Trial-Oriented Visual Rearrangement

Yuyi Liu[1,2], Xinhang Song[1,2*], Tianliang Qi[1,2], Shuqiang Jiang[1,2]
[1]Institute of Computing Technology, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, China
{yuyi.liu, xinhang.song, tianliang.qi}@vipl.ict.ac.cn
sqjiang@ict.ac.cn

Abstract

Towards visual room rearrangement for embodied agents, this paper tackles the intricate challenge of restoring a disarrayed scene configuration to its intended goal state. The task necessitates a range of sophisticated capabilities, including efficient spatial navigation, precise and accurate object interaction, sensitive scene change detection, and meticulous restoration techniques. The inherent complexity of this endeavor stems from the diverse nature of potential object changes, encompassing movements within the space, alterations in appearance, and changes in existence—where objects may be introduced or removed from the scene. Previous methods, either end-to-end reinforcement learning or modular approaches, struggle with handling these changes in a unified manner due to the heterogeneous nature of the inference spaces. To address this, this paper proposes a Trial-Oriented Visual Rearrangement (TOR) framework, which leverages the principles of stronger embodiment to prune the joint reasoning space and identify a smaller shared space for processing various object changes. TOR maintains a differential point cloud representation to capture environmental changes and uses two core mechanisms, assessment and trial, to iteratively restore the scene to the goal state. Experimental results demonstrate the effectiveness of TOR in restoring both object movement and appearance changes and show its generalization to complex multi-room environments.

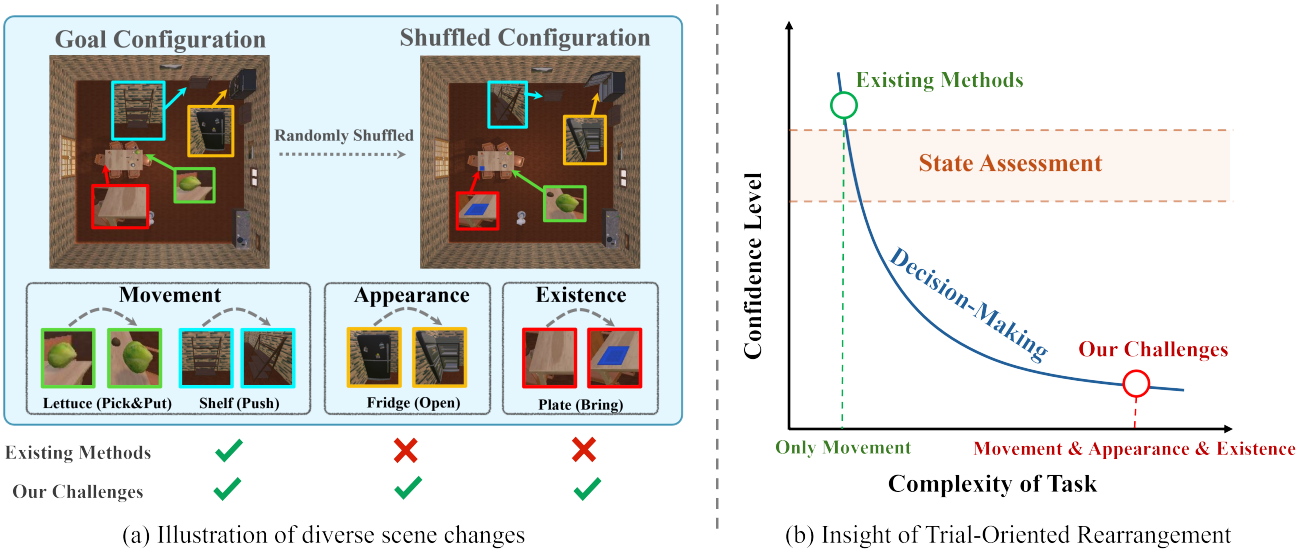
1. Introduction

Visual room rearrangement remains a core challenge for embodied agents, whose goal is to recover a shuffled scene configuration to the goal state. This task demands a wide range of capabilities from the agent, including foundational skills such as efficient navigation and accurate interaction with objects. However, the principal challenge lies in scene change detection and scene change restoration. In real-world scenarios, changes in objects exhibit considerable diversity, primarily categorized into three types, movement, appearance and existence changes, which together constitute the inherent complexity of visual rearrangement, as shown in Fig.1(a)

Recent works can be divided into end-to-end reinforcement learning methods and modular methods. Reinforcement learning methods [11, 35] try to leverage a lot of experience in training and memorize the environment states in parametric mapping mechanism. However, due to the limitations of model parameters and training data, the end-to-end RL methods struggle to handle such difficult task with excessive complexity of state space. Alternatively, modular methods seek to partition the task into perception and planning modules, which construct and compare the explicit scene graphs of two environment states to infer rearrangement goals. While achieving reasonable performance on addressing movement changes, the modular methods encounter challenges to handle appearance and existence changes in a unified manner. The underlying reason lies in the highly heterogeneous nature of the inference spaces for movement, appearance, and existence changes. Integrating these factors within a unified reasoning framework can result in explosive inference complexity, thereby reducing the confidence in decision-making. If decoupled, each change form would require a individual module, reducing overall re-usability.

To address this issue, our motivation is to prune the joint reasoning space and identify a smaller shared space for unified processing various object changes. Traditional pruning methods typically impose prior distributions on the reasoning space. Specifically, for embodied agents performing tasks in real-world physical environments, their prior knowledge inherently includes various physical laws. Although a fully accurate world model adhering to all physical laws has yet to emerge, every action taken by an embodied agent inherently complies with these laws. Therefore, we can leverage the continuous interactions with environment of the embodied agent to directly prune search spaces that violate physical principles. Besides, as illustrated in

*Corresponding authors



(a) Illustration of diverse scene changes

(b) Insight of Trial-Oriented Rearrangement

Figure 1. (a) The types of object changes are highly varied and can be primarily classified into three categories: movement, appearance, and existence changes. Together, these changes form the fundamental complexity of visual rearrangement tasks. However, existing methods only focus on movement changes and fail to handle appearance and existence changes in a unified manner. (b) As the task setting becomes more complex, the confidence in decision-making tends to decrease, whereas the confidence in perceiving and assessing environmental states remains relatively stable. Thus, we can achieve a unified framework for various scene changes by iteratively employing trial and assessing the effectiveness of each action based on whether the current scene more closely approximates the goal state.

Fig.1(b), while decision confidence may decrease with increasing task complexity, the confidence in assessing environmental states remains relatively stable. Based on this observation, for each action performed by agent, we can evaluate whether the current scene is closer to the target state. This evaluation serves as a measurement of the action's effectiveness, enabling the agent to refine its decision-making process and complete the task more effectively.

We propose Trial-Oriented Visual Rearrangement (TOR), to the best of our knowledge, a more embodied pioneering attempt to tackle visual rearrangement task, compared to previous works with pre-defined polices. Our method views the agent's intelligent behavior as an inseparable integration of its physical presence, perceptual abilities, and dynamic interactions with environment. Specifically, we maintain a differential point cloud representation to capture scene changes, recording information about newly protruding and missing parts in shuffled scene configuration. To restore these changes throughout the scene, we design two core modules, interaction assessment module and trial module, which operate alternately. The assessment module analyzes the trends in differential point cloud to quantify the extent to which the behavior reduces the discrepancy between the current and goal scene configuration. Meanwhile, the trial module, informed by current observations and feedback from assessment module, recurrently attempts the next optimal operation from trial space until a successful restoration is achieved. Through this iterative cycle of trail-assessment-refinement, the agent restores all objects to their goal states in succession.

We evaluate our TOR model on AI2THOR rearrangement benchmark based on RoomR dataset. The experimental results demonstrate the effectiveness of our method in restoring both object movement and appearance changes. Given that RoomR dataset includes only single-room scenes and lacks existence changes, we build a more practical and challenging dataset called Versatile Indoor Rearrangement (VIR) based on multi-room environments, covering diverse object changes encountered in real-world scenarios. The results on VIR exhibit the exceptional superiority of our method even in such complex environments.

2. Related Works

2.1. Embodied AI

Embodied AI refers to artificial intelligence that is integrated into a physical form and interacts with its environment in a way that mimics human-like perception, reasoning, and action. The key characteristics of embodied AI include real-time environmental interaction and feedback, which can be simulated on software platforms supporting such scenarios [12, 16, 21, 30]. Various tasks for embodied agents have facilitated the development of methodologies and techniques that enhance the perceiving and reasoning abilities of agents, such as object navigation [23, 33, 34, 38–41], scene exploration [9, 25, 26, 28], embodied question

answering [5, 6, 36] and object manipulation [10, 37]. However, most of these works pertain to the domain of weak embodiment, where the agents perform tasks with pre-defined policies without considering the feedback of interaction. In contrast, our method leverages information accumulated from unsuccessful attempts to guide agent in refining its interaction strategy to complete visual rearrangement task.

2.2. Rearrangement

Rearrangement planning [2–4, 8, 19, 20, 22] has long been a research hot spot in the field of task and motion planning [13–15, 17, 18], focusing on the capability of interacting with and manipulating objects to achieve specific environment layouts. Such rearrangement planning tasks generally do not tackle perception issues directly, instead, they assume that the state of the objects can be fully accessed. In recent years, visual room rearrangement [1, 35] has emerged as an important area within the field of embodied AI, where agents perceive the environments solely through visual input. The spatial scope of rearrangement is also not limited to the desktop but extends across multiple rooms, presenting greater challenges for intelligent agents in understanding and reasoning about the environment.

2.3. AI2THOR Rearrangement Challenge

We focus on the AI2THOR Rearrangement Challenge [35], which is built on an open source interactive environment simulator, AI2THOR [21]. The rearrangement task can be divided into two variants: one-phase and two-phase. In the one-phase task, the agent has the same perspective on both the current environment and the target environment simultaneously. In the two-phase task, which is our main focus, the agent should first explore the target environment autonomously to memorize it, then take actions to rearrange the shuffled environment. Prior works focus on semantic-level scene understanding and memorizing, including methods of 3D sematic mapping [31] and object relationship graph [11, 29, 32]. CAVR [24] proposed a category agnostic model for visual rearrangement task, which utilizes differential point cloud detection to represent scene change. Distinguishing from previous work that is only capable of object movement, our work propose a trial-oriented framework to handle various scene changes uniformly, including movement, appearance and existence changes.

3. Visual Rearrangement

3.1. Task Definition

According to the rearrangement setting formally defined by Batra et al[1], this task is a special case of Partially Observable Markov Decision Processes(POMDP) and requires an agent to transform an environment from an initial state s_0 to a goal state $s^* \in S^*$ via a sequence of actions $a \in A$. The set of goal states S^* and initial state s_0 both belong to the world state space S, which is factorized as the Cartesian product of all rigid-body pose spaces: $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \ldots \mathcal{S}_n$, where $\mathcal{S}_i = SE(3) = \mathbb{R}^3 \times SO(3)$ denotes the space of i_{th} rigid-body space, with $\mathbb{R}^3$ and $SO(3)$ representing 3D locations and rotations space, respectively. But due to the noisy and incomplete onboard perception in embodied AI, the agent typically have no access to world states and operate solely from sensory observations $o \in O$ and goal specification $g = \phi(s_0, S^*)$, where $\phi : \mathcal{S} \times 2^{\mathcal{S}} \mapsto \mathcal{G}$ denotes a goal specification function. The goal specification g encompasses a variety of forms, including GeometricGoal, ImageGoal, LanguageGoal, ExperienceGoal, et al.

We focus on an instance of general rearrangement task specified by ExperienceGoal[35], which takes the agent's experience in goal state as goal specification and has two stages, **walkthrough** and **unshuffle**. In **walkthrough** stage, the agent is placed into a room with goal state s^* and should collect as much information as needed for that particular state of the room. Then we remove the agent from the room and change some objects' state. This state will be the initial state s_0, where the agent is initialized at the beginning of **unshuffle** stage and needs to convert s_0 to s^*. At each timestamp t, the agent receives egocentric RGB-D observations and executes a discrete action, where the action space consists of `move_ahead`, `turn_right`, `turn_left`, `look_down`, `look_up`, `pick`, `put`, `open`, `remove`, `push`, `stop`. The agent autonomously executes the action `stop` when it determines to complete the task.

3.2. Challenges and Insights

The primary challenges of the visual rearrangement task lie in addressing three distinct types of rigid object changes: **movement**, **appearance**, and **existence**. **Movement** changes involve the relocation of objects of varying sizes over different distances, even including cross-room transportation, requiring the agent to possess precise path planning and operation control. **Appearance** changes, aligned with the capabilities of simulators and requirements of real-world scenarios, mainly focus on the opening or closing state of containers in this paper. **Existence** changes pertain to the disappearance of objects (e.g., taken by a guest) or the occurrence of new items (e.g., empty bottles left behind). The agent must discover newly introduced objects that do not belong to the original room and remove them. Given the agent's inability to create objects from nothing, disappearance of originally existing objects is not considered in this paper.

In visual rearrangement task, the agent must infer and execute a sequence of actions to restore the scene to its initial state. This process requires exploring a reasoning space that encompasses all possible action sequences, approaching infinity with timestep. Existing methods infer object move-

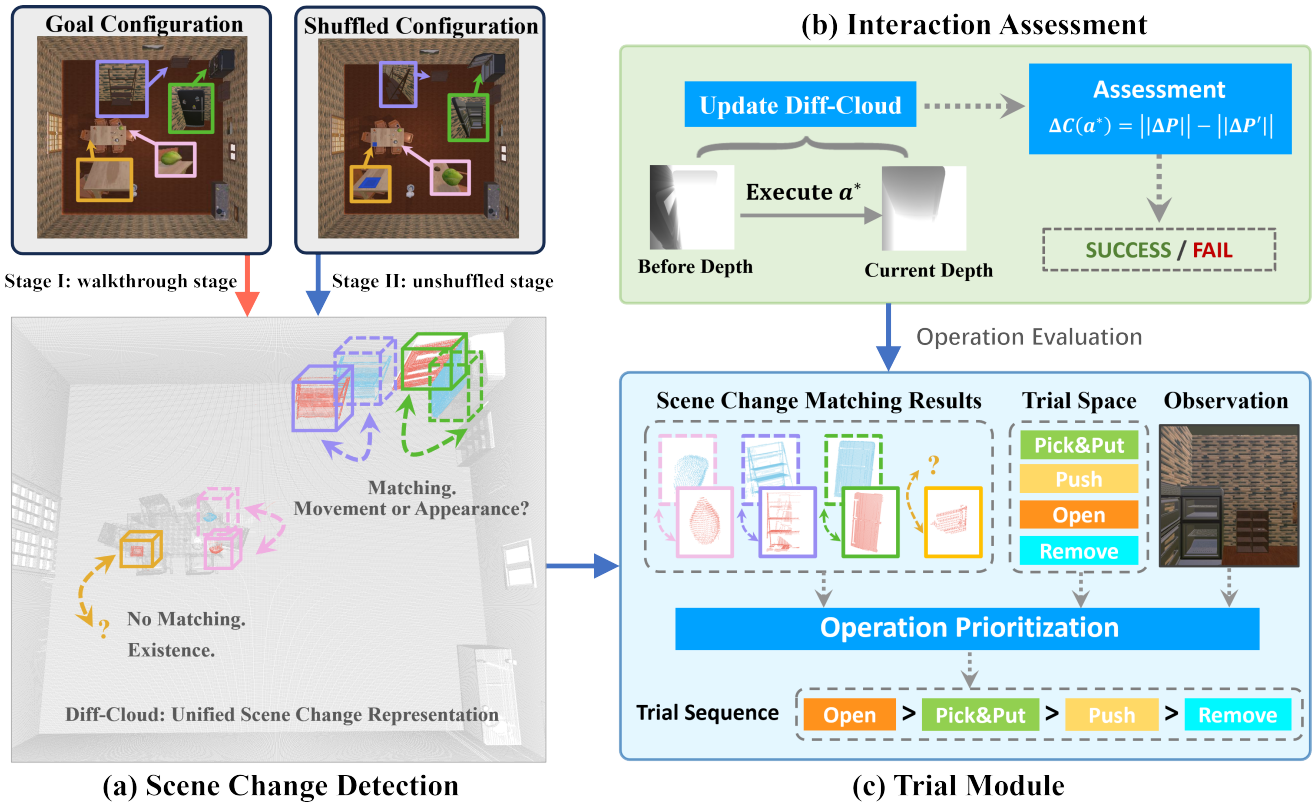


Figure 2. **Pipeline of our TOR model.** (a) To provide a unified representation of various scene changes, we build and maintain the diff-cloud dynamically, including the moved (blue points) and protruding (red points) point cloud in the shuffled configuration, compared to the goal configuration. (b) Interaction assessment module analyzes the variation of diff-cloud, evaluates effectiveness of operations, and provides feedback to trial module. (c) To transport each protruding cluster to its goal state, the trial module first evaluates and prioritizes all potential operations in trial space according to the scene information and then attempts operations sequentially based on the feedback from assessment module.

ment by scene change detection and matching, and convert the reasoning into action sequence, keeping complexity d_m manageable. However, when the agent needs to handle appearance and existence changes additionally, with complexities defined as d_a and d_e, the joint reasoning space approximates Cartesian product of these with complexity around $O(d_m * d_a * d_e)$, which is excessively large. To effectively pruning the reasoning space, we try to employ a trial mechanism, which iteratively attempts and prunes physically infeasible actions, turning reasoning problem into discrimination task. This shifts complexity from multiplicative to additive, reducing it to around $O(d_m + d_a + d_e)$.

4. Method

In this section, a modular approach, which includes scene change detection module, interaction assessment module, and trial module, is presented to tackle the challenging task, as shown in Fig.2. The scene change detection module (Sec.4.1) utilizes a dynamic differential point cloud to unify the representation of diverse scene changes (movement, appearance and existence). The interaction assessment module (Sec.4.2) evaluates the effectiveness of interaction behavior, providing the assessment of actions for trial module. Sec.4.3 introduces a trial strategy, which enables the agent to refine its operation based on not only the visual perception but also the feedback from assessment module, progressively accomplishing the task.

4.1. Scene Change Detection

4.1.1. Differential Point Cloud

Representing the diverse forms of object changes in a unified format is an important issue for the rearrangement setting. Several conventional choices, like voxel-based semantic maps and scene graph, can capture significant spatial movements but struggle with localized, fine-grained changes in appearance and position of objects. Our approach builds off recent work that use the differential point cloud (diff-cloud) to capture various scene changes precisely[24], as the point cloud contains rich geometric, positional, and scale information of objects and remains

robust against varied observation angles and obstructions from other objects. Our work differs from previous in that we maintain the diff-cloud dynamically, considering the interactions between the agent and the environment, such as moving objects (like furniture) or changing object states (like doors and stoves). In contrast, previous methods only construct diff-cloud based on static scene information and consider interaction with the environment only after the point cloud is fully constructed.

In particular, during walkthrough stage, the agent only explores the environment without interacting with it. We generate the egocentric point cloud p_w^{ego} using depth observation D. Given current pose x^w, we transform the egocentric point cloud p_w^{ego} from the agent's coordinate system to world coordinate system. The geocentric point cloud with current pose (p_w^{geo}, x^w) is recorded in the agent's memory.

Then during unshuffle stage, the agent operates in two modes of exploration and interaction alternately. For exploration mode, we also calculate the geocentric point cloud p_u^{geo} with pose x^u. If current view x^u aligns with a previous pose x^w existing in the memory, we contrast two corresponding point clouds and extract the moved and protruding parts of current configuration relative to goal configuration. For interaction mode, we refresh the diff-cloud within the current field of view rather than update it by adding up, so that the changes can be captured immediately.

4.1.2. Scene Change Matching

To recover the scene configuration to goal state, we need to match changes across various locations in the scene. Specifically, we first cluster the two components of the diff-cloud, the protruding part Ω^p and the moved part Ω^m, to obtain item-level information, yielding $\Omega^p = \{\omega_i^p\}$ and $\Omega^m = \{\omega_j^m\}$. Subsequently, we build off recent work[24] that trains a geometric feature extractor based on PointNet++[27] and conduct weighted bipartite graph matching between Ω^m and Ω^p, according to the similarity of geometric feature. If ω_i^p matches nothing, its associated variation is regarded as existence change; otherwise, it is regarded as movement or appearance change. The results of scene change matching will be considered in the following restoration strategy in Sec.4.3.

4.2. Assessment of Interaction

Although diff-cloud enables identification of diverse scene variations, determining the precise restorative interactions remains challenging. However, there exists a unified assessment method that can reliably determine whether each interaction effectively reduces the discrepancy between current and target scene configuration, making it possible to restore the scene by iteratively employing trial and assessing the effectiveness of each action as detailed in Sec.4.3.

Specifically, we evaluate the interaction behaviors by quantifying the impact on diff-cloud. Let ΔP_t denote the diff-cloud at time t. After executing an action a_t, the updated diff-cloud is represented as ΔP_{t+1}. To assess the effectiveness of the action, we compute the reduction in the discrepancy using the following metric:

$$\Delta C(a_t) = \|\Delta P_t\| - \|\Delta P_{t+1}\|$$

where $\|\cdot\|$ is a norm operator that quantifies the magnitude of the diff-cloud as the number of differing points. The action a_t is deemed effective if $\Delta C(a_t) > \tau$,, where τ is a predefined threshold, indicating a significant reduction in the discrepancy.

4.3. Trial Module

While it remains nontrivial to accurately plan the optimal action sequence for scene configuration rearrangement, multiple trials offer a viable path. The trial module leverages the scene observation and information accumulated from unsuccessful attempts to guide agent in refining its interaction strategy.

Specifically, to determine the type of change and the restoration strategy associated with each protruding cluster ω_i^m, we first prioritize all potential operations in the trial space, leveraging the results of scene change matching and current observations. The score function is defined as:

$$S(a, \omega_i^p) = \lambda_1 \phi_{\text{match}}(\omega_i^p, \Omega^m, a) + \lambda_2 \phi_{\text{context}}(\omega_i^p, o, a)$$

Here, $\phi_{\text{match}}(\omega_i^p, \Omega^m, a)$ is a function based on diff-cloud matching and determines whether the object is newly introduced or originally existing. If ω_i^p matches one moved cluster in Ω^m, ϕ_{match} returns 0 for "Throwout" or 1 for other actions, and it reverses the output if there's no match. The term $\phi_{\text{context}}(\omega_i^p, o, a)$ evaluates the applicability of action a based on the current observation, such as the object's shape, position, and environmental constraints. To calculate ϕ_{context}, we project cluster ω_i^p onto the current RGB observation and obtain bounding box and its local visual feature with RoI pooling on global feature map. Then ϕ_{context} predict a score based on the visual feature through a two layer MLP. To train this network, we collect a static dataset in various scenes containing the data of features and labels, where agent is randomly spawned around objects and attempts to interact with different actions and labels are annotated according to success or failure of these actions. λ_1 and λ_2 are weighting factors balancing the contributions of geometric matching and contextual reasoning.

The scores derived from static scene information provide valuable references for inferring restoration strategy, but are also prone to misjudgments, making the feedback from the assessment module crucial. Therefore, after the execution of the highest-scoring operation a^*, we evaluate its effectiveness by assessment module. If the operation successfully resolves the scene change associated with the protruding cluster ω_i^p, it is marked as restored, and the process advances to the subsequent item. In cases where the operation

fails, we iterate to the next highest-scoring option, continuing the trial process until a viable solution is identified or the cluster is deemed unsolvable. This integration of static scoring and dynamic trial allows the agent to effectively handle the various object changes in rearrangement setting.

5. Experiments

5.1. Experiment Setup

Dataset We evaluate our model on the official AI2THOR Rearrangement Challenge based on RoomR dataset [21, 35]. Due to bugs related to "open/close" action in the 2022 rearrangement challenge, we adopt the latest official dataset prior to submission for verification in this paper, where the bug is fixed to prevent exploitation for tricks. This dataset consists of 80 rooms with 4000 tasks for training, and 20 rooms with 1000 tasks separately for validation and test. In each task of RoomR dataset, the states of 1 to 5 objects are transformed, with changes in position or openness.

RoomR dataset only focusing on movement changes of pickable small objects and appearance changes of openable objects within one single room, whose action space and inference space are both limited to a subset of real-world rearrangement task settings. In order to better characterize the spatial complexity that fits the indoor environment in reality, we build a more practical and challenging dataset called Versatile Indoor Rearrangement (VIR) for the two-stage rearrangement task on the ProcTHOR simulator [7].

In our VIR dataset, object changes fall into three patterns: movement, appearance and existence, which cover the common morphological changes in daily life. For movement pattern, we consider the position variation of both small pickable objects and unpickable objects that cannot be grasped in hand and can only be pushed. For appearance pattern, we adopt the same settings as RoomR dateset, focusing on object openness. For the existence pattern, we introduce a new pickable object in unshuffle stage and expect the agent to identify and remove the redundant item. We randomly select 9000 scenes from ProcTHOR simulator and divide them into 6000 general settings (involving only pickable and openable object changes), 1500 movement settings (adding position change for one unpickable object), and 1500 existence settings (introducing a new object from general setting). For each scene, we randomly generate one rearrangement task. We split 7000, 1000, 1000 scenes for training, validation and test separately in VIR dataset, while retaining the same ratio of various settings in each split.

To characterize the complexity and difficulty of one rearrangement task, we define a scenario based task complexity:

$$Complexity_{scene} = \sum_{obj \in scene} C_{obj} * S_{scene}$$

where C_{obj} represents the state change of an object and S_{scene} represents the area measurements of the scene. For objects whose state has changed, we define the state change of an object through three patterns proposed in preceding context:

$$C_{obj} = \begin{cases} V_{obj} * |D_{walk} - D_{un}| & \text{Movement} \\ V_{obj} * \sin|O_{walk} - O_{un}| & \text{Appearance} \\ V_{obj} * \max(L_{scene}, W_{scene}) & \text{Existence} \end{cases}$$

where V_{obj} represents the volume of an object, and D_{walk}, D_{un}, O_{walk}, O_{un} respectively represent 3D position coordinates of pickable or moveable objects and openness of openable objects in walkthrough and unshuffle stage. Particularly, if an object is eliminate from walkthrough stage, its displacement distance is maximum size of the scene. Fig. 3 illustrates the comparison of task complexity distribution between our VIR dataset and RoomR dataset in the form of scatter plot. It can be seen that our VIR dataset expands the problem space and encompasses a diverse range of both task complexity and scene area, while RoomR dataset mainly covering a subset of ours in fields of low task complexity and small scene area.

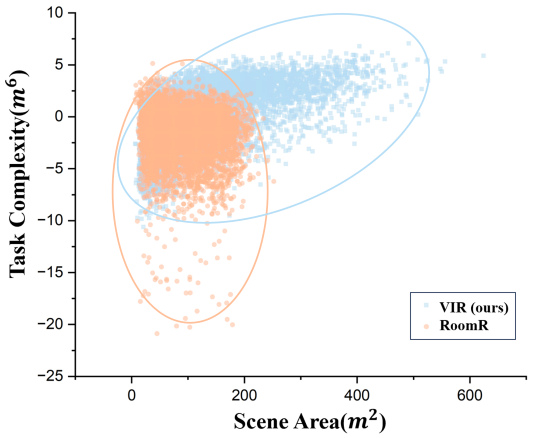


Figure 3. **Comparison of task complexity distribution between VIR and RoomR datasets.**

Evaluation metrics. Following Weihs[35], we employ four metrics to evaluate the agent's performance from different perspectives. **Success** metric measures the proportion of tasks for which the agent has restored all objects to their goal states. This metric is the most strict and unforgiving. We consider an object-level metric, **Fixed Strict**, which measures the proportion of objects successfully rearranged per task and is equal to 0 if there exists any newly misplaced objects. **Misplaced** metric equals the number of misplaced objects at the end of the episode divided by the number of misplaced objects at the start of the episode. Note that this metric can be larger than 1 if the agent, during the unshuffle stage, misplaces more objects than were originally misplaced at the start. The above metrics are quite

Table 1. Comparison on AI2THOR Rearrangement Challenge

Method	Validation				Test			
	Suc (%) ↑	FS (%) ↑	Mis ↓	E ↓	Suc (%) ↑	FS (%) ↑	Mis ↓	E ↓
TIDEE[29]	1.4	10.8	0.952	0.950	1.0	7.1	1.006	1.008
MaSS[31]	2.5	12.3	0.931	0.920	2.1	9.5	1.015	1.018
CAVR[24]	9.1	26.8	0.749	0.761	8.9	24.2	0.797	0.800
Ours	**20.9**	**44.1**	**0.575**	**0.572**	**18.5**	**40.6**	**0.633**	**0.639**

"Suc": Success; "FS": Fixed Strict; "Mis": Misplaced; "E": Energy Remaining.

Table 2. Comparison on our VIR dataset

Method	Validation				Test			
	Suc (%) ↑	FS (%) ↑	Mis ↓	E ↓	Suc (%) ↑	FS (%) ↑	Mis ↓	E ↓
TIDEE[29]	0.4	6.8	0.976	0.954	0.3	6.5	0.986	0.983
MaSS[31]	0.8	9.1	0.964	0.976	0.5	7.9	1.028	1.026
CAVR[24]	4.3	15.9	0.885	0.872	4.5	16.5	0.846	0.863
Ours	**12.1**	**28.2**	**0.734**	**0.744**	**11.3**	**27.1**	**0.752**	**0.772**

"Suc": Success; "FS": Fixed Strict; "Mis": Misplaced; "E": Energy Remaining.

strict and do not give any partial credit even if the agent restores objects to a state that is very close to the goal state.

% Energy Remaining metric is defined as the amount of energy remaining after the unshuffle stage divided by the total energy at the start of unshuffle stage, where the energy represents the difference between two state of an object and is defined by an energy function $D : S \times S \Rightarrow [0, 1]$.

5.2. Implementation Details

5.3. Comparisons with Related Works

We compared our model with three modular methods on the 2023 AI2THOR Rearrangement Challenge and our VIR dataset based on ProcTHOR simulator. The experimental results are reported in Table 1 and Table 2. We briefly introduce these three baselines as follows:

Mass: This model uses a search-based policy to rapidly find objects and builds voxel-based semantic map of the environment, which is leveraged to identify the movement of objects. After the inference of all rearrangement goals, this model transports them to their goal state in succession.

TIDEE: This method maintains the 2D occupancy map for exploration and navigation, and keeps track of objects and their labels over time. After the exploration of two stages, it infers the spatial relationship changes for all objects to identify the moved ones that need to be rearranged.

CAVR: This model is also designed for object movement, which leverages the observation distance map to explore the environment efficiently and performs scene change detection and scene change matching using point cloud, avoiding the reliance of category inference.

As shown in Table 1, the proposed model gains the best performance on all metrics. Specifically, our model significantly improves the scene-level success rate by 9.6% and the proportion of successfully restored objects by 16.2%. These highlights the capability of our model to handle the various object changes and restore the scene configuration entirely. In terms of error reduction, our method reduces the misplaced metric to 0.633, outperforming all baselines and demonstrating robust accuracy in correctly restoring objects without introducing new misplacement. The decrease in energy remaining indicates that TOR model can restore the scene configuration closer to goal state even when the task is not fully completed. As illustrated in Table 2, our method exhibits substantial advantages as object state changes become increasingly complex and diverse, achieving more than a twofold improvement in room-level success rates. This performance highlights the superiority of the trial-oriented interaction strategy in addressing such intricate tasks.

5.4. Ablations

We evaluate several variants of our model to study the following questions. First, we conduct ablation studies on RoomR test set to investigate the impact of two mechanisms (dynamic trial and static scoring) of trial module in Table.3. When both mechanisms are removed, the model is equivalent to the existing CAVR[24] method, which only addresses movement changes in the scene. In the variant with only dynamic trial (i.e., without static scoring), all potential operations in the trial space are randomly ordered and exe-

Table 3. Ablation Study

DT	SS	Suc (%)	FS (%)	E	Mis	TT
-	-	8.9	24.2	0.797	0.800	1.00
-	✓	11.3	31.5	0.737	0.742	1.00
✓	-	17.1	40.0	0.642	0.660	2.16
✓	✓	**18.5**	**40.6**	**0.633**	**0.639**	**1.33**

"✓" indicates corresponding mechanism is used while "–" indicates it is disabled; "DT":Dynamic Trial; "SS": Static Scoring; "Suc": Success; "FS": Fixed Strict; "E": Energy Remain; "Mis": Misplaced; "TT": Average Trial Times.

cuted sequentially until a successful restoration is achieved. Conversely, in the variant with only static scoring, the agent simply executes the highest-scoring operation, with no subsequent attempts even if the action fails. The experiment suggests that the dynamic trial mechanism contributes most in our method and the static scoring is mainly responsible for improving execution efficiency.

Additionally, we conduct experiments on the validation set of RoomR to determine hyper-parameters, including diff-cloud granularity and reduction threshold of assessment module. As shown in Fig.4, we set diff-cloud granularity from 0.1mm to 1cm and reduction threshold from 10 to 200, and calculate the average metrics of 1000 tasks with error bars based on 68% confidence interval. The results illuminate that our method achieves optimal performance and balanced computational efficiency with diff-cloud granularity of 0.01m and reduction threshold of 50, which is applied in other experiments in this paper.

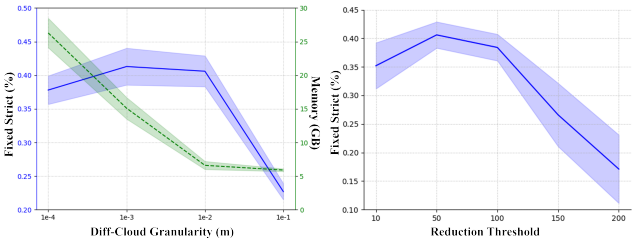


Figure 4. **Parameter ablations of diff-cloud granularity and reduction threshold.** Error bars represent a 68% confidence interval over 1000 tasks in validation set of RoomR.

5.5. Reasons For Task Failures

We explore and categorize the task failure reasons of different methods in Fig.5, which benefits the analysis of each approach's strengths and limitations while guiding subsequent improvement. There are four reasons that cover all possible situations: 1) The agent successfully identifies all objects exhibiting changes in the scene, yet fails to restore all objects to their goal states. 2) The agent fails to detect all changed objects. 3) The agent fails to complete the task within the allotted time. 4) The agent alters the state of objects not requiring rearrangement. The experiment suggests that compared to voxel map used by MaSS and semantic scene graph employed by TIDEE, diff-cloud representation significantly enhances the accuracy of scene change detection. Additionally, through the trial-oriented method, we improve the agent's ability to rearrange disagreement and reduce newly misplaced objects. And the main failure of our method is undetected changes, suggesting we can acquire large gains from improving the coverage and fidelity of diff-cloud building.

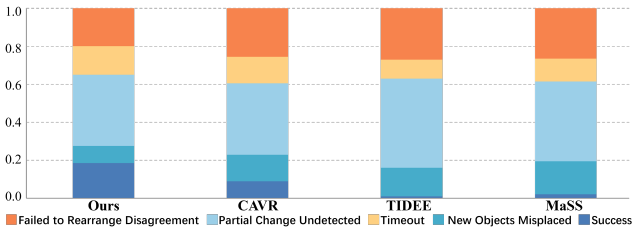


Figure 5. **Comparison of failure reasons distribution.** Each of the five colors represent the proportion of tasks that are solved or fail due to one of four reasons.

6. Conclusion

In this paper, we propose a trial-oriented model (TOR) for visual rearrangement task, which addresses the considerable complexity of diverse object changes, encompassing movement, appearance and existence. We construct a modular framework consisting of scene change detection, interaction assessment and trial module, which allows the agent to refine its operations from both visual observation and feedback of previous interaction assessment, thereby enhancing task performance. We compared our model with three previous modular methods on RoomR dataset and a more practical VIR dataset collected by us. The experiment results demonstrate the effectiveness of our model.

Acknowledgements: This work was supported by Beijing Natural Science Foundation under Grant JQ22012 and L242020, in part by National Natural Science Foundation of China under Grant 62125207, 62272443, 62032022, U23B2012, and 62495084.

References

[1] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020. 3

[2] Ohad Ben-Shahar and Ehud Rivlin. Practical pushing

planning for rearrangement tasks. *IEEE Transactions on Robotics and Automation*, 14(4):549–565, 1998. 3

[3] Akansel Cosgun, Tucker Hermans, Victor Emeli, and Mike Stilman. Push planning for object placement on cluttered table surfaces. In *2011 IEEE/RSJ international conference on intelligent robots and systems*, pages 4627–4632. IEEE, 2011.

[4] Michael Danielczuk, Andrey Kurenkov, Ashwin Balakrishna, Matthew Matl, David Wang, Roberto Martín-Martín, Animesh Garg, Silvio Savarese, and Ken Goldberg. Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1614–1621. IEEE, 2019. 3

[5] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2018. 3

[6] Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Neural modular control for embodied question answering. In *Conference on Robot Learning*, pages 53–62. PMLR, 2018. 3

[7] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *NeurIPS*, 2022. 6

[8] Mehmet R Dogar, Michael C Koval, Abhijeet Tallavajhula, and Siddhartha S Srinivasa. Object search by manipulation. *Autonomous Robots*, 36:153–167, 2014. 3

[9] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006. 2

[10] Linxi Fan, Yuke Zhu, Jiren Zhu, Zihua Liu, Orien Zeng, Anchit Gupta, Joan Creus-Costa, Silvio Savarese, and Li Fei-Fei. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In *Conference on robot learning*, pages 767–782. PMLR, 2018. 3

[11] Samir Yitzhak Gadre, Kiana Ehsani, Shuran Song, and Roozbeh Mottaghi. Continuous scene representations for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14849–14859, 2022. 1, 3

[12] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020. 2

[13] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4(1):265–293, 2021. 3

[14] Hector Geffner and Blai Bonet. *A concise introduction to models and methods for automated planning*. Morgan & Claypool Publishers, 2013.

[15] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning and acting*. Cambridge University Press, 2016. 3

[16] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 2

[17] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *2011 IEEE International Conference on Robotics and Automation*, pages 1470–1477. IEEE, 2011. 3

[18] Erez Karpas and Daniele Magazzeni. Automated planning for robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):417–439, 2020. 3

[19] Jennifer E King, Marco Cognetti, and Siddhartha S Srinivasa. Rearrangement planning using object-centric and robot-centric action spaces. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3940–3947. IEEE, 2016. 3

[20] Jennifer E King, Vinitha Ranganeni, and Siddhartha S Srinivasa. Unobservable monte carlo planning for nonprehensile rearrangement tasks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4681–4688. IEEE, 2017. 3

[21] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017. 2, 3, 6

[22] Athanasios Krontiris, Rahul Shome, Andrew Dobson, Andrew Kimmel, and Kostas Bekris. Rearranging similar objects with a manipulator using pebble graphs. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 1081–1087. IEEE, 2014. 3

[23] Weijie Li, Xinhang Song, Yubing Bai, Sixian Zhang, and Shuqiang Jiang. Ion: Instance-level object navigation. In *Proceedings of the 29th ACM international conference on multimedia*, pages 4343–4352, 2021. 2

[24] Yuyi Liu, Xinhang Song, Weijie Li, Xiaohan Wang, and Shuqiang Jiang. A category agnostic model for visual rearrangment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16457–16466, 2024. 3, 4, 5, 7

[25] Medhini Narasimhan, Erik Wijmans, Xinlei Chen, Trevor Darrell, Dhruv Batra, Devi Parikh, and Amanpreet Singh. Seeing the un-scene: Learning amodal semantic maps for room navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 513–529. Springer, 2020. 2

[26] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017. 2

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 5

[28] Santhosh K Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman. An exploration of embodied visual explo-

ration. *International Journal of Computer Vision*, 129(5): 1616–1649, 2021. 2

[29] Gabriel Sarch, Zhaoyuan Fang, Adam W Harley, Paul Schydlo, Michael J Tarr, Saurabh Gupta, and Katerina Fragkiadaki. Tidee: Tidying up novel rooms using visuo-semantic commonsense priors. In *European conference on computer vision*, pages 480–496. Springer, 2022. 3, 7

[30] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 2

[31] Brandon Trabucco, Gunnar Sigurdsson, Robinson Piramuthu, Gaurav S Sukhatme, and Ruslan Salakhutdinov. A simple approach for visual rearrangement: 3d mapping and semantic search. *arXiv preprint arXiv:2206.13396*, 2022. 3, 7

[32] Beibei Wang, Xiaohan Wang, and Yuehu Liu. Learning object relation graph and goal-aware policy for visual room rearrangement. In *2023 China Automation Congress (CAC)*, pages 5035–5040. IEEE, 2023. 3

[33] Xiaohan Wang, Yuehu Liu, Xinhang Song, Beibei Wang, and Shuqiang Jiang. Generating explanations for embodied action decision from visual observation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2838–2846, 2023. 2

[34] Xiaohan Wang, Yuehu Liu, Xinhang Song, Beibei Wang, and Shuqiang Jiang. Camp: Causal multi-policy planning for interactive navigation in multi-room scenes. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[35] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5922–5931, 2021. 1, 3, 6

[36] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied question answering in photorealistic environments with point cloud perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6659–6668, 2019. 3

[37] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. 3

[38] Haitao Zeng, Xinhang Song, and Shuqiang Jiang. Multi-object navigation using potential target position policy function. *IEEE Transactions on Image Processing*, 32:2608–2619, 2023. 2

[39] Sixian Zhang, Xinhang Song, Yubing Bai, Weijie Li, Yakui Chu, and Shuqiang Jiang. Hierarchical object-to-zone graph for object navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15130–15140, 2021.

[40] Sixian Zhang, Weijie Li, Xinhang Song, Yubing Bai, and Shuqiang Jiang. Generative meta-adversarial network for unseen object navigation. In *European Conference on Computer Vision*, pages 301–320. Springer, 2022.

[41] Sixian Zhang, Xinhang Song, Weijie Li, Yubing Bai, Xinyao Yu, and Shuqiang Jiang. Layout-based causal inference for object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10792–10802, 2023. 2