav Ghosh*, Dipanjan Das, Brojeshwar Bhowmick

Visual Computing and Embodied Intelligence Lab, TCS Research, Kolkata, India
{karan.mirakhor, g.sourav10, dipanjan.da, b.bhowmick}@tcs.com

Abstract

Object rearrangement in a multi-room setup should produce a reasonable plan that reduces the agent's overall travel and the number of steps. Recent state-of-the-art methods fail to produce such plans because they rely on explicit exploration for discovering unseen objects due to partial observability and a heuristic planner to sequence the actions for rearrangement. This paper proposes a novel task planner to efficiently plan a sequence of actions to discover unseen objects and rearrange misplaced objects within an untidy house to achieve a desired tidy state. The proposed method introduces several innovative techniques, including **(i)** a method for discovering unseen objects using commonsense knowledge from large language models, **(ii)** a collision resolution and buffer prediction method based on Cross-Entropy Method to handle blocked goal and swap cases, **(iii)** a directed spatial graph-based state space for scalability, and **(iv)** deep reinforcement learning (RL) for producing an efficient plan to simultaneously discover unseen objects and rearrange the visible misplaced ones to minimize the overall traversal. The paper also presents new metrics and a benchmark dataset called MoPOR to evaluate the effectiveness of the rearrangement planning in a multi-room setting. The experimental results demonstrate that the proposed method effectively addresses the multi-room rearrangement problem.

Introduction

Organizing an untidy household according to user preferences is a challenging task that encompasses multiple aspects, including perception, planning, navigation, and manipulation (Batra et al. 2020). When an agent engages in multi-room object rearrangement, it must rely on sensor data and prior knowledge to devise a comprehensive plan that involves sequencing the object movement to achieve the desired tidy state. The specifications for this goal state can be defined using various modalities such as geometry, images, language, etc. (Batra et al. 2020).

Current research on object rearrangement primarily focuses on single-room setups, emphasizing perception and commonsense reasoning. However, these studies often assume established navigation and manipulation abilities, neglecting the importance of efficient planning in the rearrangement process. Approaches like (Kant et al. 2022; Sarch et al. 2022) use image or language-based commonsense reasoning to detect misplaced objects but struggle with cases involving blocked goal positions or object swapping. Alternatively, methods like (Weihs et al. 2021; Gadre et al. 2022; Trabucco et al. 2022) concentrate on egocentric perception, employing various scene representations to identify misplaced objects and using greedy planners for rearrangement based on user specifications. (Sarch et al. 2022) performs user-specific rearrangement using semantic relations for object identification and exploration, followed by rearrangement using a heuristic planner. Many existing methods explicitly explore the room to locate objects outside the egocentric view, compensating for partial information. However, recent methods (Sarch et al. 2022; Trabucco et al. 2022; Ghosh et al. 2022) face challenges with high traversal costs due to sub-optimal planners, especially in larger multi-room settings (see Fig 1).

Enhancing planning performance is essential for making rearrangement feasible, as it allows us to minimize the time and resources required by the agent to reach the desired goal state. In the context of rearrangement task planning, (Ghosh et al. 2022) proposes a method that assumes complete room visibility from a bird's eye perspective. Their approach aims to overcome planning challenges, including the combinatorial expansion of rearrangement sequencing and swap case resolution, without requiring an explicit buffer. However, their Euclidean distance-based reward does not minimize overall agent traversal during planning, as shown in Fig 1. Their state representation lacks scalability to large numbers of objects. Additionally, their parameter network suffers from two main issues: **(i)** predicting the goal location of non-blocked misplaced objects, which is already known from the goal state in rearrangement, leading to performance degradation. **(ii)** predicting the buffer location for swap cases without considering the object's geometry and the available free space, resulting in poor generalization. Furthermore, depending on accurate ground-truth object positions in both the current and goal states, is not viable in the real-world. In contrast, our focus is on a novel, practical and broader aspect of the object rearrangement problem in the multi-room setting, under partial observability using the egocentric camera view of the agent. Our main emphasis is on efficient task planning, which is necessary for effective rearrangement as illustrated in Fig 1.

Task planning for efficient multi-room object rearrangement under partial observability as depicted in Fig 2, presents

*These authors contributed equally.

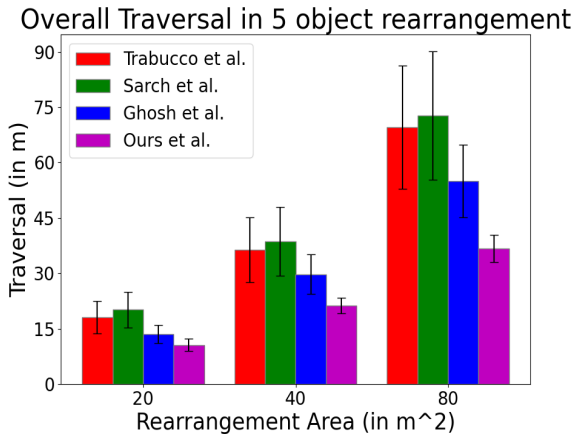

Figure 1: The graph shows the agent traversal for existing methods (Sarch et al. 2022; Trabucco et al. 2022; Ghosh et al. 2022) v/s Ours with increasing rearrangement area highlighting need for efficient planning. The error bars show the standard deviation in the average traversal.

several significant challenges such as **(i)** uncertainty regarding the positions of unseen objects owing to partial observability, **(ii)** scalability to a large number of objects, **(iii)** combinatorial expansion of search space for the sequencing due to simultaneous object discovery (for unseen objects) and rearrangement, **(iv)** minimizing the overall traversal by the agent during simultaneous object discovery and rearrangement. **(v)** resolving blocked goal and swap cases (object collision) without explicit buffer.

In this paper, we present a novel hierarchical method for task planning that aims to overcome the challenges mentioned earlier. Initially, our agent utilizes egocentric perception to explore the house once and capture the semantic and geometric configuration (Batra et al. 2020) of objects and receptacles, thus obtaining the goal state. Following this, the objects within the rooms are shuffled to make an untidy current state. Our hierarchical method then divides the task planning problem into three components - the discovery of unseen objects, collision resolution, and planning - to minimize the agent's overall traversal while simultaneously conducting object search, collision resolution, and rearrangement. **First**, we propose a novel commonsense knowledge-based Unseen Object Discovery Method using large language models (LLMs) (Liu et al. 2019; Kant et al. 2022), that leverages the object-room-receptacle semantics to predict the most probable room-receptacle for an unseen object. **Second**, we propose a novel Cross-Entropy Method (CEM) based collision resolution to produce buffer spaces for swap cases considering the objects' geometry and the size of receptacle-free spaces. **Third**, we use a Deep RL-based planner to produce an action sequence for simultaneous object search and rearrangement. Additionally, we introduce a Directed Graph based representation as the Deep RL state space to effectively represent the geometric positions of objects in the goal and the current state along with the agent's initial position. The proposed representation effectively encodes the scene geometry, facilitating rearrangement planning and enabling scalability of the Deep RL state space to accommodate a large number of objects and scene invariant. By combining all the previously mentioned components in a thoughtful manner, we successfully address the combinatorial optimization problem in rearrangement.

The major contributions of this paper are :

1. To the best of our knowledge, this is the *first end-to-end method* to address the task planning problem for multi-room rearrangement from egocentric view under partial observability, using a user-defined goal state.
2. A novel **Unseen Object Discovery Method** that leverages object-room-receptacle semantics using LLM to predict the most probable room-receptacle for an unseen object.
3. Introducing **Cross-Entropy Method based Collision Resolution** to find buffer spaces for swap cases considering the objects' geometry and the size of free spaces.
4. A new scalable and scene invariant **Directed State Graph** containing the geometric information about the agent and objects in the current and goal state as the Deep RL state.
5. Employing a Deep RL planner to mitigate the combinatorial expansion in the complexity of rearrangement sequencing, and optimize the the overall agent traversal.
6. A new set of **Evaluation criteria** to gauge the efficacy of our method in terms of the agent traversal and the steps taken to complete rearrangement.
7. We present the **MoPOR - Benchmark Dataset** to overcome the shortcomings in existing benchmark (Weihs et al. 2021) for assessing multi-room task planning.

Methodology

Preliminaries

For our setup, we use the apartment scenes from ProcThor (Deitke et al. 2022). To begin our multi-room rearrangement, we perform exploration (Sarch et al. 2022) once in the entire scene to capture the user-specified goal state. Using the RGB-D image and egomotion information at each step from Ai2Thor (Kolve et al. 2017), the agent generates a 2D occupancy map ($\mathcal{M}^{2D}$) for navigation and a 3D map ($\mathcal{M}^{3D}$) to augment the positions of objects and receptacles in 3D to a global reference frame. To acquire semantic labels (**L**) and 2D bounding boxes for both objects and receptacles, we apply a d-DETR (Zhu et al. 2021) detector on every RGB image. The corresponding 3D centroids (**P**) are obtained using depth images, camera intrinsics and extrinsics. Using object segmentation (Rusu and Cousins 2011) on the point cloud of $\mathcal{M}^{3D}$ the agent records the corners of the 3D bounding boxes (**B**) of each object in the goal state. At the end of the goal state exploration, we generate an object list $\mathbf{O} = \{[\mathbf{L}_i, \mathbf{P}_i, \mathbf{B}_i], i = 1, 2, .., N\}$ and a room-receptacle list $\mathbf{R} = \{[\mathbf{L}_i^R, \mathbf{P}_i^R], i = 1, 2, .., N_R\}$. Here, $\mathbf{L}^R$ indicates the receptacle semantic labels containing the room-receptacle name from Ai2Thor (Kolve et al. 2017). Following the capture of the goal state, we make the room untidy by randomly shuffling a selection of objects. Simultaneously, we position the agent at a random location within this untidy room. As the agent's awareness is confined to the visible portion of the untidy room within its egocentric view, hence only a set of

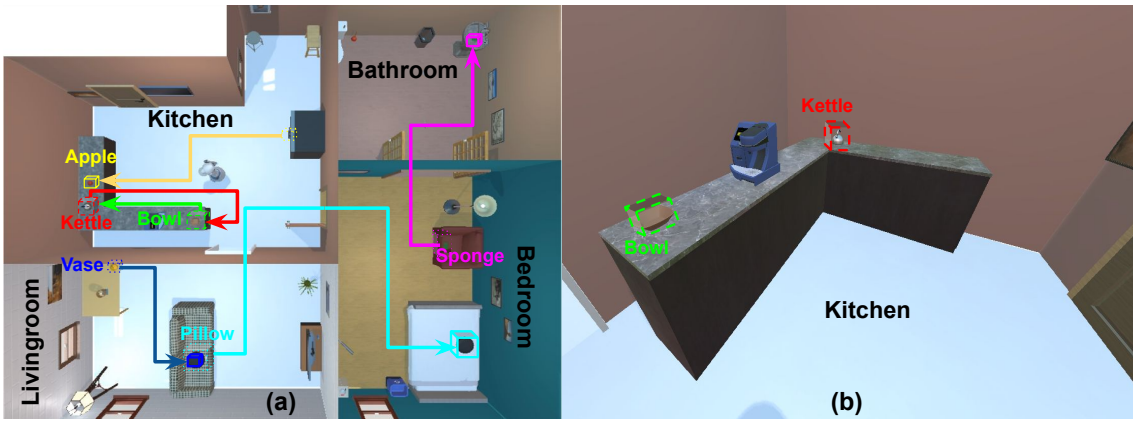

Figure 2: (a) illustrates a bird's eye view of our multi-room rearrangement task and (b) represents the initial ego-view of the agent in the untidy current state. The intended goal state for the objects is depicted by solid 3D bounding boxes, whereas the the initial locations of visible objects in the untidy current state is depicted by the dashed boxes. Meanwhile, the initial positions of unseen objects in the untidy current state is indicated by the dotted 3D bounding boxes. The apple (yellow), an unseen object is inside the kitchen-fridge, while the vase (blue), pillow (pastel cyan) and sponge (magenta) is on the living-table, living-sofa and bedroom-chair respectively. A blocked goal case is illustrated between the vase (blue) and pillow (pastel cyan). Additionally, a swap case is depicted between the bowl (green) and kettle (red).

objects are visible - $\mathbf{O}^V = \{[\mathbf{L}_i^V, \mathbf{P}_i^V], i = 1, 2, .., N_V\}$. Additionally, the agent creates a 2D grid map of free receptacle spaces $\mathcal{M}^{\mathrm{R}}$ in the current state using the instance segmentation mask from Ai2Thor along with the depth image and egomotion to aid in the collision resolution. Comparing $\mathbf{O}$ with $\mathbf{O}^V$ allows for determining only the semantics of unseen objects in the current state $\mathbf{O}^{\overline{V}} = \{[\mathbf{L}_i^{\overline{V}}], i = 1, 2, .., N_{\overline{V}}\}$.

Overview

Given $\mathbf{O}$, $\mathbf{R}$, $\mathbf{O}^V$ and $\mathbf{O}^{\overline{V}}$, the agent must efficiently plan a sequence of actions $\mathcal{A} = \{a_1, ..., \text{done}\}$ to discover $\mathbf{O}^{\overline{V}}$ and simultaneously rearrange the misplaced objects in $\mathbf{O}^V$ to their desired goal position in the current state. The Unseen Object Discovery Method leverages the object-room-receptacle relationship using $\mathbf{O}^{\overline{V}}$ and $\mathbf{R}$ to predict the probable location $\mathbf{P}^{\overline{V}R}$ for $\mathbf{O}^{\overline{V}}$. The Collision Resolution and Buffer Management method, uses the locations and object bounding boxes from $\mathbf{O}$ and $\mathbf{O}^V$ along with $\mathcal{M}^{\mathrm{R}}$ to detect object collision and predict the resolved goal location $\mathbf{P}^B$ for the blocked goal and swap case objects. With $\mathbf{O}$, $\mathbf{O}^V$, $\mathbf{O}^{\overline{V}}$, $\mathbf{P}^{\overline{V}R}$, and $\mathbf{P}^B$, the agent constructs a compact representation of the state space using the Directed State Graph. The Deep RL planner uses the state space to produce the most optimal action $a \in \mathcal{A}$ to either pick-place $\mathbf{O}^V$ or search $\mathbf{O}^{\overline{V}}$ with the objective of minimizing the overall traversal and the number of steps.

Unseen Object Discovery Method

In the context of multi-room rearrangement, the agent must identify unseen objects within the untidy current state, whether in the same room or different rooms, to plan actions effectively. For example, as shown in Fig 2, when the agent is in the kitchen, it cannot see items like the apple in

Algorithm 1: Algorithm for Task planner

Input: Agent's egoview RGB-D & egomotion
Result: Actions $\mathcal{A} = \{a_1, .., a_N, done\}$

Create $O, R, \mathcal{M}^{\mathrm{2D}}, \mathcal{M}^{\mathrm{3D}}$ from Goal State;
Create $O^V, O^{\overline{V}}, \mathcal{M}^{\mathrm{R}}$ from Current State;
while *a is not done* **do**
  $P^{\overline{V}R} \leftarrow$ *UODM* $(O^{\overline{V}}, R)$;
  if *Collision between* O_i^V *and* O_j^V **then**
    if *Swap case* **then**
      $P_i^B \leftarrow$ *buffer for* O_i^V *near* P_j^V;
      $P_j^B \leftarrow$ *buffer for* O_j^V *near* P_i^V;
    else if O_i^V *blocks goal of* O_j^V **then**
      $P_j^B \leftarrow P_j^V$;
    else if O_j^V *blocks goal of* O_i^V **then**
      $P_i^B \leftarrow P_i^V$;
  $s \leftarrow$ *DSG* $(O, P^B, O^V, O^{\overline{V}} \cup P^{\overline{V}R})$;
  $a = \arg\max_{a \in \mathcal{A}} Q_\theta(s, a)$;
  if $a == O_i^V$ **then**
    Pick-Place object O_i^V;
  else if $a == O_i^{\overline{V}}$ **then**
    if $O_i^{\overline{V}}$ *is found during traversal* **then**
      go to 24;
    else if $O_i^{\overline{V}}$ *is found in* $P_i^{\overline{V}R}$ **then**
      Pick-Place object $O_i^{\overline{V}}$;
    else
      Remove predicted receptacle from R;
  Remove receptacle/s without any $O^{\overline{V}}$ *from R*;
  Update $(O^V, O^{\overline{V}}, R, \mathcal{M}^{\mathrm{R}})$;

the fridge-kitchen, or the vase (blue), pillow (pastel cyan), and sponge (magenta) in the living room and bedroom. To address this, we propose the Unseen Object Discovery Method (UODM), leveraging Large Language Models' (LLMs) commonsense knowledge to predict probable room-receptacles for unseen objects. However, LLMs may not always provide human-commonsense compliant predictions for untidy scenes, as detailed in Appendix[1]. Therefore, we leverage the semantic relationship between $\mathbf{O}^{\overline{V}}$ and $\mathbf{R}$ by finetuning their output embeddings from LLM using the AMT dataset (Kant et al. 2022). Unlike previous methods such as (Sarch et al. 2022), which focus solely on object-receptacle relationships, our method takes into account the object and room-receptacle semantic relationship to handle multi-room setups. We generate the RoBERTa embeddings $\mathbf{E}^{\overline{V}R}$ for pairwise concatenated labels of unseen objects $\{\mathbf{L}_i^{\overline{V}}\}_{i=1,2,..,N_{\overline{V}}}$ and room-receptacles $\{\mathbf{L}_i^R\}_{i=1,2,..,N_R}$. As each object is paired with every receptacle, the total number of embeddings for all the object-room-receptacle (ORR) pairs is $N_E = N_{\overline{V}} \times N_R$. We devise a two-step approach to enhance the accuracy and reduce the search space for finding the exact ORR from the N_E pairs. **First**, we use an MLP-based Filter Network (FN) to predict the probability for ORR from N_E. We train this network using a Cross-Entropy Loss on the ground truth class labels for each ORR in the dataset. Here, the class labels $\{i = 1 : \textit{Probable Class}, 2 : \textit{Implausible Class}\}$ indicate the probability of finding a misplaced object at a given room-receptacle. In the **second** step, we use a regression-based Ranking Network (RN) to estimate the probability scores for embeddings of probable class, with N_{SR} representing the total number of such embeddings. We use a fully connected MLP, and train this network using MSE Loss, with respect to the ground truth probability scores from human annotations. The room-receptacles with the highest scores from RN are selected as the probable room-receptacles for the unseen objects $O^{\overline{V}}$. Finally, the predicted room-receptacle's position $\{\mathbf{P}_i^{\overline{V}R}\}_{i=1,..N_{\overline{V}}}$ from the goal state is used as the location for $\mathbf{O}^{\overline{V}}$ in the current state, as receptacles are static in the scene.

To prevent fruitless searches, we implement simple strategies as shown in Line 19:25. Line 24, illustrates a scenario where the agent encounters a receptacle on its path that does not contain any $O^{\overline{V}}$. As a result, the agent excludes it from future search attempts.

Collision Resolution and Buffer Management

To ensure the robustness of our task planning in collision scenarios, we need to identify and resolve the blocked goal and swap cases by leveraging the geometry of objects (B) and the size of free spaces in receptacle occupancy map ($\mathcal{M}^{\mathrm{R}}$). We identify the collision cases between objects in $\mathbf{O}^V$, using $B_i, B_j \in B$ for each pair of objects and $\mathcal{M}^{\mathrm{R}}$. We project B_i, B_j onto $\mathcal{M}^{\mathrm{R}}$ and find the maximum size rectangle B_i^M and B_j^M that encloses their projections respectively. Additionally, we require the projected goal positions p_i^M, p_j^M of the two objects in $\mathcal{M}^{\mathrm{R}}$. To identify the type of collision case, we check **(i)** $B_i^M \cap B_j^M(p_j^M)(B_j^M$ at $p_j^M) \neq \phi$ and **(ii)** $B_i^M(p_i^M) \cap B_j^M \neq \phi$. Three cases arise based on these conditions: **(i)** If none of them are true, there is no collision, **(ii)** If only one of the conditions is true, it is a blocked goal case, and **(iii)** If both conditions are true, it is a swap case.

To resolve swap cases, we find a buffer space in a time bound and effective manner, using a stochastic optimization over $p_k \in \mathcal{M}^{\mathrm{R}}$ that maximizes the objective function $\mathcal{F}(p_k)$ in Eq 1[2]. To perform this optimization, we use the Cross-Entropy Method (CEM) - a simple derivative-free optimization algorithm, which is parallelizable and moderately robust to local optima (Rubinstein and Kroese 2004). CEM samples a batch of N_f points from the free spaces in $\mathcal{M}^{\mathrm{R}}$ at each iteration. Then, it fits a Gaussian distribution to the best $M_f \leq N_f$ samples based on $\mathcal{F}(p_k)$. Further samples for the next batch of N_f are taken from this Gaussian.

$$\mathcal{F}(p_k) = \begin{cases} e^{-\mathcal{E}(p_k, p_j^M)}, & \text{if } B_i^M(p_k) \cap B_j^M(p_j^M) = \phi \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Here, $\mathcal{E}(p_k, p_j^M) = \|(p_k - p_j^M)\|_2$. The buffer locations obtained from CEM are used to update P^B for swap case objects as shown in Line 7:8. Once a swap object is placed in its buffer space, it is considered a blocked goal case. To resolve a blocked goal case, the goal blocked object is made static temporarily, unless the goal blocking object vacates its goal position. This is done by updating P_B, as in Line 10:12.

Rearrangement Planner

Rearrangement planning is a long-horizon problem that suffers from the combinatorial expansion with increasing number of objects. To this end, we employ data-driven Q-learning (Watkins and Dayan 1992) that leverages the Bellmann principle (Bellman 1957) for achieving optimality in long horizon problems. Moreover, to aid the planner we use the Directed spatial graph as the state space which allows for scene invariance and scalability to a large number of objects.

Directed State Graph We present a directed spatial graph ($\mathcal{G} = \{V, E\}$) to create a concise state space representation to aid the planner in efficient rearrangement sequencing. The edges in this directed graph represent all the feasible paths for rearrangement completion, disregarding the redundant information present in an undirected graph, thereby improving the training efficiency (refer Appendix[2]). The nodes V (shown in Fig 3) contain **(i)** the agent node with 3D position of the agent, **(ii)** the source nodes with current object positions and labels from $\{O^V, \{O^{\overline{V}} \cup P^{\overline{V}R}\}\}$ and **(iii)** the goal nodes with the goal object positions and labels from O. Unlike (Ghosh et al. 2022), we include the agent node in the Deep RL state space, as it enables the planner to effectively select the initial action based on the agent's position with respect to the objects'. The directed edges of the graph connect: **(i)** the agent node to every source node, **(ii)** the source node of

[1] http://tinyurl.com/14043Appendix

[2] shows an objective function to find a buffer for B_i^M near $B_j^M(p_j^M)$

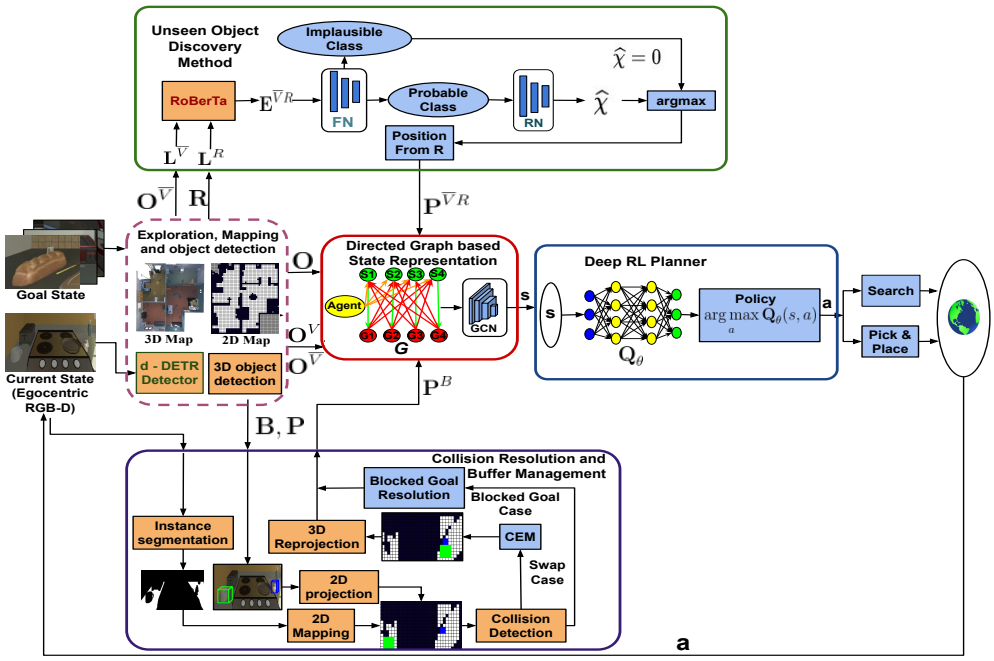

Figure 3: Overall hierarchical pipeline of our proposed method.

each object to its respective goal node, and **(iii)** the goal node of each object to the current node of every other object. The edge attributes $E_c = \{\mathcal{D}(p_i^G, p_j^G)_{i \neq j}\}$ include the length of the shortest collision free path $\mathcal{D}(p_i^G, p_j^G)_{i \neq j}$ from the node position p_i^G to connected node position p_j^G. $\mathcal{D}(p_i^G, p_j^G)_{i \neq j}$ is computed using BFS algorithm between the 2D projections of p_i^G, p_j^G on $\mathcal{M}^{2D}$. For unseen objects in the current state, the source object nodes in G are augmented with $P^{\overline{V}R}$ from UODM. Similarly, for blocked goal and swap cases, the goal nodes of such objects are updated with P^B from Collision Resolution and Buffer Method. Therefore, this directed graph representation aids the Deep RL to concisely comprehend the geometric details of the current and goal state. We use a *Graph Convolution Network (GCN)* to generate meaningful graph embedding from $\mathcal{G}$, that enables the Deep RL state space to remain scalable and scene invariant.

Deep RL Planner The task planner should efficiently plan a sequence of actions to simultaneously **(i)** rearrange visible objects and **(ii)** search for unseen objects at probable receptacles. This reduces the agent's overall travel time as **(i)** it eliminates the requirement of explicit exploration to find unseen objects and **(ii)** the rearrangement of visible objects inevitably leads to the discovery of some unseen objects. To accomplish these objectives, we utilize a Conservative Q-Learning based on reinforcement learning, which is similar to the approach presented in (Kumar et al. 2020). The state space for Deep RL is defined as s and action $a \in \mathcal{A}$ denotes the selected object in $\mathbf{O}^V$ or $\mathbf{O}^{\overline{V}}$. Our approach follows the principles of a *Markov Decision Process (MDP)*, where a reward $r(s, a)$ is received at each time step t for selecting a from the policy $\pi(a|s) = \arg\max_{a \in \mathcal{A}} Q_\theta(s, a)$, that moves the agent from the current state s to the next state $\bar{s}$. So according to Bellman equation Eq 2, our objective is to minimize the temporal difference error to get the desired action.

$$L_{TD} = \underset{(s,a,\bar{s}) \leftarrow R_B}{0.5\,\mathbb{E}} [(r(s,a) + \gamma \max_{\bar{a} \in \mathcal{A}} Q_{\bar{\theta}}(\bar{s}, \bar{a}) - Q_\theta(s,a))^2] \tag{2}$$

Here, θ and $\bar{\theta}$ are the parameters of the Q-Network and target Q-Network respectively and R_B is the replay buffer. The Q-Network tends to overestimate the policy value, which is undesirable and affects the sampling efficiency (refer Appendix). To prevent this, we employ the Conservative Q-Learning technique similar to that used in (Kumar et al. 2020). This ensures that the expected value of a policy under the learned Q-function provides a lower-bound estimate of its true value. Therefore, the combined loss obtained on adding the conservative lower bound loss to Eq 2 is shown in Eq 3

$$L_{CQL} = \underset{\substack{s \leftarrow R_B \\ a \sim \pi(a|s)}}{\alpha\,(\mathbb{E}} [Q_\theta(s,a)] - \underset{s,a \leftarrow R_B}{\mathbb{E}} [Q_\theta(s,a)]) + L_{TD} \tag{3}$$

Here $\alpha \geq 0$ is tradeoff factor between the conservative loss term and the TD error. When it comes to Long Horizon planning, using the sparse reward is not an efficient method for training Deep RL, as noted in (Gehring et al. 2021). We employ a three-component hierarchical dense reward structure: (i) *Rearrange-able Object Reward :* where selecting a misplaced object yields a negative reward equivalent to the action path length needed for pick-and-place; (ii) *Static Object Reward :* penalizing non-rearrangeable object movement to avoid redundancy; (iii) *Completion Reward :* offering a

high positive reward for correctly rearranging only the misplaced objects; and (iv) *Collision Resolution Reward :* prioritizing the rearrangement of goal-occupying objects to resolve blockages. Additional information on the reward structure and choice of rewards is provided in the Appendix[2].

We employ an off-policy technique to train our Conservative Q-Learning method using a diverse set of rearrangement configurations, which is similar to the approach proposed by (Kalashnikov et al. 2018). To balance exploration and exploitation, we use the ϵ greedy method, as described in (Kalashnikov et al. 2018). To ensure stable Off-policy training, we update $\bar{\theta}$ weights using polyak averaging on θ, which is similar to (Bester, James, and Konidaris 2019).

Experiment

In this section, we describe the dataset[3], metrics, and detailed results of our proposed method[4] and its modules, in addressing the multi-room rearrangement problem.

Unseen Object Discovery Dataset

The AMT dataset (Kant et al. 2022) consists of 268 object categories present in 12 distinct rooms and 32 receptacle types. Each object-room-receptacle (ORR) pair is evaluated by 10 annotators who rank them into one of three classes: correct (positively ranked), misplaced (negatively ranked), or implausible (not ranked). By calculating the mean inverse of the ranks assigned to each ORR, we obtain the ground-truth scores. For our specific problem, our preference order for ORRs is as follows: misplaced class, correct class, and lastly, the implausible class. Hence, we re-label the classes and their scores as (i) misplaced and correct class $\rightarrow$ probable class, while (ii) implausible class remains the same.

Benchmark Dataset for Testing - MoPOR

The existing benchmark dataset, RoomR (Weihs et al. 2021), is utilized to evaluate rearrangement policies across different room scenarios. However, it has certain limitations. It restricts the number of objects to a maximum of 5 and does not allow object placement within another receptacle, nor does it include blocked goal or swap cases. Consequently, it cannot adequately assess planning aspects such as the number of steps, agent traversal, blocked goals, or swap cases. To overcome these limitations, we introduce MoPOR, a new benchmark dataset designed specifically for testing task planners in Ai2Thor. MoPOR encompasses a diverse collection of single-room scenes from iThor and multi-room scenes from ProcThor (Deitke et al. 2022). It supports up to 108 object and receptacle categories. This dataset enables a wide range of rearrangement scenarios involving up to 40 objects. Furthermore, MoPOR includes random partial observability cases, object placement within receptacles in the current state, as well as blocked goal and swap cases. Moreover, it's worth noting that object placement configurations in MoPOR impact the effectiveness of sub-optimal planning policies in terms of agent traversal. Additional information on the MoPOR dataset can be found in the Appendix[2].

[3]Dataset link : https://tinyurl.com/MoPORDataset

[4]Code link : https://tinyurl.com/MultiRoomCode

Training

The training details of UODM and DSG with Deep RL planner are available in the Appendix[2].

Evaluation Criteria

Metrics detailed in (Weihs et al. 2021) solely evaluate the rearrangement methods based on the successful completion of the task. To ensure a fair evaluation of our method compared to existing methods and ablations, we introduce a new evaluation criteria to emphasize the effectiveness of rearrangement planning in minimizing the number of steps taken and the overall traversal of the agent.

- **SRN** : **S**uccess **R**atio measures rearrangement episode success and efficiency by combining the binary success rate (S) and the **N**umber of steps (N_S) taken by the agent to rearrange a set number of objects (N). A higher SRN indicates a more efficient and successful rearrangement episode, as it implies a lower N_S for a given N. (**SRN** = $S \times N/N_S$)
- **EOD**: **E**fficiency in unseen **O**bject **D**iscovery is the ratio of the number of initially unseen objects ($N_{\overline{V}}$) to the number of search attempts ($N_{S\overline{V}}$). A lower **EOD** indicates a higher $N_{S\overline{V}}$ for a given $N_{\overline{V}}$, signifying a less efficient in discovering unseen objects.(**EOD** = $N_{\overline{V}}/N_D$)
- **TTL**: The metric **T**otal **T**raversal **L**ength quantifies the cumulative distance covered by the agent while successfully executing a rearrangement episode. In an identical test configuration, a lower **TTL** indicates a more efficient rearrangement sequencing.

Baselines

We ablate our method against ground-truth perception, various methods for object search and different planners. We employ **(i)** ***Ours-GT*** to examine the impact of inaccurate perception on our method using the accurate ground-truth perception from Ai2Thor (Kolve et al. 2017). To understand the importance of UODM in our method, we replace it by a random search policy in **(ii)** ***Ours-RS***, which predicts probable receptacles for unseen objects with uniform probability and a greedy exploration strategy (Chaplot et al. 2020) in **(iii)** ***Ours-GE*** that optimizes for map coverage to discover all the unseen objects. To gauge the efficacy of our planner we replace the Deep RL planner in our method with a heuristic planner in **(iv)** ***Ours-HP*** that greedily selects an action with the shortest agent traversal to complete the object pick-place.

Results

Ablations The state-of-the-art methods dealing with user-defined goal state do not demonstrate their results in multi-room rearrangement task. Due to this limitation, we gauge the performance of our method in Tab 1, by comparing it against the set of baselines in a multi-room setting on MoPOR - Benchmark Datset. Tab 1 indicates that our method is scalable to a large number of objects, with consistently increasing **SRN** values for swap cases and both scenarios of partial observability (**P.O.** : objects outside the field of view, **F.O.** : objects inside closed receptacles). In addition, the consistently high **SRN** for a growing number of swap cases indicates that *Ours* and *Ours-GT* can effectively handle

#O	#V	#U		#S	**Ours-GT**			**Ours**			Ours-RS			Ours-GE			Ours-HP		
		P.O	**F.O**		**SRN**	**EOD**	**TTL**	**SRN**	**EOD**	**TTL**	**SRN**	**EOD**	**TTL**	**SRN**	**EOD**	**TTL**	**SRN**	**EOD**	**TTL**
10	6	4	0	2	**0.59**	**0.55**	**37.03**	**0.41**	**0.53**	**37.82**	0.19	0.20	60.72	0.15	0.13	70.09	**0.41**	**0.53**	45.82
	6	0	4	2	**0.55**	**0.51**	**38.25**	**0.39**	**0.48**	**40.27**	0.13	0.15	65.47	0	NC	NC	**0.39**	**0.48**	50.27
20	12	8	0	4	**0.61**	**0.60**	**66.94**	**0.43**	**0.56**	**69.29**	0.22	0.24	89.45	0.19	0.17	101.68	**0.43**	**0.56**	82.42
	12	0	8	4	**0.58**	**0.55**	**68.95**	**0.40**	**0.53**	**72.38**	0.17	0.18	98.37	0	NC	NC	**0.40**	**0.53**	89.67
30	18	12	0	6	**0.64**	**0.67**	**90.25**	**0.45**	**0.62**	**94.76**	0.27	0.28	120.45	0.24	0.19	131.92	**0.45**	**0.62**	109.42
	18	0	12	6	**0.61**	**0.61**	**95.71**	**0.43**	**0.58**	**98.58**	0.23	0.22	132.16	0	NC	NC	**0.43**	**0.58**	118.72

Table 1: Results for Multi-Room Rearrangement with comparison against Baselines. Here, #O indicates number of objects, #V indicates number of visible objects, #U indicates number of Unseen objects, #S indicates number of Swap Cases, P.O. indicates partially occluded cases i.e. objects which are outside the field of view presently and F.O. stands for fully occluded cases i.e. objects placed inside closed receptacles. *Ours-GE* does not handle F.O. cases, therefore its EOD is non-computable (NC) due to division by zero. The table shows that *Ours-GT* and *Ours* outperform the other baselines in terms of the evaluation criteria.

swap cases, using the Cross-entropy based method for buffer prediction. During the time of rearrangement of visible objects and search for unseen objects, agent consequentially finds other unseen objects. This finding of unseen objects aids the gradual increment in **SNR** and **EOD** with the increasing number of objects

As anticipated, *Ours-GT* results in significantly better **SNR**, **EOD** and **TTL**, compared to Ours and all the baselines, because it uses ground-truth object detection and labelling.

The variation in the results between Ours and the baselines primarily arises from their approaches to handle partial observability cases, since Ours and the baselines employ the same buffer prediction method for swap cases. In both the cases of partial observability (**P.O.** & **F.O.**), Ours performs significantly better than Ours-GE, Ours-RS and Ours-HP in terms **SRN**, **EOD** and **TTL**. This is due to the efficacy of the Unseen Object Discovery Method (UODM) and the efficient planning of Deep RL during simultaneous object search and rearrangement. Ours-GE incurs a high traversal cost in terms of **TTL** because it explicitly explores the entire apartment to find the **PO** objects. Moreover, Ours-GE fails to address the **FO** cases (**SRN = 0**) because the greedy exploration policy (Chaplot et al. 2020) based on map coverage does not consider opening and closing receptacles to find **FO**. Whereas, Ours-RS randomly visits receptacles to discover **PO** or **FO** cases, which again increases **TTL**. In contrast, our approach performs similarly in both cases of partial observability because UODM comprehends a semantic relationship between an object and any type of room-receptacle - rigid or articulated. To assess the exploration strategy's effectiveness in object search based on ENR, we define each newly generated location or sequence of navigational steps from the exploration policy as a single search attempt. A higher quantity of search attempts in Ours-GE and Ours-RS results lower ENR and SNS. In case of **PO**, the values of **EOD** and **TTL** for Ours-RS is slightly higher than Ours-GE because Ours-RS simultaneously searches objects and performs rearrangement, but Ours-GE performs additional exploration to find unseen objects. Ours-HP[5] only shows a higher **TTL** compared to Ours for both **PO** and **FO** cases in partial observability due to the greedy planning method.

Please refer to the Appendix[2] to find results for the analysis of the choice of hyper-parameters for each of our modules.

Comparison Against State-of-the-Art Methods for Room-Rearrangement The existing methods train and show results on a single-room rearrangement task. Therefore, for a fair comparison, we compare our method against them in a single-room setting on MoPOR - Benchmark Dataset and RoomR (Weihs et al. 2021). Please refer to the Appendix[2] to find the detailed comparison tables and discussion.

Qualitative Results

To show the qualitative results of our method in multi-room rearrangement, we have created multiple test scenario videos to highlight the robustness of our method. We also evaluate our method in a new environment- Habitat, as highlighted in our supplementary video[6]. This transfer does not require any additional training for our UODM or Deep RL planner. This demonstrates how our method excels in seamless sim-to-sim transfer, reinforcing its suitability for deployment in real-world scenarios. Please refer the supplementary video[6].

Limitations

Our method has limitations in finding unseen objects that are concealed by other static objects - those that will not be moved during rearrangement because their current and goal locations are identical. Additionally, our approach presumes the availability of flawless motion planning and manipulation.

Conclusion

This paper introduces a novel task planner designed for tidying up an apartment while dealing with partial observability, which can be adapted to various situations and generates a sequence of actions that minimizes the agent's overall traversal and the number of steps taken during simultaneous unseen object discovery and rearrangement. In future work, we plan to explore the deployment of our method in real-world.

[5]Ours-HP has similar UODM and Collision Resolution Method

[6]http://tinyurl.com/SuppVideo

References

Batra, D.; Chang, A. X.; Chernova, S.; Davison, A. J.; Deng, J.; Koltun, V.; Levine, S.; Malik, J.; Mordatch, I.; Mottaghi, R.; Savva, M.; and Su, H. 2020. Rearrangement: A Challenge for Embodied AI. *ArXiv*, abs/2011.01975.

Bellman, R. 1957. A Markovian Decision Process. *Journal of Mathematics and Mechanics*, 6(5): 679–684.

Bester, C. J.; James, S. D.; and Konidaris, G. D. 2019. Multi-pass q-networks for deep reinforcement learning with parameterised action spaces. *arXiv preprint arXiv:1905.04388*.

Chaplot, D.; Gandhi, D.; Gupta, S.; Gupta, A.; and Salakhutdinov, R. 2020. Learning to Explore using Active Neural SLAM. In *Eighth International Conference on Learning Representations, ICLR 2022*.

Deitke, M.; VanderBilt, E.; Herrasti, A.; Weihs, L.; Salvador, J.; Ehsani, K.; Han, W.; Kolve, E.; Farhadi, A.; Kembhavi, A.; and Mottaghi, R. 2022. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *NeurIPS*. Outstanding Paper Award.

Gadre, S. Y.; Ehsani, K.; Song, S.; and Mottaghi, R. 2022. Continuous Scene Representations for Embodied AI. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14829–14839.

Gehring, C.; Asai, M.; Chitnis, R.; Silver, T.; Kaelbling, L. P.; Sohrabi, S.; and Katz, M. 2021. Reinforcement Learning for Classical Planning: Viewing Heuristics as Dense Reward Generators. In *International Conference on Automated Planning and Scheduling*.

Ghosh, S.; Das, D.; Chakraborty, A.; Agarwal, M.; and Bhowmick, B. 2022. Planning Large-scale Object Rearrangement Using Deep Reinforcement Learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; and Levine, S. 2018. Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. In Billard, A.; Dragan, A.; Peters, J.; and Morimoto, J., eds., *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, 651–673. PMLR.

Kant, Y.; Ramachandran, A.; Yenamandra, S.; Gilitschenski, I.; Batra, D.; Szot, A.; and Agrawal, H. 2022. Housekeep: Tidying Virtual Households using Commonsense Reasoning. In *European Conference on Computer Vision*.

Kolve, E.; Mottaghi, R.; Han, W.; VanderBilt, E.; Weihs, L.; Herrasti, A.; Gordon, D.; Zhu, Y.; Gupta, A.; and Farhadi, A. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1179–1191. Curran Associates, Inc.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Rubinstein, R. Y.; and Kroese, D. P. 2004. *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-Carlo Simulation (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN 038721240X.

Rusu, R. B.; and Cousins, S. 2011. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China: IEEE.

Sarch, G.; Fang, Z.; Harley, A. W.; Schydlo, P.; Tarr, M. J.; Gupta, S.; and Fragkiadaki, K. 2022. TIDEE: Tidying Up Novel Rooms using Visuo-Semantic Commonsense Priors. In *European Conference on Computer Vision*.

Trabucco, B.; Sigurdsson, G. A.; Piramuthu, R.; Sukhatme, G. S.; and Salakhutdinov, R. 2022. A Simple Approach for Visual Rearrangement: 3D Mapping and Semantic Search. In *Workshop on Learning from Diverse, Offline Data*.

Watkins, C. J.; and Dayan, P. 1992. Q-learning. *Machine learning*, 8: 279–292.

Weihs, L.; Deitke, M.; Kembhavi, A.; and Mottaghi, R. 2021. Visual Room Rearrangement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5922–5931.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.