Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation

Daniel Honerkamp[1*], Martin Büchner[1*], Fabien Despinoy[2], Tim Welschehold[1], Abhinav Valada[1]

***Abstract*—To fully leverage the capabilities of mobile manipulation robots, it is imperative that they are able to autonomously execute long-horizon tasks in large unexplored environments. While large language models (LLMs) have shown emergent reasoning skills on arbitrary tasks, existing work primarily concentrates on explored environments, typically focusing on either navigation or manipulation tasks in isolation. In this work, we propose MoMa-LLM, a novel approach that grounds language models within structured representations derived from open-vocabulary scene graphs, dynamically updated as the environment is explored. We tightly interleave these representations with an object-centric action space. Importantly, we demonstrate the effectiveness of MoMa-LLM in a novel semantic interactive search task in large realistic indoor environments. The resulting approach is zero-shot, open-vocabulary, and readily extendable to a spectrum of mobile manipulation and household robotic tasks. Through extensive experiments in both simulation and the real world, we demonstrate substantially improved search efficiency compared to conventional baselines and state-of-the-art approaches. We make the code publicly available at http://moma-llm.cs.uni-freiburg.de.**

I. Introduction

Interactive embodied AI tasks in large, unexplored, human-centered environments require reasoning over long horizons and a multitude of objects. Recent advancements have demonstrated the potential of large language models (LLMs) in generating high-level plans [1]–[4]. However, these efforts have predominantly focused on fully observed environments such as tabletop manipulation, or a priori explored scenes, struggling to generate executable and grounded plans suitable for real-world execution. This problem is strongly exacerbated in large scenes with numerous objects and long time horizons [5], [6]. In turn, this increases the risk of generating impractical sequences or hallucinations [7], [8]. Furthermore, operating in interactive scenes comprising articulated objects introduces a multitude of potential states and failure cases.

To address these challenges, we propose grounding LLMs in dynamically built scene graphs. Our approach incorporates a scene understanding module that constructs open-vocabulary scene graphs from dense maps and Voronoi graphs. These diverse representations are then tightly interweaved with an object-centric action space. Leveraging the current scene representation, we extract structured and compact textual representations of the scene to facilitate efficient planning with pre-trained LLMs.

To evaluate our approach, we formulate an interactive semantic search task, extending previous tasks [9] to more complex scenarios. The agent is tasked with finding a target object within an indoor environment, encapsulating real-world challenges where the agent must navigate through the environment, open doors, and search inside cabinets and drawers to find the desired object. This task is challenging as it requires reasoning about manipulation and navigation skills, operating in unexplored environments, spanning large apartments with numerous rooms and objects. Furthermore, we introduce a novel evaluation paradigm for object search tasks, employing full efficiency curves to remove the dependency on arbitrary time budgets and propose the *AUC-E* metric to distill these curves into a single metric. We perform extensive experimental evaluations in both simulation and the real world, outperforming state-of-the-art approaches across diverse fields. Our approach is zero-shot, open-vocabulary, and inherently scalable to various mobile manipulation and household robotic tasks.

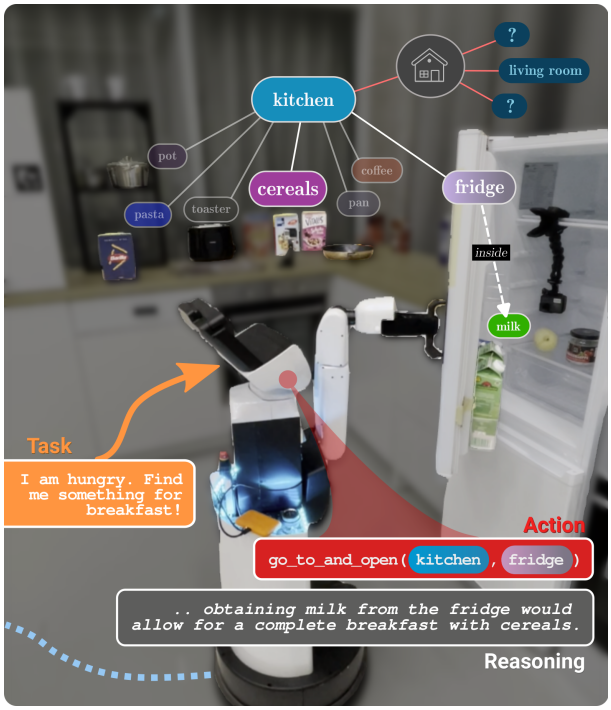

Fig. 1. MoMa-LLM performs long-horizon interactive object search in household environments from language queries using dynamically built scene graphs.

To summarize, our main contributions are

- A scalable scene representation centered around a dynamic scene graph with open-vocabulary room identification.
- Structured compact knowledge extraction to ground LLMs in scene graphs for large unexplored environments.
- Semantic interactive search task for large scenes with numerous objects and receptacles.
- Novel evaluation paradigm for object search tasks through full efficiency curves, instead of a single time budget.
- We make the code publicly available at http://moma-llm.cs.uni-freiburg.de.

* Equal contribution.
[1] Department of Computer Science, University of Freiburg, Germany.
[2] Toyota Motor Europe (TME).
This work was funded by Toyota Motor Europe (TME) and an academic grant from NVIDIA.

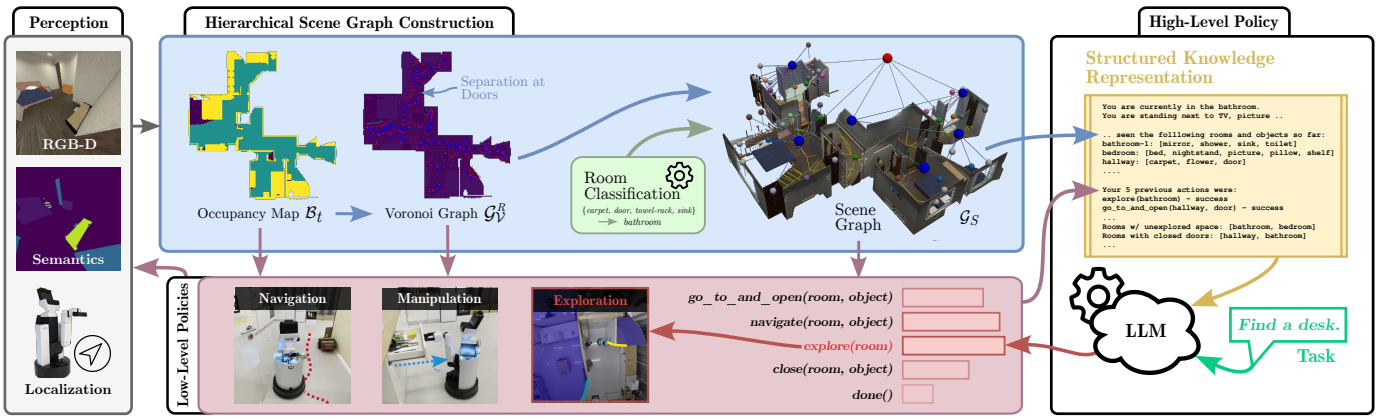

Fig. 2. MoMa-LLM: From posed RGB-D images and semantics, we construct a semantic 3D map from which we extract a various occupancy maps in the BEV space and construct a navigational Voronoi graph. Through room clustering and room-object assigments we then build up a hierarchical scene graph. From this scalable scene representation, we extract the task-relevant knowledge and encode it into a structured language representation. A large language model then produces high-level commands which are executed by low-level subpolicies. These in turn draw on and update the scene representations.

II. Problem Statement: Embodied Reasoning

We introduce the task of *semantic, interactive object search*. In contrast to most existing works [10]–[13], interactive object search requires manipulation of the environment to navigate and explore it. Doors may block pathways and objects may be stored away in receptacles. We extend the interactive task [9] to a much larger number of objects and receptacles and a prior distribution of realistic room-object and object-object relations. As a result, other objects in the scene can provide valuable information about the position of the target. While existing tasks such as the Habitat challenge and RoboTHOR use semantic placements, they do not support any physical interactions or objects placed within receptacles. The task is deemed successful if the agent has observed an instance of the target category and calls $done()$. Full details in Suppl. S.2.

III. MoMa-LLM

We propose *MoMa-LLM*, which grounds large-language models in hierarchical 3D scene graphs $\mathcal{G}$ that hold object- and room-level entities as well as a navigational graph. The LLM provides high-level actions that are executed through low-level skills as shown in Fig. 2. In general, we assume access to ground truth perception as the focus of this work is on the reasoning aspect.

A. Hierarchical 3D Scene Graph

To provide an LLM with structured input, we craft a hierarchical scene graph that includes a navigational Voronoi graph.

Dynamic RGB-D Mapping: The agent perceives posed RGB-D frames $\{I_0, \dots, I_t\}$ including semantics from the environment. The contained points are projected to a 3D voxel grid. We turn the voxel grid into a two-dimensional bird's-eye-view (BEV) occupancy map $\mathcal{B}_t$ by inferring the highest occupied positions except for those classified as navigable area $\mathcal{F}_t$, which are considered for exploration. As we tackle an interactive problem, our map is dynamically updated based on novel explored areas and object dynamics in the scene.

Voronoi Graph: Subsequently, we abstract from the created dense maps by computing a navigational graph $\mathcal{G}_\mathcal{V}$ that is used in downstream tasks to associate objects in close vicinity or estimate geodesic distances. Based on $\mathcal{F}_t$, we compute a Generalized Voronoi Diagram (GVD) that holds a set of points $\mathcal{V}$ with the same clearance to the closest obstacles drawn from $\mathcal{B}_t$. Given the paths of obtained medial axes, we construct edges $\mathcal{E}$ among $\mathcal{V}$ and obtain our navigational Voronoi graph $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{E})$, that undergoes additional processing steps as delineated in Suppl. S.4

3D Scene Graph: Our approach operates on an attributed 3D scene graph $\mathcal{G}_S$ that holds different abstraction levels, namely rooms and objects. We first separate the global Voronoi graph $\mathcal{G}_\mathcal{V}$ into multiple regions covering distinct rooms. To do so, we eliminate edges and nodes of $G_\mathcal{V}$ near doors. Using a mixture of Gaussians, we generate a two-dimensional probability distribution over all observed doors. Edges of $\mathcal{G}_\mathcal{V}$ are scored based on this distribution and disregarded when exceeding a threshold along with isolated nodes. Following this, we obtain the separated Voronoi graph $G_\mathcal{V}^R$. We then infer the high-level connectivity among rooms by calculating the shortest paths between nodes of $\mathcal{G}_\mathcal{V}$ that belong to disjoint components of $\mathcal{G}_\mathcal{V}^R$. Whenever a path traverses just two distinct rooms as given by $G_\mathcal{V}^R$, they count as immediate neighbors.

Finally, we map objects to rooms. For each object $o \in \mathcal{G}_S$, we identify the node $n_R \in \mathcal{G}_\mathcal{V}^R$ that minimizes the distance to the closest viewpoint from which the object was seen. To do so, we calculate the shortest path from the object o to this viewpoint, which consists of the path on the Voronoi graph and the Euclidean distance d from the Voronoi nodes to the object and viewpoint, respectively. Objects are then assigned to the room of the node n^R in $\mathcal{G}_\mathcal{V}^R$. This prohibits the erroneous assignments of objects to a neighboring room through walls. Doors may be connected to multiple rooms.

Room Classification: Similar to Chen *et al.* [14], we perform room classification by providing an LLM with the set of object categories contained in each room. We perform this as open-set classification, in which we let the LLM freely pick the room categories deemed most appropriate. The resulting LLM prompts are detailed in Fig. S.2. Classification is performed at each high-level policy step, as the explored scene evolves.

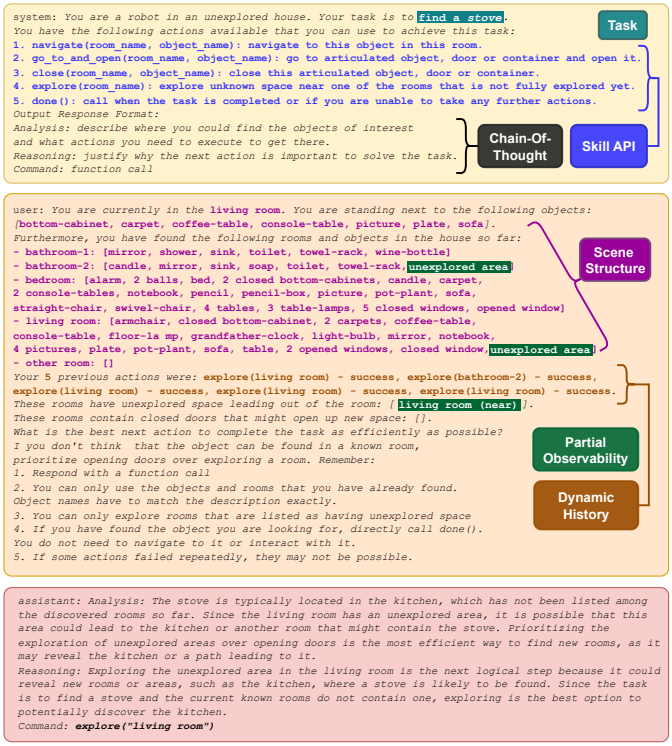

Fig. 3. High-level Reasoning Prompt: We encode the extracted scene to provide structured information to a language model.

B. Grounded High-Level Planning

We design an object-centric action space, which is tightly intertwined with the different granularities of the scene representation. It consists of the following high-level actions: *navigate(room_name, object_name)*, *go_to_and_open(room_name, object_name)*, *close(room_name, object_name)*, *explore(room_name)* and *done()*. The subpolicies then generate actions in the low-level action space and return once they succeed or encounter a failure. Throughout execution, they continuously update the scene representations, as described in Suppl. S.1.

We encode the knowledge of the scene graph into natural language by extracting the relevant components and embedding them in a problem-specific structured manner. Our method fulfills three properties: (i) grounding - guiding the LLM to adhere to the physical realities of the scene, (ii) specificity - avoiding long or irrelevant context that increases hallucinations and the difficulty of the planning problem [7], [8], and (iii) open-set - our method is open-vocabulary and performs in a zero-shot manner. An exemplary prompt is shown in Fig. 3.

Scene Structure: We encode the main room-object structure from the scene graph into a structured list of rooms and their containing objects and encode path distances (based on an A*-planner) by binning them and mapping them to adjectives [15], as detailed in Suppl. S.5. We then employ the following filtering for compact text encodings: we summarize matching nodes with a counter, filter out open doors that provide no new connectivity, and encode object states directly within the object name.

Partial Observability: The initially unknown environment requires explicit reasoning about exploration-exploitation trade-offs. We identify frontiers to explorable areas [16], then leverage the scene graph to provide them with semantic meaning. We associate each frontier with a room, then apply a hole-filling to the BEV map to differentiate whether a frontier is an encapsulated area within a room or leading out to new areas. The second type of unexplored space is receptacles that may contain target objects. We find that the language model is capable of inferring affordances from the object descriptions and states. If trying to open objects that cannot be opened, the according subpolicy will fail and the LLM has to reason about an appropriate response.

History in Dynamic Scenes: We aim to find the most compact representation of previous actions to fulfill the Markov property. For each high-level decision, we encode the latest, dynamically updated scene representation and start a new query to the LLM. To account for previous interactions, we provide the LLM with a history of the last h actions. As the scene graph changes dynamically, we re-align the previous room- and object-centric function calls to the current scene representation, see Suppl. S.5.

Re-trial and Re-planning: As meaningful feedback for failures in the real world remains an open problem [2], we rely on a simple success state for subpolicies, stating *"success"*, *"failure"*, or *"invalid argument"* in case the output of the LLM could not be matched to the scene graph. We differentiate two cases of replanning: if the agent attempted interactions or commands that are deemed invalid or infeasible before execution, we have not gained any new information and continue the conversation. After five failures without state change, we terminate the episode as unsuccessful. If a subpolicy attempted execution but failed, we re-encode the latest scene, update the action history, and let the LLM make a normal next decision with the updated state.

IV. Experiments

Baselines: We provide all baselines except Unstructured LLM with a ground truth $done()$ decision. *Random* uniformly choses among all available actions. *Greedy* triggers the closest feasible action. *ESC-Interactable* scores frontiers based on object-object, object-room co-occurrences and their distance [11]. We extend it to interactive search by scoring openable objects the same way. *HIMOS* learns to combine subpolicies with hierarchical reinforcement learning and a semantic map memory [9]. *Unstructured LLM* provides the scene graph in a JSON format without additional structure to the language model. The prompt is adapted from SayPlan [7] to our scene graph. We use *GPT-4* for the high-level reasoning and *GPT-3.5* for the simpler room classification task [17]. Refer to Suppl. S.6 for more details.

Metrics: We evaluate the success rate (SR) and success weighted by path length (SPL) [18], which does not take the costs of object interactions into account. Both metrics rely on an arbitrary maximum allowed time budget. We argue that the desired time budget depends heavily on the use case and propose the use of a *search efficiency curve*. For each possible budget (number of steps), we calculate the share of episodes that succeeded with this or fewer number of steps. We further reduce this to a single number by calculating the area under the efficiency curve, termed *AUC-E*.

TABLE I
INTERACTIVE OBJECT SEARCH RESULTS IN SIMULATION

Model	SR	SPL	AUC-E	Object Interactions	Distance Traveled	Infeasible Actions
Random	88.8	45.8	71.8	7.0	42.4	–
Greedy	81.1	47.2	68.8	8.6	24.7	–
ESC-Interactive	90.3	52.0	77.5	5.8	28.2	–
HIMOS	**93.1**	47.5	75.7	5.3	43.2	–
Unstructured LLM	81.4	55.0	72.8	**3.9**	**19.2**	0.51
MoMa-LLM (ours)	92.6	**59.9**	**82.5**	4.2	**19.2**	**0.35**

Notes: Object interactions, distance traveled and infeasible actions averaged over all episodes - including early terminated failures. Infeasible Actions: avg. number of steps the LLM produced an infeasible action, resulting in re-planning with continued conversation (cf. Sec. IV-C.4). AUC-E score integrated up to 5,000 steps, at which almost all methods make no further progress.

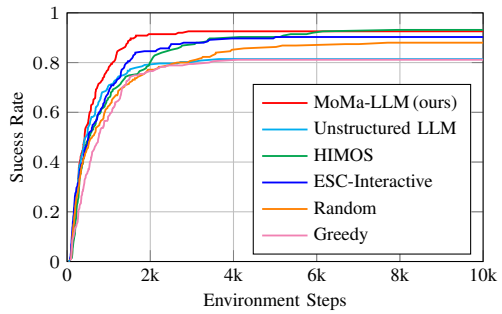

Fig. 4. Interactive search efficiency curve in simulation. Each point depicts the success rate for a given maximum time budget (x-axis).

Simulation Experiments: We instantiate the task in the iGibson simulator [19] with a Fetch robot. For each scene in the test split of the iGibson challenge, we evaluate the agents over 25 procedurally generated episodes with randomized start poses, target objects, and object distributions. In general, we found our policy to be robust to under-segmented rooms even though objects from multiple rooms were, e.g., considered part of a single room. By leveraging the pose from which an object is observed, we reduce wrong object-room assignments *through* walls to a minimum. Following the door-wise separation of rooms, our approach however is prone to *open* room concepts such as combined kitchen and living rooms. For more evaluations and graph depictions, refer to Suppl. S.1.

The results and efficiency curves for the search task are shown in Tab. I and Fig. 4. Given appropriate subpolicies, the heuristics can complete a significant share of episodes. However, they are not sufficient for an efficient search strategy, resulting in low SPL and AUC-E. Similarly, while HIMOS achieves a higher SR, it is unable to explore efficiently as the RL agent struggled with the much larger action space that resulted from the many more interactable instances in our scenes. ESC is able to exploit the co-occurrences to improve over the other baselines. However, given its pair-wise comparisons, it is unable to optimize over longer action sequences. In contrast, MoMa-LLM achieves similar success rates as HIMOS with a much higher search efficiency, both in SPL and AUC-E. We find that the structured prompt representation is essential for this, with the Unstructured LLM performing much worse and resulting in almost 50% more invalid actions. This picture is fortified by the full efficiency curves in Fig. 4, which show that MoMa-LLM achieves the highest performance for all given time budgets. It travels much shorter distances and opens fewer objects, indicating efficient and target-driven behavior.

TABLE II
INTERACTIVE OBJECT SEARCH RESULTS IN THE REAL WORLD

Model	Success Rate	Navig Fails	Manip Fails	Distance Traveled	Object Interact.
ESC-Inter.	80%	2	0	33.9	3.5
MoMa-LLM	80%	1	1	**17.9**	**2.2**

Notes: Dist. travelled is the average distance travelled per episode in meters. Object interactions are the average number of object interactions per episode.

Fig. 5. We construct a real-world apartment covering four rooms and 54 objects and transfer the model to a Toyota HSR robot.

Real-World Experiments: We then transfer our policy to the real world. We create a real-world apartment, consisting of four rooms and use a Toyota HSR robot. For details see Suppl. S.3. We evaluate both MoMa-LLM and the most efficient baseline, ESC. The results are shown in Tab. II, Fig. 5, and the accompanying video. Both methods succeeded in 8/10 episodes and we find that the Voronoi- and scene-graph construction directly transfer to unseen and quite different layouts. Similarly, the system directly transfers to the change in mobile manipulation subpolicies as shown in Fig. 2. The two failures stemmed from irrecoverable failures of the subpolicies: collisions of the base during navigation or of the arm while opening a door. Comparing the two methods, we find confirmation of the simulation results, with MoMa-LLM moving and opening objects more target-driven and efficient. Furthermore, the agent was able to react to the (unseen) failure cases of the subpolicies, such as re-trying to open a drawer when the gripper slipped off the handle.

In contrast to semantic heuristics, our approach is readily expandable to a wide range of household and mobile manipulation tasks. Representative of this, we introduce a *fuzzy search task*, in which the robot only receives a fuzzy task description such as *"Find me something for breakfast."* (Tab. S.3). The agent is capable of finding objects that satisfy respective queries, and correctly reasoning about task completion. In addition to that, we observe that the agent acknowledges whenever currently non-existing subpolicies would have to take over to complete the task.

V. CONCLUSION

We developed a method to ground language models for high-level reasoning with scalable, dynamic scene graphs and efficient low-level policies. We demonstrated the importance of extracting structured knowledge for large and unexplored scenes, outperforming fully learned or co-occurrence-based methods. Moreover, we showed the transfer to a real-world apartment and the extendability to abstract tasks, opening the door towards general household tasks.

REFERENCES

[1] 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," in *Proc. of the Conference on Robot Learning*, 2023.

[2] Z. Liu, A. Bahety, and S. Song, "REFLECT: Summarizing robot experiences for failure explanation and correction," in *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[3] B. Li, P. Wu, P. Abbeel, and J. Malik, "Interactive task planning with language models," *arXiv preprint arXiv:2310.10645*, 2023.

[4] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2motion: From natural language instructions to feasible plans," *arXiv preprint arXiv:2303.12153*, 2023.

[5] E. Greve, M. Büchner, N. Vödisch, W. Burgard, and A. Valada, "Collaborative dynamic 3d scene graphs for automated driving," *arXiv preprint arXiv:2309.06635*, 2023.

[6] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," in *Robotics: Science and Systems*, 2022.

[7] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra *et al.*, "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning," *arXiv preprint arXiv:2307.06135*, 2023.

[8] C. Agia, K. Jatavallabhula, M. Khodeir, O. Miksik, V. Vineet *et al.*, "Taskography: Evaluating robot task planning over large 3d scene graphs," in *Proc. of the Conference on Robot Learning*, 2022, pp. 46–58.

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

[10] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," in *Int. Conf. on Learning Representations*, 2020.

[11] 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," *arXiv preprint arXiv:2301.13166*, 2023.

[12] J. Chen, G. Li, S. Kumar, B. Ghanem, and F. Yu, "How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers," *arXiv preprint arXiv:2305.16925*, 2023.

[13] F. Schmalstieg, D. Honerkamp, T. Welschehold, and A. Valada, "Learning long-horizon robot exploration strategies for multi-object search in continuous action spaces," in *Robotics Research*, 2022, pp. 52–66.

[14] W. Chen, S. Hu, R. Talak, and L. Carlone, "Leveraging large language models for robot 3d scene understanding," *arXiv preprint arXiv:2209.05629*, 2022.

[15] G. Chalvatzaki, A. Younes, D. Nandha, A. T. Le, L. F. Ribeiro, and I. Gurevych, "Learning to reason over scene graphs: a case study of finetuning gpt-2 into a robot language model for grounded task planning," *Frontiers in Robotics and AI*, vol. 10, 2023.

[16] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. of the IEEE Int. Symp. on Comput. Intell. in Rob. and Aut.*, 1997.

[17] OpenAI, "Gpt-4 technical report," *arXiv*, pp. 2303–08 774, 2023.

[18] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[19] C. Li, F. Xia, R. Martín-Martín *et al.*, "igibson 2.0: Object-centric simulation for robot learning of everyday household tasks," *arXiv preprint arXiv:2108.03272*, 2021.

[20] A. Kurenkov, M. Lingelbach, T. Agarwal, E. Jin, C. Li, R. Zhang *et al.*, "Modeling dynamic environments with scene graph memory," in *Int. Conf. on Machine Learning*, 2023, pp. 17 976–17 993.

[21] D. Honerkamp, T. Welschehold, and A. Valada, "N^2m^2: Learning navigation for arbitrary mobile manipulation motions in unseen and dynamic environments," *IEEE Transactions on Robotics*, 2023.

[22] P. He, J. Gao, and W. Chen, "Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing," *arXiv preprint arXiv:2111.09543*, 2021.