RoboEXP: Action-Conditioned Scene Graph via Interactive Exploration for Robotic Manipulation

Hanxiao Jiang[1] Binghao Huang[1] Ruihai Wu[3] Zhuoran Li[4]
Shubham Garg[2] Hooshang Nayyeri[2] Shenlong Wang[1] Yunzhu Li[1]
[1]University of Illinois Urbana-Champaign [2]Amazon [3]Peking University [4]National University of Singapore

CONTENTS

I. RELATED WORKS

Scene graphs [1, 2] represent objects and their relations [3–5] in a scene via a graph structure. Previous studies generate scene graphs from images [6, 2] or 3D scenes [7] with hierarchical and semantic information, and further with the assistance of large language models (LLMs) [8]. They leverage scene graphs for image captioning [9, 10], image retrieval and generation [1, 11], visual-language tasks [3, 12], navigation [13, 14] and task planning [15–17]. While previous works model scene graphs in static 2D or 3D scenes, we generate action-conditioned scene graphs that integrate actions as core elements, depicting interactive relationships between objects and actions. This action-centric approach opens avenues for physical exploration and diverse downstream robotics tasks.

Neuro-symbolic representations integrates neural networks' perceptual abilities with the symbolic reasoning for robots in complex and dynamic environments. Prior works explored understanding scenes and describing robotic skills in symbolic texts to interpret demonstrations [18, 19], ground abstract actions for robotic primitives [20] and generate action plans [21–24]. Our proposed framework also constructs symbolic representations of the environment, but in the form of action-conditioned scene graphs for robotic manipulation.

Robotic exploration aims to autonomously navigate, interact with, and gather information from environments it has never encountered before. It is applicable in search and rescue [25–33], planetary exploration [34–37], object goal navigation [38–57], and mobile manipulation [58–63]. The primary guiding principle behind robotic exploration is to reduce the uncertainty of the environment [64, 65, 27, 66–68], making uncertainty quantification key for robotic exploration tasks. Curiosity-driven exploration has recently emerged as a promising approach, showing effective results in various contexts [69–72]. Most past works have focused on exploration in the context of mobility [73, 25–31, 38–54, 58–63], with the primary goal of modeling and understanding the static environment to complete specific tasks. Recently, exploration has also been studied in the context of manipulation [74–79], aiming to better understand the scene by changing the state of the environment. Our work introduces a new active exploration strategy for manipulation, uniquely defining a novel scene graph-guided objective to guide the exploration process.

Active perception aims to select specific actions for an agent to improve its ability to perceive and understand the environment [80, 81]. Unlike passive perception, actions offer more flexibility, such as control over better viewpoints [82–84], sensor configurations [85, 86], or adjustments to environmental configurations [87]. It can also reveal certain scene properties that cannot be perceived in a passive manner, such as dynamic parameters [69, 88] or articulation [89, 77, 90]. Previous studies have explored active perception in 3D reconstruction [91, 92, 79, 93, 94], object recognition [95–97], camera localization [98], and robotic manipulation [99, 100]. Our work falls into the category of actively exploring the environment to reveal what's inside or underneath objects. Differing from most previous active perception efforts, which are driven by handcrafted rules [101], information gain [102, 103], or reinforcement learning [69, 104], our approach to active perception is guided by grounding the rich commonsense knowledge encoded in a large language model into an explicit scene graph representation.

Language models for robotics. Large language models (LLMs) [105–107] and large multimodality models (LMMs) [108, 109] are bringing overwhelming influence into the robotics field, for their strong capacity in common-sense knowledge and long-horizon reasoning. Previous studies have harnessed the common-sense knowledge of such large models to generate action candidates [110] and action sequences for task planning [111, 107, 112, 57], and generate code for robotic control and manipulation [113–115]. More recently, VILA [116] utilized GPT-4V [108, 109] for vision-language

planning. In our RoboEXP system, we leverage GPT-4V for decision-making in two crucial roles. First, as the *action proposer*, it ensures both effectiveness and efficiency in proposing appropriate strategies to expand potential nodes in our action-conditioned 3D scene graph. Second, as the *action verifier*, it ensures the plausibility and smoothness of actions and operations in our system. Moreover, instead of memorizing everything using large models in a brute force way, our system employs explicit memory to enhance the decision-making process.

II. Additional Details of Problem Statement

We provide more details on the definition of our action-conditioned 3D scene graph and our interactive scene exploration in this section.

A. Action-Conditioned 3D Scene Graph

An action-conditioned 3D scene graph (ACSG) is an actionable, spatial-topological representation that models objects and their interactive and spatial relations in a scene. Formally, ACSG is a directed acyclic graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ where each node represents either an object (e.g., a *door*) or an action (e.g., *open*), and edges $\mathbf{E}$ represent their interaction relations. The object node $\mathbf{o}_i = (\mathbf{s}_i, \mathbf{p}_i) \in \mathbf{V}$ encodes the semantics and geometry of each object (e.g., the semantic embedding of a fridge $\mathbf{s}_i$, and its shape in the form of a point cloud $\mathbf{p}_i$), whereas the action node $\mathbf{a}_k = (a_k, \mathbf{T}_k) \in \mathbf{V}$ encodes high-level action type a_k and low-level primitives $\mathbf{T}_k$ to perform the actions. Between the nodes are edges encoding their relations, which we categorize into four types: 1) between objects $\mathbf{e}_{\mathbf{o}\to\mathbf{o}}$ (e.g., the *door handle belongs* to the *fridge*), 2) from objects to actions $\mathbf{e}_{\mathbf{o}\to\mathbf{a}}$ (e.g., *toy* can be *picked up*), 3) from action to objects $\mathbf{e}_{\mathbf{a}\to\mathbf{o}}$ (e.g., a *banana* can be reached if we *open* the cabinet), or 4) from one action to another $\mathbf{e}_{\mathbf{a}\to\mathbf{a}}$ (e.g., the cabinet can be *opened* only if we *move away* the *condiment*). Our action-conditioned 3D scene graph greatly enhances existing 3D scene graphs, as it explicitly models the action-conditioned relations between objects.

B. Interactive Exploration

We formulate the interactive scene exploration task into an active perception and exploration problem to construct the action-conditioned 3D scene graph (ACSG) (see Algorithm 1).

The Algorithm 1 simply mentions "add spatial relations" and "add action preconditions" as part of the function of the memory module, but without detailed explanation. In the algorithm, we have demonstrated how to construct the edges from objects to actions $\mathbf{e}_{\mathbf{o}\to\mathbf{a}}$ and from actions to objects $\mathbf{e}_{\mathbf{o}\to\mathbf{a}}$; however, there is a lack of description for the other two types of edges.

Add Spatial Relations. The logic involves analyzing the spatial relationships among objects using spatial heuristics and incorporating the resulting spatial relation edges between objects $\mathbf{e}_{\mathbf{o}\to\mathbf{o}}$ (see Algorithm 2).

Add Action Preconditions. The approach is to assess the feasibility of implementing the actions. We utilize the

Algorithm 1 Interactive Exploration

input: $\mathbf{O}^0$, $\mathbf{G}^0 = (\mathbf{V}^0, \mathbf{E}^0), \mathbf{U}^0 \leftarrow \mathbf{V}^0$
while $|\mathbf{U}^{t-1}| \neq 0$ **do**
  if choose object $\mathbf{o}_i \in \mathbf{U}^{t-1}$ **then** % explore object
    add spatial relations % memory
    obtain action $\mathbf{a}$ to explore $\mathbf{o}_i$ % decision-making
    if action $\mathbf{a} \notin \mathbf{V}^{t-1}$ **then**
      $\mathbf{V}^t, \mathbf{U}^t = \mathbf{V}^{t-1} \cup \{\mathbf{a}\}, \mathbf{U}^{t-1} \cup \{\mathbf{a}\}$ % add node
      $\mathbf{E}^t = \mathbf{E}^{t-1} \cup \{\mathbf{e}_{\mathbf{o}_i\to\mathbf{a}}\}$ % add edge
      $\mathbf{U}^t = \mathbf{U}^t \setminus \mathbf{o}_i$ % mark as explored
    end if
  else choose action $\mathbf{a}_k \in \mathbf{U}^{t-1}$
    if no obstruction **then** % decision-making
      take action $\mathbf{a}_k$ % action
      obtain new observation $\mathbf{O}^t$ % perception
      if found new objects $\mathcal{O} \notin \mathbf{V}^{t-1}$ **then**
        $\mathbf{V}^t, \mathbf{U}^t = \mathbf{V}^t \cup \{\mathcal{O}\}, \mathbf{U}^{t-1} \cup \{\mathcal{O}\}$ % add nodes
        $\mathbf{E}^t = \mathbf{E}^t \cup \{\mathbf{e}_{\mathbf{a}_k\to\mathcal{O}}\}$ % add edges
        $\mathbf{U}^t = \mathbf{U}^t \setminus \mathbf{a}_k$ % mark as explored
      end if
    else
      add action preconditions % memory
    end if
  end if
end while
output: $\mathbf{G}^t$ % final scene graph

Algorithm 2 Add Spatial Relations

input: $\mathbf{G}^{t-1} = (\mathbf{V}^{t-1}, \mathbf{E}^{t-1})$
$\mathbf{E}^t = \mathbf{E}^{t-1}$
for $\mathbf{o} \in \mathbf{V}^{t-1}$ **do** % check relations
  if relation from $\mathbf{o}$ to $\mathbf{o}_i$ **then** % memory
    $\mathbf{E}^t = \mathbf{E}^t \cup \{\mathbf{e}_{\mathbf{o}\to\mathbf{o}_i}\}$ % add edge
  end if
  if relation from $\mathbf{o}_i$ to $\mathbf{o}$ **then**
    $\mathbf{E}^t = \mathbf{E}^t \cup \{\mathbf{e}_{\mathbf{o}_i\to\mathbf{o}}\}$ % add edge
  end if
end for
output: $\mathbf{G}^t$ % new scene graph

Algorithm 3 Add Action Preconditions

input: $\mathbf{G}^{t-1} = (\mathbf{V}^{t-1}, \mathbf{E}^{t-1}), \mathbf{U}^{t-1}$
if object $\mathbf{o}$ obstruct **then** % decision-making
  choose action $\mathbf{a}$
  $\mathbf{V}^t = \mathbf{V}^{t-1} \cup \{\mathbf{a}\}$, $\mathbf{U}^{t-1} \cup \{\mathbf{a}\}$ % add node
  $\mathbf{E}^t = \mathbf{E}^{t-1} \cup \{\mathbf{e}_{\mathbf{o}\to\mathbf{a}}\}$ % add edge
  $\mathbf{E}^t = \mathbf{E}^{t-1} \cup \{\mathbf{e}_{\mathbf{a}\to\mathbf{a}_k}\}$ % add edge
end if
output: $\mathbf{G}^t, \mathbf{U}^t$ % new scene graph & plan

(a) Prompts of Proposer

System: You are an assistant tasked with aiding in the construction of a complete scene graph for a tabletop environment. The objective is to identify all objects hidden from the current observation in the tabletop setting. Your role involves selecting appropriate actions or opting not to take any action based on commonsense knowledge in response to queries with current observations. Your responses will guide a robot in efficiently exploring the environment. Approach each step thoughtfully, and analyze the fundamental problem deeply, considering the potential vagueness or inaccuracy in the queries. Adhere to the provided formats in your instructions.

User: Analyze and provide your final answer for each new query object/part category, considering the given surrounding objects and observations in the tabletop scene from different viewpoints. The query object/part will be enclosed in a green bounding box, though it may not always be fully accurate. Format your responses as follows: "[Analysis]: <your reasoning process>; \n\n [Final Answer]: <skill>". Be comprehensive and avoid repeating my question. Choose from three skills: 1. Open the doors or drawers. 2. Pick up / Open the top object. 3. No action. The primary goal is to select an action that has the potential to reveal hidden objects. The secondary goal is to act efficiently, performing only necessary actions to uncover hidden objects. For example, if an object contains doors or drawers and can potentially store something inside, opt for the first skill "Open the doors or drawers". If an object has no bottom side and can potentially cover something beneath it, choose the second skill " Pick up / Open the top object"; otherwise, select the third skill "No action" to ensure efficiency.

Assistant: Got it. I will output the reasoning process step-by-step, explain why I choose the skill but not others and follow the output format.

User: [Query Object] + [Query Images]

Assistant: [Reply from GPT-4V]

(b) Prompts of Verifier

System: You are an assistant tasked with evaluating the feasibility of actions within a tabletop environment. Your role is to select suitable objects that could obstruct open actions based on queries and current observations. Provide guidance for a robot's planning process. Approach each step thoughtfully, analyzing the underlying problem thoroughly while considering potential vagueness or inaccuracy in the queries. Follow the provided formats in your instructions.

User: Provide an analysis and your final answer each time I present a new query object/part category, the list of surrounding objects you need to consider and observations of the corresponding in the tabletop scene from different viewpoints. The query object/part is enclosed in a green bounding box, which may not always be fully accurate. Present your reasoning process and final answer in the format "[Analysis]: <your reasoning process>; \n\n [Final Answer]: <list of objects>". Be comprehensive and avoid repeating my question. Use the given list of surrounding objects, maintaining the provided names. Only consider the surrounding objects in the given list. The objective is to identify all objects that could potentially block open actions. If an object obstructs the door or drawer from opening, include it in the final list of objects. Analyze the action movement and identify the blocking objects.

Assistant: Got it. I will output the reasoning process step-by-step, explain why I choose the object but not others and follow the output format.

User: [Query Object] + [Query Images]

Assistant: [Reply from GPT-4V]

Fig. 1: **Prompts of the Decision-Making module.** We present the full prompts for the two pivotal roles of our decision-making module, **proposer** in (a), **verifier** in (b). The prompts are used for all our experiments without modification and extra examples.

Fig. 2: **All Testing Objects.** We present various objects utilized in our work, encompassing different types of cabinets, fruits, dolls, condiments, beverages, food items, tapes, tableware, and fabric.

decision-making module to verify whether there are any prerequisite actions that need to be completed beforehand, and then adjust the plan accordingly (see Algorithm 3).

For the reward defined in the main paper, intuitively, to maximize this reward at each discrete timestamp, we should prioritize exploring the unexplored nodes in the current scene graph that are likely to lead to the discovery of new nodes (e.g., opening a cabinet that has not been opened, or lifting a piece of clothing that might cover a small object). The key challenge lies in how we can perceive the objects in the scene, infer possible actions and their relations from the sensory data, and take actions with the current scene graph.

III. Additional Details of RoboEXP System

In this section, we outline the detailed structure of our RoboEXP system, including perception, memory, decision-making, and action modules, in Sec. III-A. We then discuss our system's design for the interactive scene exploration task and the usage of our system in following sections, focusing on its application in closed-loop exploration processes that may require multi-step or recursive reasoning and handle potential interventions.

A. RoboEXP System

Our system comprises four key components: perception, memory, decision-making, and action modules. Raw RGBD images are captured through the wrist camera in different viewpoints and processed by the perception module to extract scene semantics, including object labels, 2D bounding boxes, segmentations, and semantic features. The obtained semantic information is then transmitted to the memory module, where the 2D data is merged into the 3D representation. Such 3D information serves as a valuable guide for the decision module, aiding in the selection of appropriate actions to further interact or observe the environment and unveil hidden objects. The action module is activated to execute the planned action, generating new observations for the perception modules. This closed-loop system ensures the thoroughness of our task in interactive scene exploration.

Perception Module. Given multiple RGBD observations from different viewpoints, the objective of the perception

module is to detect and segment objects while extracting their semantic features. To enhance generality, we opt for the open-vocabulary detector GroundingDINO [117] and the Segment Anything in High Quality (SAM-HQ) [118], an advanced version of SAM [119]. For the extraction of semantic features used in subsequent instance merging within the memory module, we employ CLIP [120]. To obtain per-instance CLIP features, we implement a strategy similar to the one proposed by Jatavallabhula et al. [121]. Specifically, we extend the local-global image feature merging approach by incorporating additional label text features to augment the semantic CLIP feature for each instance. Furthermore, we exclusively focus on instance-level features, disregarding pixel-level features, thereby accelerating the entire semantic feature extraction process.

Memory Module. The memory module is designed to construct our ACSG of the environment by assimilating observations over time. For the low-level memory, to ensure stable instance merging from 2D to 3D, we employ a similar instance merging strategy as presented in Lu et al. [122], consolidating observations from diverse RGBD sources across various viewpoints and time steps. In contrast to the original algorithm, which considers only 3D IoU and semantic feature similarity we additionally incorporate label similarity and instance confidence. To enhance algorithm efficiency, we represent low-level memory using a voxel-based representation, which allows for more efficient computation and memory updates. Meanwhile, given the crowded nature of objects in our tabletop setting, we have implemented voxel-based filtering designs to obtain a cleaner and more complete representation of the objects for storage in our memory.

The memory module handles merging across different viewpoints and time steps. To merge across different viewpoints, we project 2D information (RGBD, semantic features, mask, bounding box) to 3D and leverage the instance merging strategy mentioned earlier to attain consistent 3D information. Addressing memory updates across time steps presents a challenge due to dynamic changes in the environment. For instance, a closed door in the previous time step may be opened by our robot in the current time step. To accurately reflect such changes, our algorithm evaluates whether elements within our memory have become outdated, primarily through depth tests based on the most recent observations. This process ensures that the memory accurately represents the environment's current state, effectively managing scenarios where objects may change positions or states across different time steps.

For the high-level graph of our ACSG, the memory module analyzes the relationships between objects and the logical associations between actions and objects. Depending on changes in low-level memory and relationships, the memory module is tasked with updating the graph. This involves adding, deleting, or modifying nodes and edges within our graph.

Decision-Making Module. The primary goal of the decision module is to identify the appropriate object and corresponding skill to enhance the effectiveness and efficiency of interactive scene exploration. In the context of our task, distinct objects may necessitate distinct exploration strategies. While humans can easily discern the most suitable skill to apply (e.g., picking up the top Matryoshka doll to inspect its contents), achieving such decisions through heuristic-based methods is challenging. The utilization of a Large Multi-Modal Model (LMM), such as GPT-4V [108, 109], shows instrumental in addressing this difficulty, as it captures commonsense knowledge that facilitates decision-making.

The LMM brings commonsense knowledge to our decision-making process and serves in two pivotal roles. Firstly, it functions as an action proposer (Fig. 1a). Given the current digital environment from the memory module, GPT-4V is tasked with selecting the appropriate skill for unexplored objects in our system. For instance, when presented with a visual prompt of an object within a green bounding box from various viewpoints, GPT-4V can discern the suitable "pick up" skill for the Matryoshka doll in the environment. For unexplored objects, our ACSG includes the attribute of whether each object node is explored or unexplored. GPT-4V, in its role as the proposer, also functions to assess whether the object holds value for further exploration. If not, the corresponding node is marked as explored, indicating that no further actions are needed.

Secondly, the LMM also serves as the action verifier (Fig. 1b). For the proposer role, it analyzes the object-centric attributes and doesn't consider surrounding information when choosing the proper skill. For example, if the proposed action involves opening a door, the proposer alone may struggle with cases where obstructions exist in front of the door (e.g., a condiment bottle). To address this, we use another LMM program to verify the feasibility of the action and identify any objects in the scene that may impede the action based on information from our ACSG.

In summary, the decision module, with its dual roles, effectively guides our system to choose efficient actions that minimize uncertainty in the environment and successfully locate all relevant objects.

Action Module. In the action module, our primary focus is on autonomously constructing the ACSG through effective and efficient interaction with the environment. We employ heuristic-based action primitives within our action module, leveraging the geometry cues in our ACSG. These primitives encompass seven categories: "open the door", "open the drawer", "close the door", "close the drawer", "pick object to idle space", "pick back object", "move wrist camera to position". Strategic utilization of these skills plays a pivotal role in accomplishing intricate tasks seamlessly within our system.

For the door and drawer relevant primitives, engagement with handles is required. In our implementation, we exploit the handle's position and geometry to discern its motion type (prismatic or revolute) and motion parameters (motion axis and motion origin). Executing this action involves utilizing the detected handle and its geometry to adeptly open doors or drawers. Upon identifying the specific handle to be operated, our system retrieves the point cloud converted from our

System: You are an assistant tasked with aiding in the construction of a complete scene graph for a tabletop environment. The objective is to identify all objects hidden from the current observation in the tabletop setting. Your role involves selecting appropriate actions or opting not to take any action based on commonsense knowledge in response to queries with current observations. Your responses will guide a robot in efficiently exploring the environment. Approach each step thoughtfully, and analyze the fundamental problem deeply, considering the potential vagueness or inaccuracy in the queries. Adhere to the provided formats in your instructions.

User: Analyze and provide the current scene graph and your final answer for the next action given the latest observations in the tabletop scene from different viewpoints. Each time you need to pick an action to do or choose "Done" to terminate. The action you can choose should be composed of (<object/part>, <skill>). Be specific on which object or part you refer to. The skills you can choose: [1. Open the door. 2. Close the door. 3. Open the drawer. 4. Close the drawer. 5. Pick up the object to idle space. 6. Pick back the object from the idle space]. Each time after you choose an action, you will receive the new observations after the action. Format your responses as follows: "[Analysis]: <your reasoning process>; \n\n [Scene Graph]: <current scene graph> \n\n [Final Answer]: <skill>". Be comprehensive and avoid repeating my question. The primary goal is to select an action that has the potential to reveal hidden objects. The secondary goal is to act efficiently, performing only necessary actions to uncover hidden objects. The third goal is to make the object go back to the initial state after exploration. For the output scene graph, you need to output all the objects in the scene, including those found during the exploration process.

Assistant: Got it. I will output the reasoning process step-by-step, explain why I choose the skill but not others and follow the output format.

User: [Query Images]

Assistant: [Reply from GPT-4V]

User: [Query Images]

Assistant: [Reply from GPT-4V]

...

Fig. 3: **Prompts of the GPT-4V baseline.** To ensure fairness in comparison to this baseline, we choose to use similar prompts, employing the chain-of-thoughts technique to enhance its performance.

voxel-based representation corresponding to that handle from our memory module. Subsequently, we employ Principal Component Analysis (PCA) to determine the principal direction of the handle, aiding in aligning the gripper for optimal engagement. Additionally, understanding the opening direction is pivotal for effectively handling doors or drawers. To ascertain this, we analyze neighboring points and deduce the most common normal as the opening direction. The combined information of the handle direction and the opening direction provides sufficient guidance for our robot arm to grasp the handle and open the prismatic part. However, in the case of a revolute joint, the motion becomes more intricate. Therefore, we further utilize the motion parameters inferred from the geometry to simulate the evolving opening direction based on the revolute joint's opening process. This well-designed heuristic empowers our system to reliably open drawers or doors in our tabletop setting.

For the pickup-related primitives, we simplify the pickup logic to exclusively consider a top-down direction. Consequently, our focus narrows down to acquiring essential information such as the object's height and xy location. We achieve this by extracting the object's point cloud from its associated voxel-based representation. Subsequently, we pinpoint the highest points within the cloud, calculating their mean to determine the optimal pickup point. This calculated point serves as a precise reference for our gripping mechanism, facilitating the successful grasping of objects in the specified direction.

Regarding viewpoint change, the primitive is parameterized with the expected pose. For example, after opening the door/drawer, to see inside, we develop the heuristic to choose the proper viewpoint from the open direction as the parameter for the primitive, allowing for the implementation of the action primitive.

B. Other Design in Interactive Exploration

One desiderata for robot exploration is the ability to handle scenarios that necessitate multi-step or recursive reasoning. An example of this is the Matryoshka doll case (Fig. 5b), which cannot be addressed using previous one-step LLM-based code generation approaches [116, 114]. In contrast, our modular design allows agents to dynamically plan and adapt in a closed-loop manner, enabling continuous LLM-based exploration based on environmental feedback.

To manage multi-step reasoning, our system incorporates an action stack as a simple but effective "planning" module. Guided by decisions from the decision module, the stack structure adeptly organizes the order of actions. For instance, upon picking up the top Matryoshka doll, if the perception and memory modules identify another smaller Matryoshka doll in the environment, the decision module determines to pick it up. Our action stack dynamically adds this pickup action to the top of the stack, prioritizing the new action over picking back the previous, larger Matryoshka doll. This stack structure facilitates multi-step reasoning and constructs the system's logic in a deep and coherent structure.

Moreover, for the interactive scene exploration task, maintaining scene consistency is crucial in practice (e.g., the agent should close the fridge after exploring it). We employ a greedy strategy returning objects to their original states. This approach keeps the environment close to its pre-exploration state, making RoboEXP more practical for real-world applications.

C. Usage of ACSG

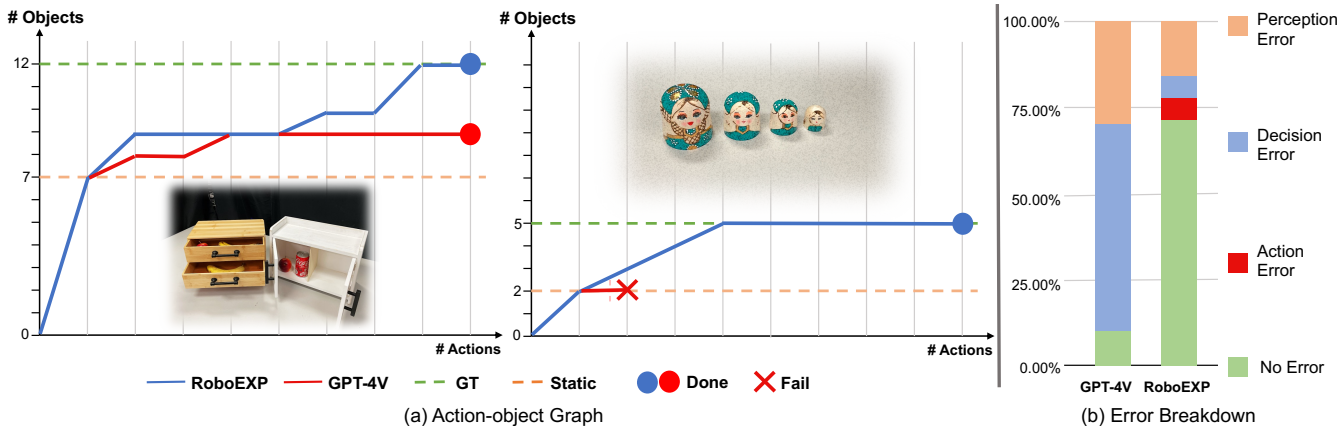
The ACSG constructed during the exploration stage shows beneficial for scenarios that require a comprehensive understanding of scene content and structure, such as household environments like kitchens and living rooms, office environments, etc. We list several examples illustrating the potential usage of the scene graph in various tasks.

Judging Object Existence. A direct application of our ACSG is to determine the presence or absence of specific objects in the current environment. For instance, during the exploitation stage of the scenario to set the dining table, if the spoon is missing, the robot can further seek human assistance.

Object Retrieval. One notable advantage of our ACSG is its ability to capture all actions and their preconditions.

TABLE I: **Quantitative Results on Different Tasks.** We compare the performance of both the GPT-4V baseline and our system across various tasks. We assess the outcomes using five distinct metrics to illustrate diverse facets of the interactive exploration process. Our system consistently outperforms the baseline across all tasks and metrics.

Task (10 variance for each)	Drawer-Only		Door-Only		Drawer-Door		Recursive		Occlusion	
Metric	GPT-4V	Ours	GPT-4V	Ours	GPT-4V	Ours	GPT-4V	Ours	GPT-4V	Ours
Success % ↑	20±13.3	**90**±10.0	30±15.2	**90**±10.0	10±10.0	**70**±15.3	0±0.0	**70**±15.3	0±0.0	**50**±16.7
Object Recovery % ↑	83±11.0	**97**±3.3	50±16.7	**100**±0.0	62±10.7	**91**±4.7	20±13.3	**80**±11.7	17±11.4	**67**±14.9
State Recovery % ↑	60±16.3	**100**±0.0	80±13.3	**100**±0.0	70±15.3	**100**±0.0	70±15.3	**100**±0.0	10±10.0	**70**±15.3
Unexplored Space % ↓	15±7.6	**0**±0.0	40±14.5	**0**±0.0	25±6.5	**0**±0.0	63±15.3	**15**±8.9	85±7.6	**30**±15.3
Graph Edit Dist. ↓	2.8±1.04	**0.2**±0.20	4.4±1.42	**0.1**±0.10	5.6±1.46	**0.5**±0.27	8.8±2.06	**2.1**±1.49	7.3±0.97	**2.5**±1.15



(a) Action-object Graph (b) Error Breakdown

Fig. 4: **Visualization of Quantitative Results.** (a) The action-object graph captures the change in the number of discovered objects relative to the number of actions taken. Our RoboEXP efficiently discovers all objects. Sometimes, the object count doesn't increase during actions due to the absence of objects in storage after opening. Additionally, some actions are employed to restore the scene state (e.g., closing the door after exploration). (b) The error breakdown of all our quantitative experiments includes 5 task settings with 10 variations each. We categorize errors into perception, decision, action, and no-error cases. For the GPT-4V baseline, manual assistance in action execution eliminates failure cases, serving as an upper bound for baseline performance. Even in this scenario, our RoboEXP largely outperforms the baseline.

Utilizing this information, retrieving any object becomes straightforward by following the graph structure and executing actions in topological order along the paths from the root to the target object node. For example, in the obstruction scenario, the ACSG can provide the sequence of actions required to fetch the tape: 1) removing the condiment blocking the cabinet door, 2) opening the cabinet via the door handle, and 3) retrieving the tape. Such insights are crucial for tasks like cooking.

Advanced Usage. The high-level representation of the environment provided by our ACSG serves as a simplified yet effective model. Similar to the approach proposed by Gu et al. [123], integrating the scene graph with Large Language Models (LLM) or Large Multi-modal Models (LMM) offers enhanced capabilities, including natural language interaction. This enables the robot to respond to human preferences expressed in natural language (e.g., fetching a coke when the person is thirsty) or through visual cues (e.g., fetching a mug when the table is dirty).

IV. Additional Details of Experiments

In this section, we assess the performance of our system across a variety of tabletop scenarios in the interactive scene exploration setting. Our primary objective is to address two key questions through experiments: 1) How does our system effectively and efficiently deal with diverse exploration scenarios and successfully construct comprehensive ACSG? 2) What is the utility of our ACSG in facilitating downstream tasks?

A. Robot and Environment Setups

All our experiments are conducted in a real-world setting. In these scenarios, we mount one RealSense-D455 camera on the wrist of the robot arm to collect RGBD observations, with the execution of actions performed by the UFACTORY xArm 7. The end effector for our robot arm is the soft gripper. Our experimental setup encompasses a diverse range of objects, as illustrated in Fig. 2. To assess the effectiveness of our system, we devised five types of experiments, each encompassing 10 distinct settings. These settings vary in terms of object number, type, and layout, as illustrated in Fig. 7.

B. Interactive Exploration and Scene Graph Building

Baseline. We employ the pure GPT-4V as our baseline model along with the chain-of-thoughts (CoT) to enhance its capabilities, as outlined in a method similar to that proposed by Hu et al. [116]. This baseline operates in a closed-loop fashion, receiving three RGB observations from different viewpoints during each iteration. At each turn, it generates the current scene graph, encompassing hidden objects, and suggests the next action to be taken. Upon determining that all tasks are completed, the model outputs "Done" (refer to the complete prompts in the Appendix). To ensure the baseline is robust, we utilize manual actions as ground truth references

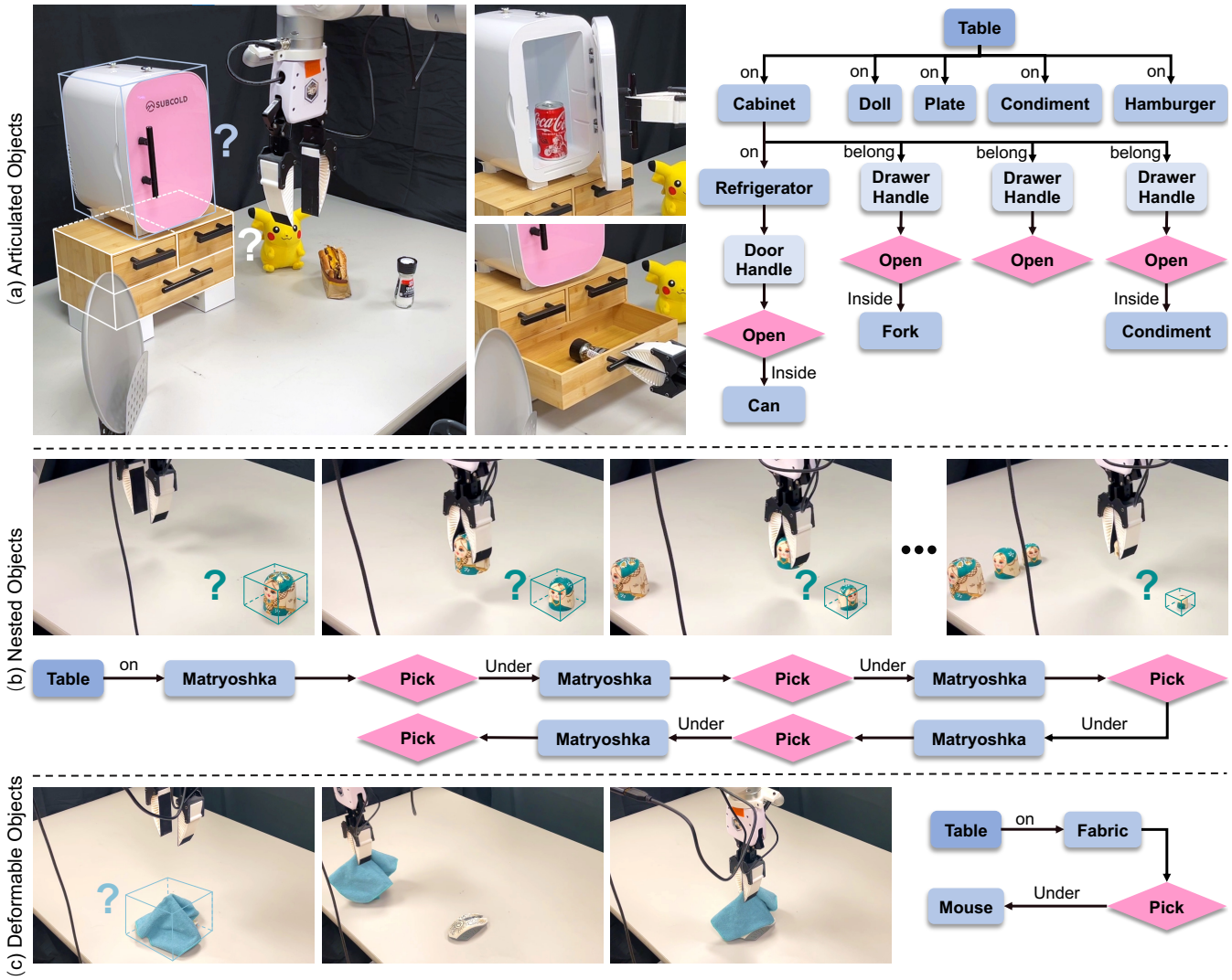


Fig. 5: **Qualitative Results on Different Scenarios.** We visualize the interactive exploration process and the corresponding constructed ACSG. (a) This scenario involves a tabletop environment with two articulated objects, accompanied by additional items either on the table or concealed in storage space. The constructed scene graph demonstrates the success of our system in identifying all objects within the environment through a series of physical interactions. (b) This scenario includes nested objects, five Matryoshka dolls, with only the top one being directly observable. Our system autonomously decides to explore the contents through a recursive reasoning process, showcasing its ability to construct deep ACSG. (c) This scenario involves a fabric covering a mouse, showcasing exploration scenarios that involve a deformable object. Our system interacts with the fabric and successfully uncovers what lies beneath it.

for the proposed actions. For instance, if the baseline suggests opening a specific drawer, we manually perform the action and prompt the model with the new observation to generate another action. In contrast, in the exploration experiments described below, all actions from our system are automatically executed by our action module on the physical robot. The full prompt of the GPT-4V baseline is illustrated in Fig. 3.

Evaluation. As mentioned in the main paper, we have designed five key metrics. To assess the effectiveness and efficiency of ACSG, we engage human evaluators in the tasks to construct the ground truth version of ACSG. The five main metrics employed for evaluation are as follows:

1) **Success:** This metric evaluates the success percentage across 10 variants for each task. We define success for each experiment as 1 when the final outputted ACSG exactly matches the GT version, and 0 otherwise.

2) **Object Recovery:** This metric quantifies the percentage of hidden objects successfully identified.

3) **State Recovery:** A binary value indicates whether the final state resembles the original state before exploration. This includes considerations for partial states and object positions (e.g., in the top drawer of a cabinet or on the table).

4) **Unexplored Space:** Evaluating the percentage of successfully explored need-to-explore space to reduce the robot's uncertainty about the scene. The identification of the need-to-explore space relies on human annotation.

5) **Graph Edit Distance (GED):** GED measures the disparity between the outputted graph and the GT graph. We adopt a simplified version of GED with six operations—three for nodes (add, delete, edit) and three for edges (add, delete, edit), with each operation incurring a cost of 1.

These metrics provide a comprehensive evaluation of the

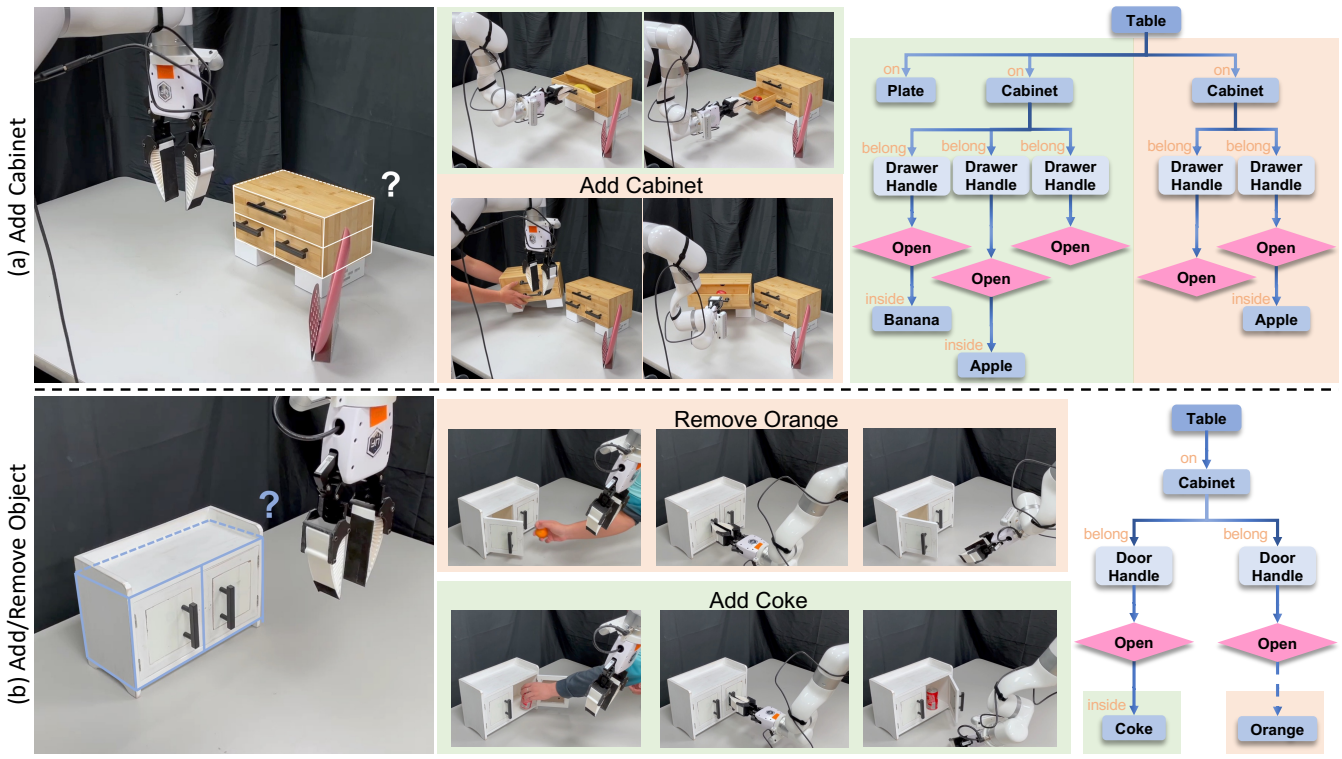


Fig. 6: **Qualitative Results on Different Intervention Scenarios.** (a) This scenario involves adding a cabinet to the tabletop setting, and our system can auto-detect the new cabinet and explore the objects inside. (b) This scenario includes removing and adding objects from and into the cabinet. Our system can monitor hand interactions and re-explore the corresponding doors.

method's performance. Additionally, we visualize the number of objects and actions during the exploration process to show the exploration strategies employed by different methods.

Comparison. Tab. I shows the quantitative results of our system and the strong baseline. The quantitative findings underscore the superior performance of our system compared to the baseline method across all metrics. It is essential to highlight that in the case of object recovery, the baseline method may occasionally choose to randomly open certain drawers or doors to unveil objects. This randomness contributes to a seemingly higher object recovery rate for the baseline, which may not necessarily correlate with its overall success. The unexplored space metric shows that our system is much more stable in exploring all need-to-explore spaces.

Fig. 4a provides additional insights, illustrating that as the number of actions increases, so does the number of objects. Specifically, we present the ground truth object number alongside the directly observable object number that can be represented by the traditional 3D scene graph. Fig. 5 shows the effectiveness of our system in different scenarios.

Human Intervention Our RoboEXP system possesses the capability to autonomously adapt to changes in the environment. We employ two types of human interventions to demonstrate these points.

The first type of intervention (Fig. 6a) involves adding new cabinets to the scene. In this scenario, we add a cabinet to the explored area, allowing our system to automatically explore the newly added cabinets and update the ACSG.

The second type of intervention (Fig. 6b) involves adding new objects to or removing existing ones from the cabinets in the current scene. Our system can monitor human interactions and discern which objects require re-exploration. Subsequently, it autonomously updates the ACSG based on re-exploration.

V. Video Timeline

See our video in https://www.youtube.com/watch?v=xZ1gfLRXSOE.

Scenario A. Exploration-Exploitation
Exploration: 00:43 - 01:16
Exploitation: 01:17 - 01:37
Scenario B. Recursive Reasoning
Exploration: 01:49 - 02:26 (Two scenarios)
Scenario C. Obstruction
Exploration: 02:33 - 02:59
Scenario D. Intervention
Exploration: 03:05 - 04:09 (Two scenarios)

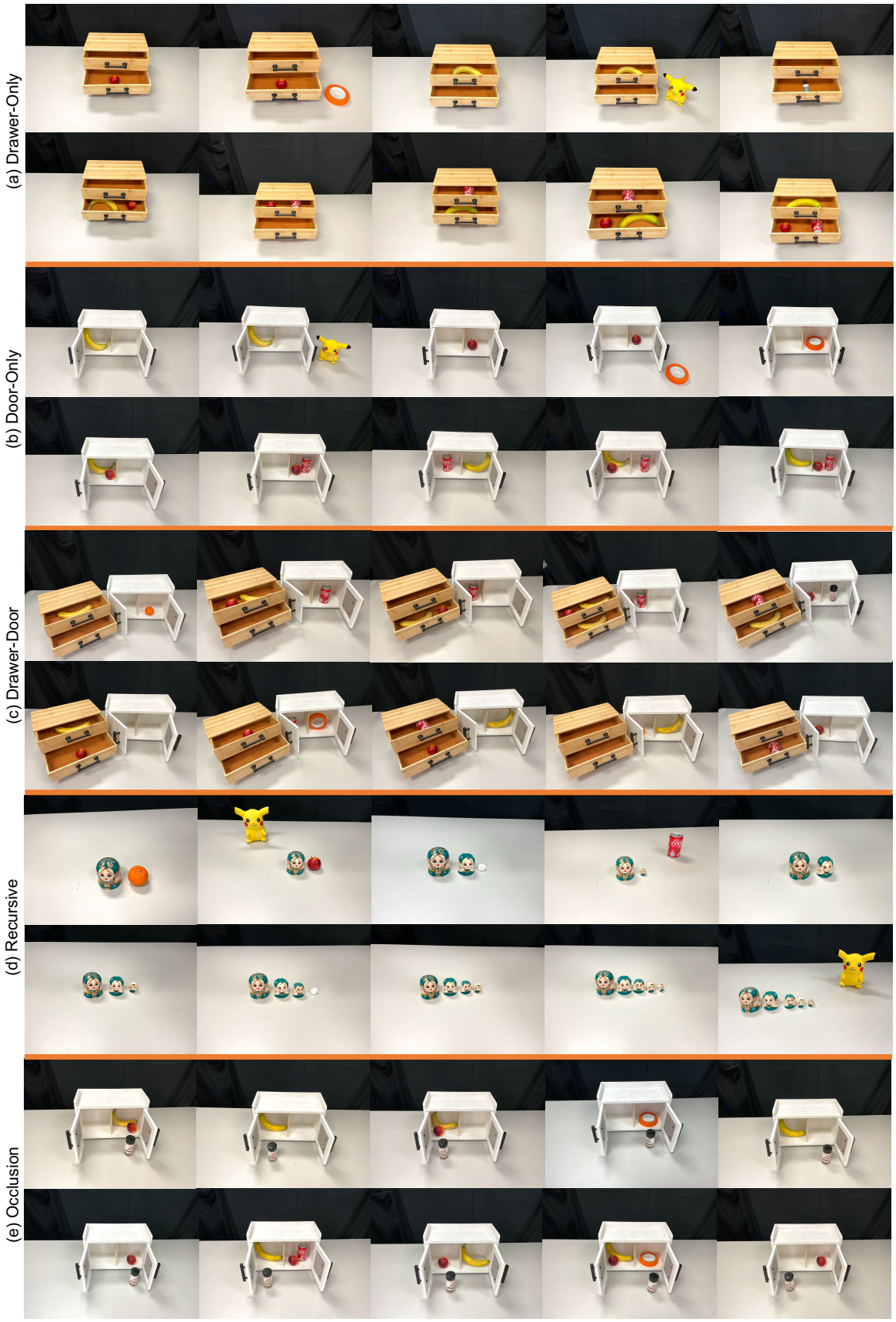


Fig. 7: **Experiment Settings.** Varied object numbers, types, and layouts in our experimental settings of the quantitative results.

REFERENCES

[1] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015. 1

[2] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017. 1

[3] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. 1

[4] Ruihai Wu, Kehan Xu, Chenchen Liu, Nan Zhuang, and Yadong Mu. Localize, assemble, and predicate: Contextual object proposal embedding for visual relation detection. In *AAAI*, 2020.

[5] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017. 1

[6] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018. 1

[7] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *ICCV*, 2019. 1

[8] Jared Strader, Nathan Hughes, William Chen, Alberto Speranzon, and Luca Carlone. Indoor and outdoor 3d scene graph generation via language-enabled spatial ontologies. *arXiv preprint arXiv:2312.11713*, 2023. 1

[9] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, 2019. 1

[10] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*, 2017. 1

[11] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 1

[12] Marcel Hildebrandt, Hang Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. Scene graph reasoning for visual question answering. *arXiv preprint arXiv:2007.01072*, 2020. 1

[13] Zachary Ravichandran, Lisa Peng, Nathan Hughes, J Daniel Griffith, and Luca Carlone. Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks. In *ICRA*, 2022. 1

[14] Zachary Seymour, Niluthpol Chowdhury Mithun, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Graphmapper: Efficient visual navigation by scene graph generation. In *ICPR*, 2022. 1

[15] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. *arXiv preprint arXiv:2307.06135*, 2023. 1

[16] Christopher Agia, Krishna Murthy Jatavallabhula, Mohamed Khodeir, Ondrej Miksik, Vibhav Vineet, Mustafa Mukadam, Liam Paull, and Florian Shkurti. Taskography: Evaluating robot task planning over large 3d scene graphs. In *CoRL*, 2022.

[17] Ziyuan Jiao, Yida Niu, Zeyu Zhang, Song-Chun Zhu, Yixin Zhu, and Hangxin Liu. Sequential manipulation planning on scene graph. In *IROS*, 2022. 1

[18] Jiayuan Mao, Tomás Lozano-Pérez, Joshua B Tenenbaum, and Leslie Pack Kaelbling. Learning reusable manipulation strategies. In *CoRL*, 2023. 1

[19] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019. 1

[20] Renhao Wang, Jiayuan Mao, Joy Hsu, Hang Zhao, Jiajun Wu, and Yang Gao. Programmatically grounded, compositionally generalizable robotic manipulation. *ICLR*, 2023. 1

[21] Zhutian Yang, Jiayuan Mao, Yilun Du, Jiajun Wu, Joshua B. Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Compositional Diffusion-Based Continuous Constraint Solvers. In *CoRL*, 2023. 1

[22] Weiyu Liu, Jiayuan Mao, Joy Hsu, Tucker Hermans, Animesh Garg, and Jiajun Wu. Composable part-based manipulation. In *CoRL*, 2023.

[23] Zhenfang Chen, Kexin Yi, Yunzhu Li, Mingyu Ding, Antonio Torralba, Joshua B Tenenbaum, and Chuang Gan. Comphy: Compositional physical reasoning of objects and events from videos. In *ICLR*, 2022.

[24] Jiayuan Mao, Tomas Lozano-Perez, Joshua B. Tenenbaum, and Leslie Pack Kaelbing. PDSketch: Integrated Domain Programming, Learning, and Planning. In *NeurIPS*, 2022. 1

[25] Farzad Niroui, Kaicheng Zhang, Zendai Kashino, and Goldie Nejat. Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. *RA-L*, 2019. 1

[26] Yugang Liu and Goldie Nejat. Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, 2013.

[27] Farzad Niroui, Ben Sprenger, and Goldie Nejat. Robot exploration in unknown cluttered environments when dealing with uncertainty. In *IRIS*, 2017. 1

[28] Barzin Doroodgar, Yugang Liu, and Goldie Nejat. A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims. *IEEE Transactions on Cybernetics*, 2014.

[29] Nicola Basilico and Francesco Amigoni. Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. *Autonomous Robots*, 2011.

[30] Yongguo Mei, Yung-Hsiang Lu, CS George Lee, and Y Charlie Hu. Energy-efficient mobile robot exploration. In *ICRA*, 2006.

[31] Stefan Oßwald, Maren Bennewitz, Wolfram Burgard, and Cyrill Stachniss. Speeding-up robot exploration by exploiting background information. *RA-L*, 2016. 1

[32] Matej Petrlik, Pavel Petracek, Vit Kratky, Tomas Musil, Yurii Stasinchuk, Matous Vrba, Tomas Baca, Daniel Hert, Martin Pecka, Tomas Svoboda, et al. Uavs beneath the surface: Cooperative autonomy for subterranean search and rescue in darpa subt. *arXiv preprint arXiv:2206.08185*, 2022.

[33] Marco Tranzatto, Takahiro Miki, Mihir Dharmadhikari, Lukas Bernreiter, Mihir Kulkarni, Frank Mascarich, Olov Andersson, Shehryar Khattak, Marco Hutter, Roland Siegwart, et al. Cerberus in the darpa subterranean challenge. *Science Robotics*, 2022. 1

[34] Philip Arm, Gabriel Waibel, Jan Preisig, Turcan Tuna, Ruyi Zhou, Valentin Bickel, Gabriela Ligeza, Takahiro Miki, Florian Kehl, Hendrik Kolvenbach, et al. Scientific exploration of challenging planetary analog environments with a team of legged robots. *Science robotics*, 2023. 1

[35] Martin J Schuster, Marcus G Müller, Sebastian G Brunner, Hannah Lehner, Peter Lehner, Ryo Sakagami, Andreas Dömel, Lukas Meyer, Bernhard Vodermayer, Riccardo Giubilato, et al. The arches space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration. *RA-L*, 2020.

[36] Florian Cordes, Ingo Ahrns, Sebastian Bartsch, Timo Birnschein, Alexander Dettmann, Stéphane Estable, Stefan Haase, Jens Hilljegerdes, David Koebel, Steffen Planthaber, et al. Lunares: Lunar crater exploration with heterogeneous multi robot systems. *Intelligent Service Robotics*, 2011.

[37] Takahiro Sasaki, Kyohei Otsu, Rohan Thakker, Sofie Haesaert, and Ali-akbar Agha-mohammadi. Where to map? iterative rover-copter path planning for mars exploration. *RA-L*, 2020. 1

[38] KAI-QING Zhou, Kai Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, L. Getoor, and X. Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. *ICML*, 2023. 1

[39] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *CVPR*, 2022.

[40] Albert J Zhai and Shenlong Wang. Peanut: predicting and navigating to unseen targets. In *ICCV*, 2023.

[41] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018.

[42] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. Thda: Treasure hunt data augmentation for semantic navigation. In *ICCV*, 2021.

[43] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *CVPR*, 2022.

[44] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation. *arXiv preprint arXiv:2204.13226*, 2022.

[45] Heming Du, Xin Yu, and Liang Zheng. Learning object relation graph and tentative policy for visual navigation. In *ECCV*, 2020.

[46] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectgoal navigation. In *ICCV*, 2021.

[47] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. *NeurIPS*, 2020.

[48] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *NeurIPS*, 2022.

[49] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *RA-L*, 2020.

[50] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019.

[51] Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to map for active semantic goal navigation. *arXiv preprint arXiv:2106.15648*, 2021.

[52] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

[53] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.

[54] Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A strong baseline for indoor object navigation. In *IROS*, 2022. 1

[55] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *CVPR*, 2022.

[56] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. *arXiv preprint arXiv:2312.03275*, 2023.

[57] Yinpei Dai, Run Peng, Sikai Li, and Joyce Chai. Think, act, and ask: Open-world interactive personalized robot navigation. *arXiv preprint arXiv:2310.07968*, 2023. 1

[58] Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. Mapping instructions to actions in 3d environments with visual goal prediction. *arXiv preprint arXiv:1809.00786*, 2018. 1

[59] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for

interpreting grounded instructions for everyday tasks. In *CVPR*, 2020.

[60] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *CVPR*, 2021.

[61] 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.

[62] Fei Xia, Chengshu Li, Roberto Martín-Martín, Or Litany, Alexander Toshev, and Silvio Savarese. Relmogen: Leveraging motion generation in reinforcement learning for mobile manipulation. *arXiv preprint arXiv:2008.07792*, 2020.

[63] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *CVPR*, 2021. 1

[64] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *RSS*, 2005. 1

[65] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. Information-theoretic planning with trajectory optimization for dense 3d mapping. In *RSS*, 2015. 1

[66] Georgios Georgakis, Bernadette Bucher, Anton Arapin, Karl Schmeckpeper, Nikolai Matni, and Kostas Daniilidis. Uncertainty-driven planner for exploration and navigation. In *ICRA*, 2022. 1

[67] Christos Papachristos, Shehryar Khattak, and Kostas Alexis. Uncertainty-aware receding horizon exploration and mapping using aerial robots. In *ICRA*, 2017.

[68] Fanfei Chen, John D Martin, Yewei Huang, Jinkun Wang, and Brendan Englot. Autonomous exploration under uncertainty via deep reinforcement learning on graphs. In *IROS*, 2020. 1

[69] Tushar Nagarajan and Kristen Grauman. Learning affordance landscapes for interaction exploration in 3d environments. In *NeurIPS*, 2020. 1

[70] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *ICLR*, 2019.

[71] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

[72] Simone Parisi, Victoria Dean, Deepak Pathak, and Abhinav Gupta. Interesting object, curious agent: Learning task-agnostic exploration. In *NeurIPS*, 2021. 1

[73] C Cao, H Zhu, Z Ren, H Choset, and J Zhang. Representation granularity enables time-efficient autonomous exploration in large, complex worlds. *Science Robotics*, 2023. 1

[74] Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *NeurIPS*, 2016. 1

[75] Lerrel Pinto and Abhinav Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *ICRA*, 2017.

[76] Tim Schneider, Boris Belousov, Georgia Chalvatzaki, Diego Romeres, Devesh K Jha, and Jan Peters. Active exploration for robotic manipulation. In *IROS*, 2022.

[77] Neil Nie, Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Structure from action: Learning interactions for articulated object 3d structure discovery. *arXiv preprint arXiv:2207.08997*, 2022. 1

[78] Cheng-Chun Hsu, Zhenyu Jiang, and Yuke Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. *arXiv preprint arXiv:2302.01295*, 2023.

[79] Linghao Chen, Yunzhou Song, Hujun Bao, and Xiaowei Zhou. Perceiving unseen 3d objects by poking the objects. In *ICRA*, 2023. 1

[80] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 1988. 1

[81] Active perception vs. passive perception. In *Proc. of IEEE Workshop on Computer Vision*, 1985. 1

[82] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon" next-best-view" planner for 3d exploration. In *ICRA*, 2016. 1

[83] Ana Batinovic, Antun Ivanovic, Tamara Petrovic, and Stjepan Bogdan. A shadowcasting-based next-best-view planner for autonomous 3d exploration. *RA-L*, 2022.

[84] Menaka Naazare, Francisco Garcia Rosas, and Dirk Schulz. Online next-best-view planner for 3d-exploration and inspection with a mobile manipulator robot. *RA-L*, 2022. 1

[85] Shengyong Chen, Youfu F Li, Wanliang Wang, and Jianwei Zhang. *Active sensor planning for multiview vision tasks*. 2008. 1

[86] Peihao Chen, Dongyu Ji, Kunyang Lin, Weiwen Hu, Wenbing Huang, Thomas Li, Mingkui Tan, and Chuang Gan. Learning active camera for multi-object navigation. *NeurIPS*, 2022. 1

[87] Mahsa Ghasemi, Erdem Bulgur, and Ufuk Topcu. Task-oriented active perception and planning in environments with partially known semantics. In *ICML*, 2020. 1

[88] Yian Wang, Ruihai Wu, Kaichun Mo, Jiaqi Ke, Qingnan Fan, Leonidas Guibas, and Hao Dong. AdaAfford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions. In *ECCV*, 2022. 1

[89] Roberto Martín-Martín and Oliver Brock. Building kinematic and dynamic models of articulated objects with multi-modal interactive perception. In *2017 AAAI Spring Symposium Series*, 2017. 1

[90] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from

interaction. In *CVPR*, 2022. 1

[91] Christopher Collander, William J Beksi, and Manfred Huber. Learning the next best view for 3d point clouds via topological features. In *ICRA*, 2021. 1

[92] Daryl Peralta, Joel Casimiro, Aldrin Michael Nilles, Justine Aletta Aguilar, Rowel Atienza, and Rhandley Cajote. Next-best view policy for 3d reconstruction. In *ECCV Workshops*. Springer, 2020. 1

[93] Muzhi Han, Zeyu Zhang, Ziyuan Jiao, Xu Xie, Yixin Zhu, Song-Chun Zhu, and Hangxin Liu. Reconstructing interactive 3d scenes by panoptic mapping and cad model alignments. In *ICRA*, 2021. 1

[94] Muzhi Han, Zeyu Zhang, Ziyuan Jiao, Xu Xie, Yixin Zhu, Song-Chun Zhu, and Hangxin Liu. Scene reconstruction with functional objects for robot autonomy. *IJCV*, 2022. 1

[95] Zhirong Wu, Shuran Song, Aditya Khosla, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets for 2.5 d object recognition and next-best-view prediction. *arXiv preprint arXiv:1406.5670*, 2014. 1

[96] Yiheng Han, Irvin Haozhe Zhan, Wang Zhao, and Yong-Jin Liu. A double branch next-best-view network and novel robot system for active object reconstruction. In *ICRA*, 2022.

[97] Björn Browatzki, Vadim Tikhanoff, Giorgio Metta, Heinrich H Bülthoff, and Christian Wallraven. Active in-hand object recognition on a humanoid robot. *IEEE Transactions on Robotics*, 2014. 1

[98] Qihang Fang, Yingda Yin, Qingnan Fan, Fei Xia, Siyan Dong, Sheng Wang, Jue Wang, Leonidas Guibas, and Baoquan Chen. Towards accurate active camera localization. In *ECCV*, 2022. 1

[99] Jun Lv, Yunhai Feng, Cheng Zhang, Shuang Zhao, Lin Shao, and Cewu Lu. Sam-rl: Sensing-aware model-based reinforcement learning via differentiable physics-based simulation and rendering. *RSS*, 2023. 1

[100] Youssef Zaky, Gaurav Paruthi, Bryan Tripp, and James Bergstra. Active perception and representation for robotic manipulation. *arXiv preprint arXiv:2003.06734*, 2020. 1

[101] Quoc V Le, Ashutosh Saxena, and Andrew Y Ng. Active perception: Interactive manipulation for improving object detection. *Standford University Journal*, 2008. 1

[102] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *CVPR*, 2018. 1

[103] Snehal Jauhri, Sophie Lueth, and Georgia Chalvatzaki. Active-perceptive motion generation for mobile manipulation. *arXiv preprint arXiv:2310.00433*, 2023. 1

[104] Steven D Whitehead and Dana H Ballard. Active perception and reinforcement learning. In *Machine Learning Proceedings 1990*. 1990. 1

[105] John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, et al. Chatgpt: Optimizing language models for dialogue. *OpenAI blog*, 2022. 1

[106] R OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[107] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023. 1

[108] OpenAI. Gpt-4v(ision) system card. *https://cdn.openai.com/papers/GPTV System Card.pdf*, 2023. 1, 4

[109] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v(ision). *arXiv preprint arXiv: 2309.17421*, 2023. 1, 4

[110] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 1

[111] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022. 1

[112] Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. *arXiv preprint arXiv:2309.12311*, 2023. 1

[113] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *ICRA*, 2023. 1

[114] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 5

[115] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. In *CoRL*, 2023. 1

[116] Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv: 2311.17842*, 2023. 1, 5, 6

[117] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying

dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 4

[118] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *arXiv preprint arXiv: 2306.01567*, 2023. 4

[119] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 4

[120] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *ICML*, 2021. 4

[121] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023. 4

[122] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *CoRL*, 2023. 4

[123] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv preprint arXiv: 2309.16650*, 2023. 6