CuriousBot: Interactive Mobile Exploration via Actionable 3D Relational Object Graph

Yixuan Wang[1,2], Leonor Fermoselle[2], Tarik Kelestemur[2], Jiuguang Wang[2], Yunzhu Li[1]

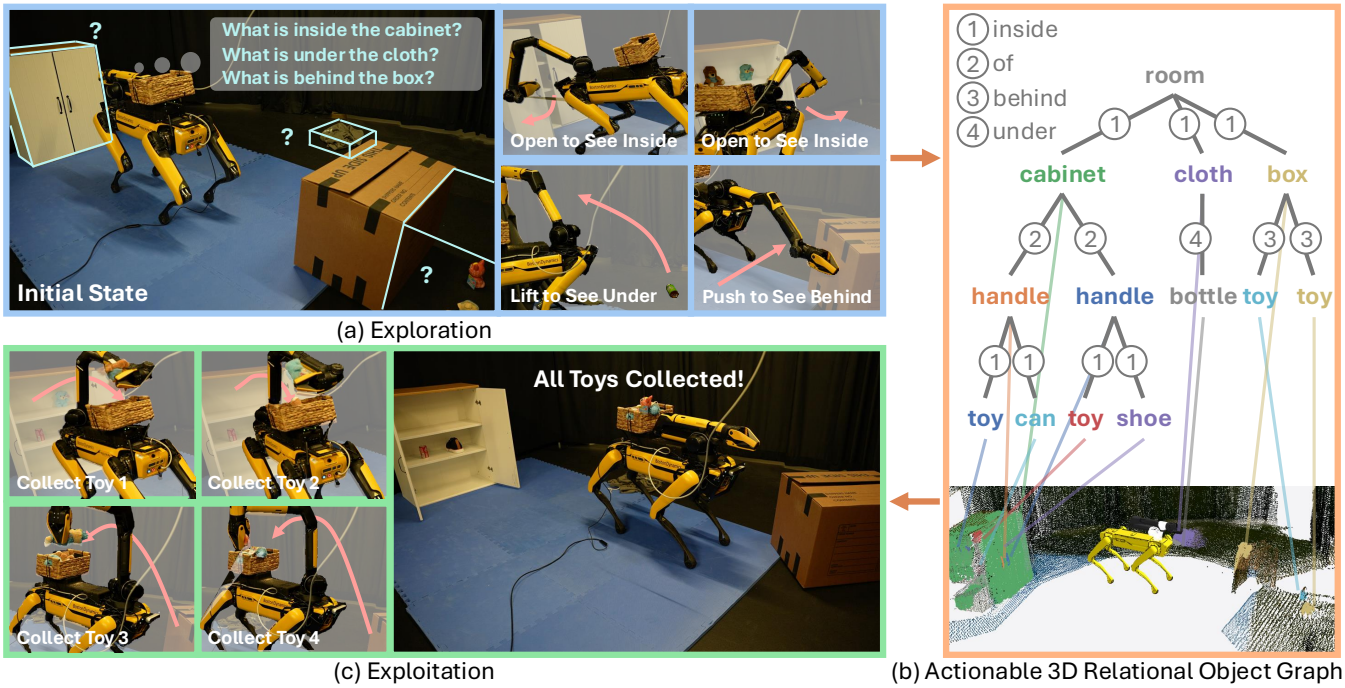


Fig. 1: **CuriousBot.** We present a mobile robotic system that can (a) interactively explore the environment, such as inspecting hidden spaces inside a cabinet or behind a box, (b) construct an actionable 3D relational object graph that encodes both the semantic and geometric information of object nodes, along with various object relationships, and (c) perform manipulation tasks by retrieving objects through traversal of the actionable 3D relational object graph.

***Abstract*— Mobile exploration is a longstanding challenge in robotics, yet current methods primarily focus on active perception instead of active interaction, limiting the robot's ability to interact with and fully explore its environment. Existing robotic exploration approaches via active interaction are often restricted to tabletop scenes, neglecting the unique challenges posed by mobile exploration, such as large exploration spaces, complex action spaces, and diverse object relations. In this work, we introduce a 3D relational object graph that encodes diverse object relations and enables exploration through active interaction. We develop a system based on this representation and evaluate it across diverse scenes. Our qualitative and quantitative results demonstrate the system's effectiveness and generalization capabilities, outperforming methods that rely solely on vision-language models (VLMs).**

I. INTRODUCTION

Exploration remains a significant challenge for mobile robots, especially in complex household environments filled with occlusions, such as objects concealed within cabinets, hidden under furniture, or obscured behind other obstacles. Traditional exploration methods primarily focus on *active perception* [1, 2], aiming to determine the optimal camera position to minimize unknown spaces, and often neglect the crucial aspect of *active interaction*, which involves deciding where and how to physically interact with the environment to reveal hidden spaces. While recent works like RoboEXP [3] have considered active interaction, their focus is primarily on tabletop manipulation, limiting their applicability in complex, real-world mobile settings.

In contrast to tabletop scenarios, mobile exploration in real-world environments introduces unique challenges:

- **Expanded exploration space**: the exploration area for mobile robots is substantially larger and needs to utilize complex navigation and mapping skills.
- **Complex occlusion relationships**: occlusions in household environments are intricate. While RoboEXP considers basic relationships like `on`, `belong`, `under`, `behind`, and `inside`, real-world settings present complex occlusions, such as items hidden beneath furniture or blocked by other objects, requiring more sophisticated reasoning and interaction strategies.
- **Larger action space**: mobile exploration involves a broader action space that includes both navigation and

https://bdaiinstitute.github.io/curiousbot/

[1]Columbia University [2]Boston Dynamics AI Institute

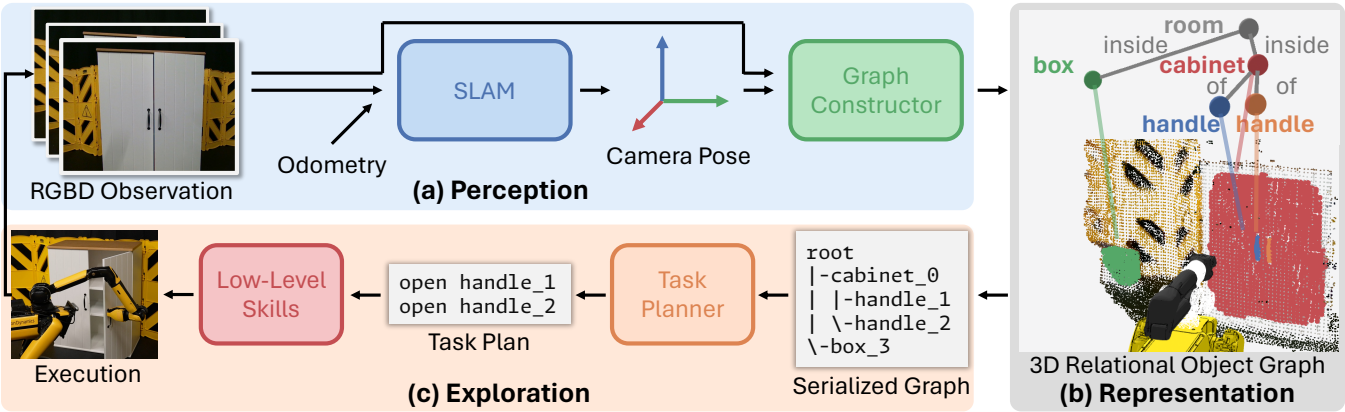


Fig. 2: **Method Overview.** (a) In the perception pipeline, SLAM processes RGBD observations and odometry estimation from the robot to output camera poses, which are used alongside the RGBD observations to construct an actionable 3D relational object graph. (b) The 3D relational object graph comprises object nodes containing both geometric and semantic information, as well as object edges that encode complex object relations. (c) The serialized object graph is fed into the task planner, and the generated task plans are executed using low-level skills to interactively explore the environment.

manipulation to handle various objects and scenes.

In this work, we tackle the challenges of active mobile exploration using our 3D relational object graph powered by Visual Foundational Models (VFMs). Our system consists of four modules - **SLAM**, **Graph Constructor**, **Task Planner**, and **Low-Level Skills**, as shown in Figure 2.

The SLAM module takes in a sequence of RGBD observations and robot odometry, and outputs the camera pose. Given observations and camera poses, our graph constructor first builds object nodes by detecting and segmenting objects via the open-vocabulary object detector and Segment Anything [4, 5]. By leveraging spatial and semantic information, we determine the relationships between nodes, which are then used for downstream task planning. The task planning module takes in the serialized object graph and generates action plans using a Large Language Model (LLM). Finally, low-level skills, consisting of several action primitives, execute the generated action plan.

We evaluate our system in various scenes requiring exploration and demonstrate its capability to handle a wide range of object categories, including articulated, deformable, and rigid objects. Furthermore, our 3D relational object graph can encode multiple occlusion relationships commonly seen in household environments, such as `of`, `on`, `under`, `behind`, and `inside`. The system can adapt to different environment layouts, such as a box-filled room or a living room. We quantitatively analyze our system by evaluating it across five tasks, each repeated ten times under different initial conditions, and identify common failure patterns. Additionally, we compare our method with the direct use of GPT-4V to guide robot exploration. Our findings indicate that our 3D relational object graph is more effective for task planning.

In summary, our contributions are threefold: i) We introduce the 3D relational object graph, which can encode a number of common object relations, enabling the mobile robot to explore diverse everyday environments. ii) We develop the CuriousBot system, which can automatically construct the 3D object graph, plan exploration, and interact with the environment to reduce unknown spaces. iii) We conduct comprehensive experiments, demonstrating that our system can fully explore environments and accurately build the object graph. The testing scenes feature diverse object categories, object relations, and scene layouts. Additionally, we provide deeper insights into our system through error breakdown and comparisons with baseline methods.

II. METHOD

As shown in Figure 2, our framework consists of four modules - SLAM, Graph Constructor, Task Planner, and Low-Level Skills, each of which will be explained in detail in the following sections.

A. Problem Statement

We define the object graph as $G = (V, E)$, where $V = \{v^0, v^1, \ldots, v^N\}$ represents the set of object nodes, and $E = \{e^0, e^1, \ldots, e^M\}$ represents the set of edges. Each node v contains both semantic attributes, such as object labels, and geometric attributes, such as point clouds and normal estimations. Each edge e represents a directed connection from node v^i to node v^j, along with their object relationship. The objective of mobile exploration is to construct a graph G that can minimize the unknown space, discover as many object nodes as possible, and establish correct object relationships.

B. SLAM

SLAM takes in a sequence of odometry estimation from robot and RGBD observations, $\mathbf{O}_{0\ldots t}$, where each $\mathbf{O}_t \in \mathbb{R}^{H\times W\times 4}$ represents one RGBD frame, and simultaneously localizes the camera pose $\mathbf{T}_t$ and constructs the map $\mathbf{M}_t$ denotes the map at time t. In practice, we use RTAB-Map for SLAM to estimate the camera pose [6].

C. Graph Constructor

Given the current RGBD observation $\mathbf{O}_t$, the corresponding camera pose $\mathbf{T}_t$, and the graph from the previous frame G_{t-1}, we construct the graph G_t at time t. We first segment the objects using YOLO-World and SAM and obtain corresponding 3D point clouds [4, 5]. Next, we associate the segmented objects with previous object nodes based on

geometric information and fuse the current observation to obtain the current object nodes. Finally, we establish object relationships by jointly considering geometric, semantic, and action-related information.

D. Task Planner

We input the serialized object graph into the LLM to plan skills. For serialization, we perform a depth-first search over the object graph and serialize it based on the object label and index. Specifically, we start at the root node and find all child nodes connected to it. We then put these nodes into a stack, which stores the nodes we plan to visit. Next, we pop the top node from the stack and check its children. If it has no children, it is a leaf node; otherwise, we repeat this operation until the stack is empty. Additionally, we keep track of information such as depth, object name, node ID, and related action details to build the serialized graph. Additionally, we provide the LLM with several prompts for planning. Our appendix provides more details regarding the method, including search algorithm details and prompts.

E. Low-Level Skills

In our work, we implement several primitive skills, including opening, lifting, pushing, collecting objects, sitting, and flipping. The skill output by the task planner consists of the skill name from our skill library and the target object index. Given this skill information, we execute the corresponding skill to explore the environment.

Specifically, we construct the following manipulation skills using heuristics, including `open`, `flip`, `lift`, `push`, `sit`, and `collect`, with more details in the appendix.

III. EXPERIMENT

In our experiments, we aim to answer the following questions: (1) What kinds of tasks can be enabled by our system, and what scenarios can our robot explore? (2) How does each component perform, and what are the common failure patterns? (3) How will the whole system perform with different design choices?

A. Experiment Setup

We conduct experiments using the Boston Dynamics Spot as the mobile manipulator. An additional RealSense 455 camera is installed at the front to enhance environmental observation, as shown in Figure 5. For computation, we use a desktop equipped with an Nvidia RTX A6000 GPU and an AMD CPU with 128GB of memory. Our system is evaluated on diverse daily objects, as shown in Figure 5. We set up the environment in a 3m $\times$ 4m room.

B. Mobile Exploration in Various Scenes

We qualitatively evaluate our system on diverse scenes, as shown in Figure 3 and Figure 6. We would like to highlight the following aspects of our system's capabilities:

Diverse Object Categories. Our system operates in scenarios containing various types of objects, such as articulated objects, deformable objects, and rigid objects, demonstrating its generalization capabilities across different object types.

Various Object Relations. Our system encodes five types of object relations commonly observed in the real world, including `behind`, `of`, `inside`, `on`, and `under`. For instance, the robot understands that there is unknown space behind the chair, requiring it to push the chair away to reveal the space behind, as shown in Figure 3.

Different Layouts. The test scenarios include scenes of varying scales and layouts, from small setups like piles of cloth to larger household environments, which demonstrate that our system can generalize to various scenes.

Diverse Interactions. The robot interacts with the environment and explores the scene in multiple ways. For example, the robot can grasp an object rigidly, such as opening a cabinet shown in Figure 3. It can also interact with objects nonprehensilely using its arm instead of the gripper, such as pushing the chair shown in Figure 3. Additionally, the robot can actively move the camera around without manipulating objects, such as sitting down to check the space under a table, as shown in Figure 6.

C. Failure Breakdown

We conduct experiments involving tasks such as flipping boxes, opening drawers, checking underneath objects, pushing boxes, and lifting cloth. Each task is repeated ten times under different initial conditions, with the success rate recorded, and the failure breakdown is shown in Figure 4. A rollout is considered successful if the robot successfully completes all exploration skills.

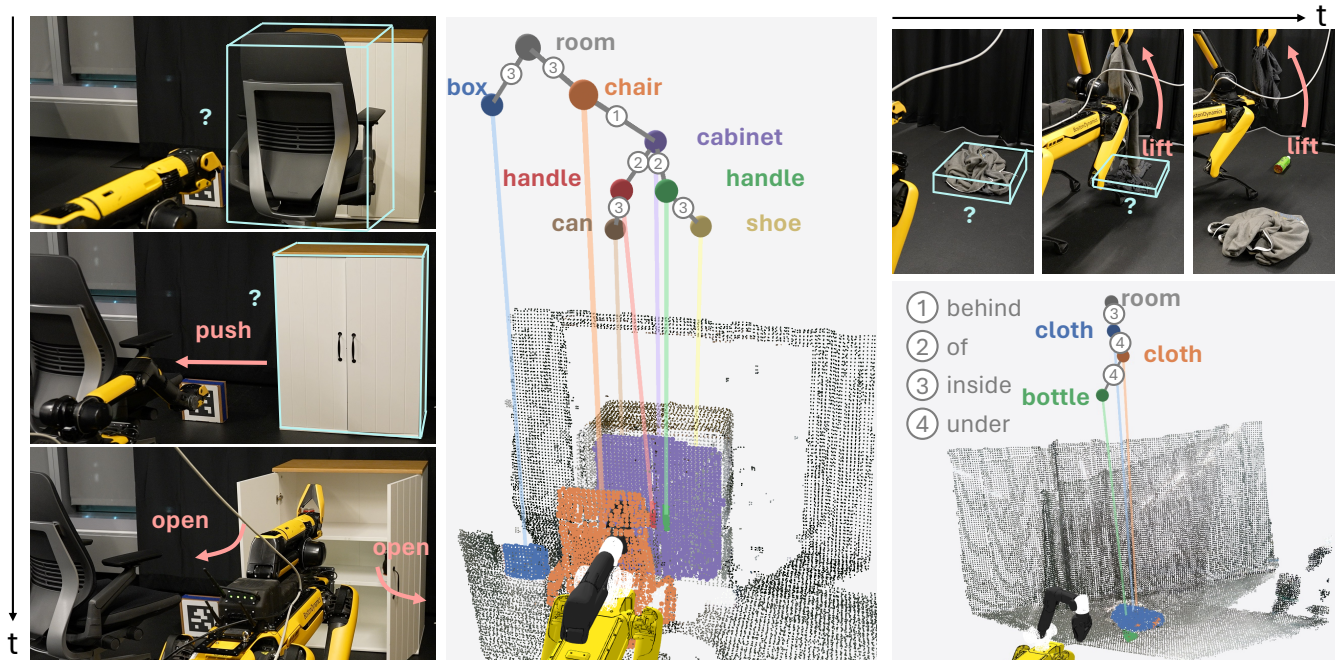
The overall success rate is 82%. For the failure cases, we categorize the primary reasons into perception failure, decision failure, and action failure. Perception failure occurs when inaccurate perception results lead to unreasonable plans in downstream task planning or incorrect action execution. Decision failure happens when, despite having a correct serialized graph, the task planner makes an incorrect decision. Action failure occurs when, despite having a correct task plan and an accurate object graph, the skill execution fails.

In cases of perception failure, two major causes are an inaccurate object graph due to imprecise SLAM and errors from the open-vocabulary object detector. For decision failure, the task planner can fail in predicting the correct skills for the corresponding object nodes. Regarding action failures, we highlight the complexity of real-world manipulation, such as the early release of the gripper, loose grasping, and unexpected interference between the robot and the object.

D. Comparisons with Baselines

To study the effectiveness of our method, we compare our system with the following baselines on the same five tasks as in Section III-C:

- **LLaVa:** We directly feed the current RGB observation and the same text prompt as our LLM task planner into LLaVa, the state-of-the-art open-source Vision Language Model (VLM) [7, 8]. Because VLM does not equip manipulation skills, a human operator will help VLM finish manipulation.



(i) Pushing Away Chair (ii) Lift Cloth

Fig. 3: **Diverse Scenes and Skills.** We evaluate our system's exploration capabilities across various tasks, including pushing the chair aside to reveal space behind it, lifting cloth to check underneath, flipping open boxes to inspect the contents, and exploring a household scene. These tasks showcase the system's ability to generalize across different object types, scenarios, and object relations. Additional tasks can be found on the project page.

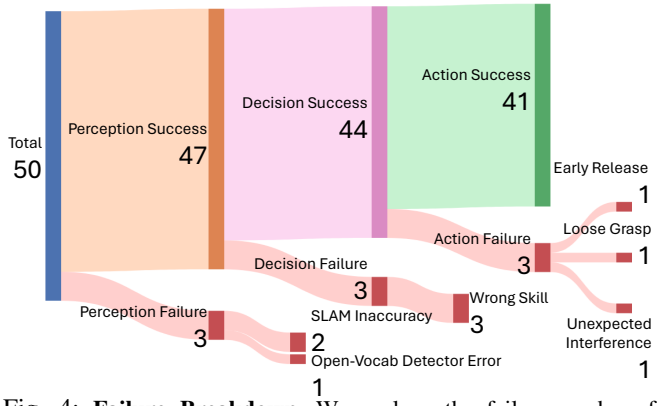


Fig. 4: **Failure Breakdown.** We analyze the failure modes of our system during exploration tasks, identifying three main causes: perception failure, decision failure, and action failure.

- **Gemini:** Similar to the LLaVa baseline, with the only change being the substitution of LLaVa with Gemini, a state-of-the-art closed-source VLM [9].
- **GPT-4o:** Similarly, we replace the VLM with GPT-4o, another state-of-the-art VLM [10].
- **Heuristics:** We implement a heuristic exploration policy where the robot will open all handles.

We evaluate performance using three metrics: 1) **Success Rate**: A rollout is considered successful if all exploration skills are correctly executed. 2) **Object Recovery (OR)**: Assuming the ground truth object nodes are V_{gt} and the discovered object nodes are V, object discovery is defined as $|V_{\text{gt}} \cap V|/|V_{\text{gt}}|$. 3) **Graph Editing Distance (GED)**: If the cost of adding, deleting, or moving one edge or node is 1, GED is defined as the total cost of editing the final graph G to match the ground truth graph G_{gt}.

Table I summarizes our quantitative results. We found our 3D relational object graph is more effective than feeding RGB observations into a VLM. This is because our representation explicitly represents the topological relationships of object nodes, leading to more effective task planning compared to requiring VLM to memorize observations and reason object relations implicitly. Additionally, our actionable object graph grounds actions within the representation, while RGB observations alone do not provide sufficient information for low-level skill selection. Additionally, while simple exploration heuristics may yield comparable performance in certain tasks, they do not generalize to other tasks.

IV. CONCLUSION

Interactive mobile exploration has been a longstanding and essential problem in robotics. However, existing approaches to mobile exploration primarily focus on active perception rather than active interaction, which limits the robot's ability to fully explore the environment. Current methods for robotic exploration via active interaction are mainly focused on tabletop scenes, overlooking the unique challenges of mobile settings, such as expansive exploration spaces, large action spaces, and diverse object relations. We introduce the 3D relational object graph, which encodes diverse object relations, and build a system capable of exploration through active interaction based on this representation. We evaluate our system in diverse scenes, demonstrating its effectiveness and generalization capabilities qualitatively. Our quantitative results further underscore its effectiveness compared to directly using VLMs.

ACKNOWLEDGEMENT

We would like to thank Jianing Qian, Chao Cao, Yun Chang, Brian Okorn, Bernadette Bucher, and Emmanuel Panov for the helpful discussions and feedback on the project. This project is supported in part by the DARPA TIAMAT program (HR0011-24-9-0430) and the Toyota Research Institute (TRI). This article solely reflects the opinions and conclusions of its authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

REFERENCES

[1] C. Cao, H. Zhu, H. Choset, and J. Zhang, "Tare: A hierarchical framework for efficiently exploring complex 3d environments." in *Robotics: Science and Systems*, vol. 5, 2021, p. 2.

[2] H. Choset, S. Walker, K. Eiamsa-Ard, and J. Burdick, "Sensor-based exploration: Incremental construction of the hierarchical generalized voronoi graph," *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 126–148, 2000.

[3] H. Jiang, B. Huang, R. Wu, Z. Li, S. Garg, H. Nayyeri, S. Wang, and Y. Li, "Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation," *arXiv preprint arXiv:2402.15487*, 2024.

[4] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024.

[5] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[6] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of field robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[7] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *NeurIPS*, 2023.

[8] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," 2023.

[9] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[10] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[11] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[12] Y. Liu and G. Nejat, "Robotic urban search and rescue: A survey from the control perspective," *Journal of Intelligent & Robotic Systems*, vol. 72, pp. 147–165, 2013.

[13] Y. Mei, Y.-H. Lu, C. G. Lee, and Y. C. Hu, "Energy-efficient mobile robot exploration," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 505–511.

[14] K. Zheng, A. Paul, and S. Tellex, "A system for generalized 3d multi-object search," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1638–1644.

[15] K. Zheng, R. Chitnis, Y. Sung, G. Konidaris, and S. Tellex, "Towards optimal correlational object search," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7313–7319.

[16] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "Esc: Exploration with soft commonsense constraints for zero-shot object navigation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 42 829–42 842.

[17] R. Ramrakhya, E. Undersander, D. Batra, and A. Das, "Habitat-web: Learning embodied object-search strategies from human demonstrations at scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5173–5183.

[18] A. J. Zhai and S. Wang, "Peanut: Predicting and navigating to unseen targets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 926–10 935.

[19] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," *arXiv preprint arXiv:1810.06543*, 2018.

[20] O. Maksymets, V. Cartillier, A. Gokaslan, E. Wijmans, W. Galuba, S. Lee, and D. Batra, "Thda: Treasure hunt data augmentation for semantic navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 374–15 383.

[21] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 829–14 838.

[22] K. Yadav, R. Ramrakhya, A. Majumdar, V.-P. Berges, S. Kuhar, D. Batra, A. Baevski, and O. Maksymets, "Offline visual representation learning for embodied navigation," in *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[23] H. Du, X. Yu, and L. Zheng, "Learning object relation graph and tentative policy for visual navigation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*. Springer, 2020, pp. 19–34.

[24] J. Ye, D. Batra, A. Das, and E. Wijmans, "Auxiliary tasks and exploration enable objectgoal navigation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 117–16 126.

[25] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa,

D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," *arXiv preprint arXiv:1911.00357*, 2019.

[26] G. Georgakis, B. Bucher, K. Schmeckpeper, S. Singh, and K. Daniilidis, "Learning to map for active semantic goal navigation," *arXiv preprint arXiv:2106.15648*, 2021.

[27] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *arXiv preprint arXiv:2004.05155*, 2020.

[28] H. Luo, A. Yue, Z.-W. Hong, and P. Agrawal, "Stubborn: A strong baseline for indoor object navigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3287–3293.

[29] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with interaction-free learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 890–18 900.

[30] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 42–48.

[31] Y. Dai, R. Peng, S. Li, and J. Chai, "Think, act, and ask: Open-world interactive personalized robot navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3296–3303.

[32] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 538–547.

[33] D. Misra, A. Bennett, V. Blukis, E. Niklasson, M. Shatkhin, and Y. Artzi, "Mapping instructions to actions in 3d environments with visual goal prediction," *arXiv preprint arXiv:1809.00786*, 2018.

[34] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi, "Visual room rearrangement," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5922–5931.

[35] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, *et al.*, "Rearrangement: A challenge for embodied ai," *arXiv preprint arXiv:2011.01975*, 2020.

[36] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Leveraging motion generation in reinforcement learning for mobile manipulation," *arXiv preprint arXiv:2008.07792*, 2020.

[37] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi, "Manipulathor: A framework for visual object manipulation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4497–4506.

[38] C. Cao, H. Zhu, F. Yang, Y. Xia, H. Choset, J. Oh, and J. Zhang, "Autonomous exploration development environment and the planning algorithms," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8921–8928.

[39] J. Yan, X. Lin, Z. Ren, S. Zhao, J. Yu, C. Cao, P. Yin, J. Zhang, and S. Scherer, "Mui-tare: Cooperative multi-agent exploration with unknown initial position," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4299–4306, 2023.

[40] C. Cao, H. Zhu, Z. Ren, H. Choset, and J. Zhang, "Representation granularity enables time-efficient autonomous exploration in large, complex worlds," *Science Robotics*, vol. 8, no. 80, p. eadf0970, 2023.

[41] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[42] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, "Evaluating the efficiency of frontier-based exploration strategies," in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE, 2010, pp. 1–8.

[43] M. Kulich, J. Faigl, and L. Přeučil, "On distance utility in the exploration task," in *2011 ieee international conference on robotics and automation*. IEEE, 2011, pp. 4455–4460.

[44] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3d," *Advanced Robotics*, vol. 27, no. 6, pp. 459–468, 2013.

[45] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys, "Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1071–1078.

[46] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2135–2142.

[47] M. Kulich, J. Kubalík, and L. Přeučil, "An integrated approach to goal selection in mobile robot exploration," *Sensors*, vol. 19, no. 6, p. 1400, 2019.

[48] J. Faigl and M. Kulich, "On determination of goal candidates in frontier-based multi-robot exploration," in *2013 European Conference on Mobile Robots*. IEEE, 2013, pp. 210–215.

[49] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of morse decompositions," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 345–366, 2002.

[50] S. Kim, S. Bhattacharya, R. Ghrist, and V. Kumar, "Topological exploration of unknown and partially known environments," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE,

2013, pp. 3851–3858.

[51] N. Savinov, A. Dosovitskiy, and V. Koltun, "Semi-parametric topological memory for navigation," *arXiv preprint arXiv:1803.00653*, 2018.

[52] T. Chen, S. Gupta, and A. Gupta, "Learning exploration policies for navigation," *arXiv preprint arXiv:1903.01959*, 2019.

[53] S. Parisi, V. Dean, D. Pathak, and A. Gupta, "Interesting object, curious agent: Learning task-agnostic exploration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 516–20 530, 2021.

[54] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International conference on machine learning*. PMLR, 2017, pp. 2778–2787.

[55] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," *arXiv preprint arXiv:1808.04355*, 2018.

[56] T. Nagarajan and K. Grauman, "Learning affordance landscapes for interaction exploration in 3d environments," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2005–2015, 2020.

[57] F. Schmalstieg, D. Honerkamp, T. Welschehold, and A. Valada, "Learning hierarchical interactive multi-object search for mobile manipulation," *IEEE Robotics and Automation Letters*, 2023.

[58] Z. Zhang, Z. Zhu, P. Li, T. Liu, X. Ma, Y. Chen, B. Jia, S. Huang, and Q. Li, "Task-oriented sequential grounding in 3d scenes," *arXiv preprint arXiv:2408.04034*, 2024.

[59] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3d scene graph construction and optimization," *arXiv preprint arXiv:2201.13360*, 2022.

[60] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time task-driven open-set 3d scene graphs," *arXiv preprint arXiv:2404.13696*, 2024.

[61] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone, "Foundations of spatial perception for robotics: Hierarchical representations and real-time systems," *The International Journal of Robotics Research*, p. 02783649241229725, 2024.

[62] Y. Chang, L. Ballotta, and L. Carlone, "D-lite: navigation-oriented compression of 3d scene graphs for multi-robot collaboration," *IEEE Robotics and Automation Letters*, 2023.

[63] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5664–5673.

[64] J. Strader, N. Hughes, W. Chen, A. Speranzon, and L. Carlone, "Indoor and outdoor 3d scene graph generation via language-enabled spatial ontologies," *IEEE Robotics and Automation Letters*, 2024.

[65] Z. Ravichandran, L. Peng, N. Hughes, J. D. Griffith, and L. Carlone, "Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9272–9279.

[66] C. Agia, K. M. Jatavallabhula, M. Khodeir, O. Miksik, V. Vineet, M. Mukadam, L. Paull, and F. Shkurti, "Taskography: Evaluating robot task planning over large 3d scene graphs," in *Conference on Robot Learning*. PMLR, 2022, pp. 46–58.

[67] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha, *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *arXiv preprint arXiv:2302.07241*, 2023.

[68] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et al.*, "Openscene: 3d scene understanding with open vocabularies," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 815–824.

[69] H. Ha and S. Song, "Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models," *arXiv preprint arXiv:2207.11514*, 2022.

[70] J. Wang, J. Tarrio, L. Agapito, P. F. Alcantarilla, and A. Vakhitov, "Semlaps:: Real-time semantic mapping with latent prior networks and quasi-planar segmentation," *IEEE Robotics and Automation Letters*, 2023.

[71] S. Koch, P. Hermosilla, N. Vaskevicius, M. Colosi, and T. Ropinski, "Lang3dsg: Language-based contrastive pre-training for 3d scene graph prediction," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 1037–1047.

[72] K. Yamazaki, T. Hanyu, K. Vo, T. Pham, M. Tran, G. Doretto, A. Nguyen, and N. Le, "Open-fusion: Real-time open-vocabulary 3d mapping and queryable scene representation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 9411–9417.

[73] H. Chang, K. Boyalakuntla, S. Lu, S. Cai, E. Jing, S. Keskar, S. Geng, A. Abbas, L. Zhou, K. Bekris, *et al.*, "Context-aware entity grounding with open-vocabulary 3d scene graphs," *arXiv preprint arXiv:2309.15940*, 2023.

[74] A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann, "Openmask3d: Open-vocabulary 3d instance segmentation," *arXiv preprint arXiv:2306.13631*, 2023.

[75] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html

[76] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[77] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.

[78] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision transformers need registers," 2023.

[79] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5021–5028.

[80] S. Chandhok, "Scenegpt: A language model for 3d scene understanding," *arXiv preprint arXiv:2408.06926*, 2024.

[81] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[82] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[83] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[84] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[85] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[86] P. Chen, X. Sun, H. Zhi, R. Zeng, T. H. Li, G. Liu, M. Tan, and C. Gan, "A2nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models," *arXiv preprint arXiv:2308.07997*, 2023.

[87] J. Yang, X. Chen, S. Qian, N. Madaan, M. Iyengar, D. F. Fouhey, and J. Chai, "Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 7694–7701.

[88] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao, "Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning," *arXiv preprint arXiv:2311.17842*, 2023.

APPENDIX

A. RELATED WORK

1) Robotic Exploration: Robotic exploration is crucial for many applications, including search and rescue [11–13], object search [14–32], and mobile manipulation [33–37]. The typical objective of exploration is to minimize the unknown areas in the environment [1, 2, 14, 15, 30, 38–52]. Recently, curiosity-driven methods have emerged as another promising approach to guide robotic exploration [53–56]. However, these methods generally focus on exploration through active perception, neglecting exploration via active interaction, which limits the robot's ability to fully explore environments, such as finding objects inside cabinets.

The work most closely related to ours is RoboEXP [3], where the robot interacts with the environment to build a complete 3D object graph of the scene. However, their focus is on tabletop scenes, while our mobile setting poses different challenges. In contrast, our approach emphasizes mobile exploration through active interaction, which introduces unique challenges such as larger exploration areas, more complex object relationships, and a broader action space.

Fabian, et. al. also explores mobile exploration via active interaction [57]. However, they only consider a subset of the diverse object relationships, which are essential for complex exploration behaviors. On the contrary, our 3D relational object graph can encode five types of object relations. Additionally, they only consider opening as a manipulation skill and rely on AR markers in the real world to guide manipulation. In contrast, we incorporate more skills, including pushing, opening, lifting, flipping, and more, without requiring additional markers.

2) 3D Scene Graph for Robotics: 3D scene graph representation is widely used in robotic manipulation and navigation [3, 58–74]. These representations often leverage 2D VFMs such as SAM, CLIP, or DINO [5, 75–78] to extract 2D visual information, which is then fused into 3D space. However, existing methods tend to focus on the semantic understanding of objects, rather than encoding complex object relations like `on`, `inside`, or `behind`. Understanding such occlusion relations is crucial for making informed decisions about where to explore and how to manipulate objects. In contrast, our representation encodes various types of occlusion relations in real-world environments, allowing the mobile robot to actively decide how to explore the environment. Although works like ConceptGraph and SceneGPT [79, 80] account for spatial relationships, they do not consider active interactions with the environment, such as opening drawers. In contrast, our representation considers how different actions can modify the environment (e.g., opening a drawer to retrieve a toy `inside`), allowing the system to choose the appropriate exploration and manipulation skills.

3) Foundational Model for Robotics: Many previous studies have used the generalization capabilities, common sense reasoning, and long-horizon planning abilities of VFMs and LLMs for robotic tasks such as manipulation [81–85], navigation [31, 67, 79, 86], and planning [87, 88]. However, these studies did not explore the potential of using VFMs and LLMs for active mobile exploration. In our work, we leverage VFMs to build 3D relational object graphs [4, 5]. We then employ an LLM for decision-making based on an explicit 3D object graph representation of the environment, which our experiments demonstrate to be more efficient and effective than relying on memorizing 2D observation history [10].

B. Method

1) Graph Constructor: Specifically, we first detect objects and obtain the corresponding 3D point clouds $P_t = \{p_t^1, ..., p_t^K\}$, where p_t^i is the point cloud of the ith object. We then associate these with previous object point clouds $P_{t-1} = \{p_{t-1}^1, ..., p_{t-1}^N\}$. The association is resolved by checking detection label consistency and calculating the Intersection over Union (IoU) between P_{t-1} and P_t, as defined in the appendix. Specifically, we create a value matrix $C \in \mathbb{R}^{K \times N}$, where each element is defined as follows:

$$C_{ij} = \begin{cases} \text{IoU}(p_t^i, p_{t-1}^j), & \text{if they have the same label} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

For the ith detected object p_t^i, if $\max_{j \in \{1,...,N\}} C_{ij}$ is below a threshold, it is considered a newly detected object. Otherwise, the ith detected object is associated with the existing object that has the highest C_{ij} value. After associating the current detection with the existing object graph, we could update the existing object nodes with the current observation.

We construct rules that jointly consider geometric information, semantic information, and action information to construct object relations. For example, we use geometric information, such as the bounding boxes of two objects, to determine whether one object is on top of another. Semantic information, like object labels, is also used. For instance, if half of the handle points are within 0.03m of cabinet points, the handle is considered part of the cabinet. Lastly, action information is helpful in determining object relations. For example, when lifting an object to reveal a hidden space, the newly found object is located beneath the lifted object. In summary, our graph encodes five rule-based object relations, including `behind`, `of`, `inside`, `on`, and `under`, as shown in Figure 3. Details regarding the rules to construct object relations can be found on the project page.

C. Low-Level Skills

- `open`: To open a cabinet, we first detect the handle object node and the cabinet object node. Using Principal Component Analysis (PCA), we determine the axis of the handle and the normal axis of the cabinet. Then we can command the end-effector to approach the handle and grasp it. Since the cabinet's articulation is unknown, we initially open the door using impedance control and adjust the target end-effector position and orientation according to manipulation feedback.
  In addition, to make the skill more robust, we take in the grasping feedback. If the grasping fails, we will retry the grasping with different action parameters.

- `flip`: We assume that the robot only flips boxes that are open. The end-effector first approaches the open box from above in a top-down pose, then pushes the box down on its side to flip it over.
- `lift`: We assume that the robot only lifts clothes. The end-effector first approaches the cloth and grasps its center of mass using the top-down pose.
- `push`: In our experiments, the robot will push large objects aside. It will first walk to the front of the object, then move aside. To push the object, it will extend its arm and move sideways while maintaining the arm in an extended position.
- `sit`: In our experiments, the robot will walk to the front of the object and then sit down to check the space beneath it. This sitting action can be executed using the Spot API.
- `collect`: To collect the object, we will first use an off-the-shelf grasping planner or heuristics to grasp it. Then, we command the end-effector to move to the designated position and place the object onto the blanket.

1) IoU Computation: To compute the IoU of two point clouds, we first compute the distance between each point from two point clouds. If a point's minimum distance to another point cloud is smaller than the predefined threshold τ, this point is considered an "intersection". After counting all intersections, we can compute IoU by dividing the intersections by the total point number. The detailed algorithm is shown in Algorithm 1.

2) Unknown Space: Using depth image and camera parameters, we could create a voxel representation of the space. For each voxel, we can have labels, such as `unexplored`, `free`, `unknown`, and `outside`. The definition of each label is listed below:

- `unexplored`: If a voxel is never viewed by a sequence of camera observations, the voxel is labeled as `unexplored`.
- `free`: If a voxel is viewed by a sequence of camera observations and it is in the free space, the voxel is labeled as `free`.
- `unknown`: If a voxel is viewed by a sequence of camera observations and it is occluded, the voxel is labeled as `unknown`.
- `outside`: If a voxel is viewed by a sequence of camera observations and it is outside of the room, the voxel is labeled as `outside`.

Such a voxel representation is critical for VLM decision. For example, when there are a lot `unknown` voxels inside the cabinet or behind the box, VLM will utilize such information regarding the unknown space to decide where to interact.

3) Relation Detection: Given such voxel representation, we can define the following object relations:

- `of`: If a child object can be possibly a part of the parent object, such as a handle is possibly a part of a cabinet, and their point cloud centroids are close enough, the child object node is `of` the parent object node.
- `inside`: If a child object is found after opening or flipping a parent object, the child object is `inside` the parent object node.
- `on`: If a child object's bounding box is within the parent object's bounding box in the x-y plane, and the child object's lowest z value is close to the parent object's highest z value, the child object is `on` the parent object node.
- `under`: If a child object is found after sitting down or lifting, the child object is `under` the parent object node.
- `behind`: If a child object is found after pushing a parent object aside, the child object node is `behind` the parent object node.

4) Graph Serialization: We serialize the graph using the depth-first search. The detailed algorithm is presented in

Algorithm 1 IoU Between Point Clouds

Input: Two point clouds $A \in \mathbb{R}^{N\times 3}$, $B \in \mathbb{R}^{M\times 3}$; threshold τ
Output: IoU matrix IoU

1: **if** $|A| = 0$ or $|B| = 0$ **then return** 0
2: Compute distances D between all points in A and B:

$$D_{k,l} = \|A[k] - B[l]\|, \quad \forall k = 1,\ldots,N; \quad l = 1,\ldots,M$$

3: Compute minimum distances for each point:

$$d_k^{(A)} = \min_l D_{k,l}, \quad \forall k = 1,\ldots,N$$

$$d_l^{(B)} = \min_k D_{k,l}, \quad \forall l = 1,\ldots,M$$

4: Create masks based on threshold τ:

$$\text{mask}_A[k] = \begin{cases} 1, & \text{if } d_k^{(A)} < \tau \\ 0, & \text{otherwise} \end{cases}$$

$$\text{mask}_B[l] = \begin{cases} 1, & \text{if } d_l^{(B)} < \tau \\ 0, & \text{otherwise} \end{cases}$$

5: Compute sums of masks:

$$S_A = \sum_{k=1}^{N} \text{mask}_A[k], \quad S_B = \sum_{l=1}^{M} \text{mask}_B[l]$$

6: Compute IoU:

$$\text{IoU} = \frac{S_A + S_B}{N + M + \varepsilon}$$

7: **return** IoU

(a) Robot (b) Objects

Fig. 5: **Experiment Setup.** (a) illustrates the use of a Spot robot equipped with an external RealSense 455. (b) showcases the diverse objects used, emphasizing the system's generalization capabilities across various object types, scenes, and object relations.

Algorithm 2.

Algorithm 2 Serialize Graph

Input: A graph G with nodes V and edges E
Output: A string `texts` representing the serialized graph

```
Initialize texts ← "root\n"
Initialize to_visit ← empty stack
Add root_node to to_visit
while to_visit is not empty do
    // Append the text of the current node to final texts
    Pop first node curr_node from to_visit
    Obtain text curr_text of curr_node
    Append curr_text to return_texts

    // Find children nodes and add to to_visit
    child_nodes ← curr_node.get_children()
    for all child_node in child_nodes do
        Add child_node to to_visit
return return_texts
```

5) Prompts Examples: Here are all the prompt examples we use. We provide minimal prompt examples to the robot for task planning.

```
# explore one cabinet with one handle
graph:
root
\---cabinet_0
    \---handle_1 [obstruction]

# answer
action:
open handle_1 # affected_objects:
cabinet_0, handle_1

# explore one cabinet with multiple
handles
graph:
root
\---cabinet_0
    |---handle_2 [obstruction]
    \---handle_1 [obstruction]

# answer
action:
open handle_2 # affected_objects:
cabinet_0, handle_2
open handle_1 # affected_objects:
cabinet_0, handle_1

# explore one cabinet with multiple
handles, while some handles are opened
graph:
root
|---chair_3 [obstruction]
|---box_4 [obstruction]
|---cloth_5 [obstruction]
\---cabinet_0
    |---handle_1 (opened)
    \---handle_2 [obstruction]

# answer
action:
push chair_3 # affected_objects: chair_3
push box_4 # affected_objects: box_4
lift cloth_5 # affected_objects: cloth_5
open handle_2 # affected_objects:
cabinet_0, handle_2

# explore several cabinets and
obstruction objects
graph:
root
|---cabinet_0
|   |---handle_1 (opened)
|   |   \---object_3
|   \---handle_2 (opened)
|       \---object_5
|---object_7
|---box_9 [obstruction]
\---cabinet_6
    \---handle_4 [obstruction]

# answer
action:
open handle_4 # affected_objects:
cabinet_6, handle_4
push box_9 # affected_objects: box_9

# explore several cabinets and
obstruction objects
graph:
root
|---cabinet_0
|   |---handle_1 (opened)
|   |   \---object_3 (moved)
|   |       \---object_8
|   \---handle_2 (opened)
|       \---object_5 (moved)
|---object_7
|---box_9
\---cabinet_6
    \---handle_4 (opened)

# answer
action:
none

# explore several cabinets and
obstruction objects
graph:
root
|---cabinet_0
|   |---handle_1 (opened)
```

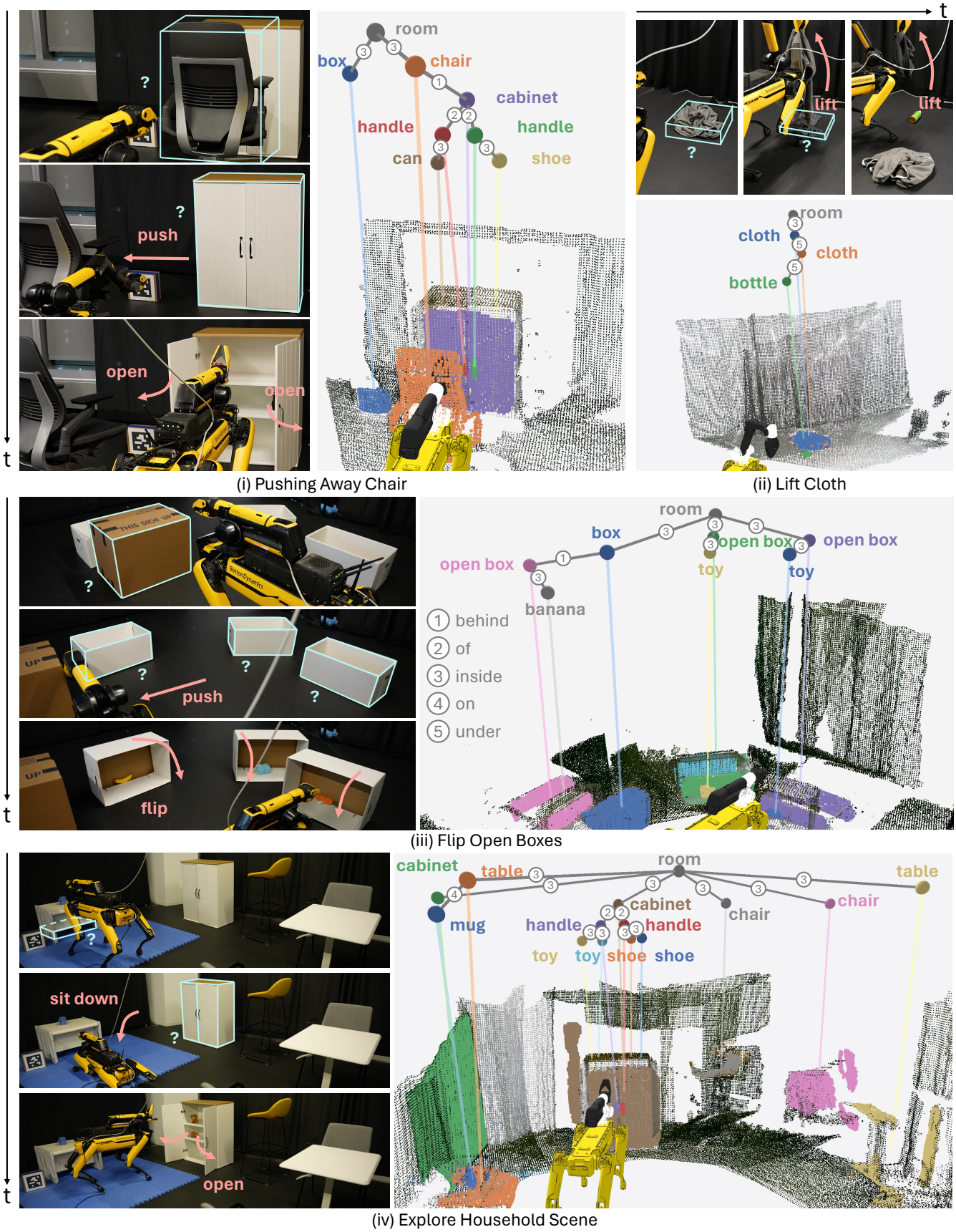


Fig. 6: **Qualitative Results.** We evaluate our system's exploration capabilities across various tasks, including pushing the chair aside to reveal space behind it, lifting cloth to check underneath, flipping open boxes to inspect the contents, and exploring a household scene. These tasks showcase the system's ability to generalize across different object types, scenarios, and object relations. Additional tasks can be found on the project page.

```
|    |    \---object_3 (moved)
|    |        \---object_9 (moved)
|    \---handle_2 (opened)
|        \---object_5 (moved)
|---object_7
\---cabinet_6
    \---handle_4 (opened)

# answer
action:
none

# explore several cabinets and
obstruction objects
graph:
root
|---sealed_box_1
|---sealed_box_2
\---open_box_3 [obstruction]

# answer
action:
flip open_box_3 # affected_objects:
open_box_3
```

D. Experiments

1) Full Qualitative Results: Due to the page limitation, we only show partial results in our main paper. Here we list more complete qualitative results in Figure 6.

2) Comparisons with Baselines: Table I shows the quantitative results of comparisons with baselines.

3) Ablation Study: We also study how our system's performance varies with the number of examples provided to the LLM. We reduce the number of examples from 7 to 1 and evaluate the performance on three tasks: flipping boxes, pushing boxes, and lifting cloth, with each task repeated three times. Table II shows performance decreases as the number of examples decreases, underscoring that the examples we provided to the LLM are both minimal and necessary for task planning.

	LLaVa	Gemini	GPT-4o	Heuristics	Ours
Flipping Boxes					
Success ↑	0%	0%	0%	0%	**80%**
OR ↑	0%	0%	0%	0%	**70%**
GED ↓	2.3	2.3	2.1	2	**1**
Opening Drawers					
Success ↑	40%	**80%**	60%	60%	**80%**
OR ↑	60%	**90%**	80%	72%	88%
GED ↓	7.5	5.9	6.1	3	**2.4**
Checking Underneath					
Success ↑	60%	40%	0%	0%	**90%**
OR ↑	60%	40%	0%	0%	**90%**
GED ↓	2.2	2.7	3.2	3.1	**1.5**
Pushing Boxes					
Success ↑	0%	0%	0%	0%	**70%**
OR ↑	0%	0%	0%	0%	**70%**
GED ↓	4.1	4.3	4	4	**1.2**
Lifting Cloth					
Success ↑	10%	40%	**100%**	0%	90%
OR ↑	10%	40%	**100%**	0%	90%
GED ↓	2	1.9	1.2	2.2	**0.3**
Average					
Success ↑	22%	32%	32%	12%	**82%**
OR ↑	26%	34%	36%	14.4%	**81.6%**
GED ↓	3.62	3.42	3.32	2.86	**1.28**

TABLE I: **Quantitative Results.** We quantitatively evaluate our system on five tasks, each repeated ten times, and compare it with four baselines: LLaVa, Gemini, GPT-4o, and heuristics. The evaluation metrics include success rate, Object Recovery (OR), and Graph Editing Distance (GED), and metrics are defined in Section III-D. Our results show that our approach is more effective at accomplishing exploration tasks and is capable of constructing more accurate object graphs.

Number of Examples	7 (Ours)	5	3	1
Success Rate	**89%**	67%	56%	11%
Object Recovery	**89%**	67%	56%	11%
GED	**0.33**	0.89	1.00	2.67

TABLE II: **Ablation Study.** We examine how our system's performance changes based on the number of examples fed into the LLM. This figure shows that performance worsens as the number of examples decreases, demonstrating that the examples we provide are both minimal and necessary.