Enhancing Object Search by Augmenting Planning with Predictions from Large Language Models

Shahriar Hossain, Abhishek Paudel, and Gregory J. Stein

***Abstract*— We enhance object search in unknown environments by integrating a large language model (LLM) with model-based planning to quickly and reliably locate an object of interest. The LLM is prompted to produce predictions about the likelihood of finding the object of interest used to define a model for planning that the robot then uses to determine its search policy, affording both good performance due to the integration of learning and reliability due to the reliance on classical planning to find the object. From our findings on 200 random maps on the ProcTHOR dataset, our proposed LLM-informed planner, utilizing GPT-4o predictions, achieves a cost reduction of 27.1% and 30.3% compared to the standard baseline and Myopic LLM-informed baseline, respectively.**

I. Introduction

We consider the task of *object search* in household environments, in which a robot is tasked to find an object of interest in the minimum expected time. Even if when the robot is provided a high-level layout of the environment—told what rooms exist and the presence of cabinets, containers, or surfaces within them that may contain the target object—deciding where to prioritize search requires making predictions about uncertainty and incorporating that into planning, a challenging task in general.

Non-learned strategies that tackle this task must often make simplifying assumptions about where the object might be found or the robot's behavior. In the absence of knowledge about the contents of any containers, a greedy planning strategy in this domain may prescribe that the robot simply navigate to and search the nearest unexplored container, repeating until the object of interest is found. However, such naive search strategies, underperform in general, because they lack knowledge about the likely locations of common household objects. Overcoming this limitation instead requires that the robot be imbued with general purpose world knowledge to aid in reasoning about where objects are likely to be found.

To improve the object search capability of robots in unseen environments the inherent knowledge of a large language model (LLM) or vision language model (VLM) can be leveraged to enable the robots to make more accurate decisions during object search. In recent times, we have witnessed planning approaches that rely entirely on LLM [1], [2] and vision language model (VLM) [3], [4]. LLMs have shown limited potential in this domain, one straightforward approach [5] involves asking an LLM where the robot should search next, searching that location, and then requerying the LLM until the object is found. However, multiple recent studies have shown that language models can struggle to serve as effective planners [6]–[9], and so can underperform in this domain as well, despite the general-purpose world knowledge implicit in such models. By contrast, model-based planning is well-suited to reason far into the future to perform the long-horizon reasoning with which LLMs struggle.

In this work, we assert that there is a need for a novel strategy that integrates learning and planning, to both leverage the general-purpose knowledge captured by an LLM and a planning framework to make use of it for effective and reliable decision-making. Taking inspiration from recent work in learning-augmented model-based planning under uncertainty [10], we present a novel approach that uses an LLM to make predictions about statistics of the belief—namely, the likelihood of finding an object of interest in a particular location—parameters used to define a model for planning. We conduct experiments in simulated ProcTHOR environments, which procedurally generate home-like environments well-suited for studying object search tasks. Across 200 maps, our LLM-informed planner, relying on GPT-4o to help make predictions about object likelihoods, shows reductions of 27.1% and 30.3% against Myopic non-learned and learned baselines, respectively, thus demonstrating the efficacy of our approach.

II. Related Work

This section initially provides a brief overview of using LLMs and VLMs as planners. Then it summarizes some model-based learning techniques used in planning.

A. LLM- *and* VLM-*Based Planning*

Huang et al. [11] demonstrate the use of VLMs in planning complex robotic actions in dynamic environments. Shirai et al. [12] introduce a ViLaln combining LLMs and VLMs to generate task planning that is machine-readable. Zhang et al. [13] focuses on visual ground planning to leverage VLMs to detect action failures and verify potential actions.

LLM-based approaches [14], [15] propose to use language-based frameworks that enhance the capabilities of robots that successfully utilize feasibility heuristics and enhance task planning and situation handling in open-world environments for robots. Additionally, Guan et al. [16] and Kambhampati et al. [17] address improvement in PDDL quality and combining LLMs with external verifiers. SayCan [18] introduces the combination of low-level tasks with LLMs to perform complex tasks while SayNav [19] leverages

Shahriar Hossain, Abhishek Paudel, and Gregory J. Stein are with the Department of Computer Science, George Mason University, Fairfax, Virginia, USA. {shossa4, apaudel4, gjstein}@gmu.edu

LLMs to generate step-by-step navigation plans in large-scale unknown environments incrementally building a 3D scene graph during exploration and using LLMs for real-time high-level planning. DynaCon [20] presents a planning system with LLMs to provide a semantic understanding of the environment enabling robots to improve task execution.

B. Object Search and Learning-for-Planning

Ye et al. [21] propose a hierarchical policy learning model that combines intrinsic and extrinsic motivations to improve object search efficiency in sparse reward environments. This approach allows robots to prioritize exploration based on expected task relevance and reward outcomes.

Li et al. [22] compares model-free and model-based learning-informed planning strategies for PointGoal navigation. While model-based approaches use learned environment models to create more intelligent, effective decisions, model-free approaches only use reinforcement learning, necessitating large amounts of data and training. According to the study, model-based planning provides more efficiency and adaptability, especially in new situations.

III. LLM-informed Model-Based Planning

A. Problem Formulation

Our robot is tasked to do object search to reach a target object in a household environment at a minimum expected cost (distance). *Containers* are objects that can contain other objects, like a bed, dresser, countertop, etc., and so the set of containers defines the robot's set of available search actions. The containers are located in different rooms in the household environment. We presume that the robot has access to a low-level navigation planner and controller that can be used to move about and interact with the environment and determine the costs of each. As such, the aim of our planner is to determine the sequence of container search actions that minimize the expected cost.

B. Approach

To achieve effective household object search, we introduce an approach for LLM-*informed model-based object search*, in which we seek to perform model-based planning wherein the robot's behavior is informed by the predictions about object locations generated by an LLM.

Our approach takes inspiration from the recent Learning over Subgoals Planning (LSP) approach of Stein et al. [10]. Their approach, designed around the aim of effective long-horizon navigation, is centered around using learning to estimate statistics associated with temporally extended actions for exploration; a learned model, trained in environments similar to those the robot sees when deployed, estimates the goodness of each such exploratory action and the likelihood exploring the space to which the action corresponds will reach the unseen goal. In the event the action fails, the robot must select another, proceeding until the goal is found.

In our approach, which we term LLM-LSP, we adopt a similar planning abstraction. For our task of object search, the robot's action space $\mathcal{A}$ consist of actions to search each

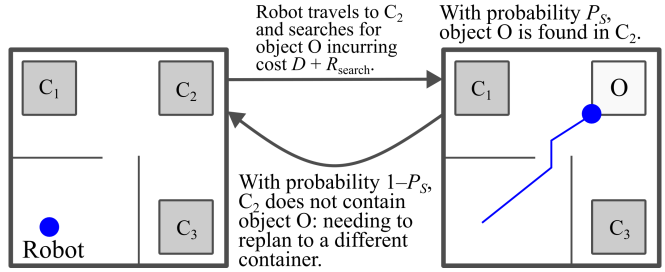

Fig. 1. Overview of our LSP-LLM approach

of the containers within the environment. Under our model, each search action has an immediate cost of first traveling to the container—corresponding to a distance $D(m_t, a_t)$ computed via A* from the occupancy grid—-and then searching the container for the object, which has a (known) cost $R_{\text{search}}(a_t)$. For each such search action a, there is an inherent probability of success $P_S(a)$ that the object is located in the container searched by action a; in the event the object is not found, which occurs with probability $1 - P_S(a)$, the robot must continue searching and so select another action. Under this model, the optimal expected cost of a state-action pair can be represented by a Bellman equation:

$$Q(\{m_t, g_t\}, a_t \in A) = D(m_t, a_t) + R_{\text{search}}(a_t) \\ + (1 - P_S(a_t)) \left[\min_{a_{t+1}} Q(\{m_t, g_t\}, a_{t+1}) \right] \quad (1)$$

The objective of planning is thus to find the sequence of search actions that minimize the expected cost to find the object of interest.

The likelihood of finding the object in each of the containers, the success probability P_S, is not known in advance and so we instead query an LLM, specifically GPT-4o, to provide them based on its general-purpose world knowledge. As mentioned above, we treat the LLM as a commonsense knowledge repository and operate under the assumption that it will contain generally reasonable understanding of where to look for objects and where *not* to look for them. Though the LLM is not particularly effective at directly planning in this domain, querying it for statistics about the world is a much more well-scoped task that does not require explicitly reasoning multiple steps into the future and so allows us to leverage the knowledge contained within the LLM without the need to rely upon it to directly make effective decisions.

C. Obtaining Object Likelihoods from an LLM

To use the LLM, we design a prompt that provides the necessary contextual information—a summary of the environment—followed by a query to estimate the likelihood the object of interest will be found at any one of the locations. Our prompt is built around four main elements: (1) a description of the setting and the role that the LLM will serve, (2) a description of the house including a list of the rooms present and the containers they contain, (3) an example for reference, and (4) the query asking for the probability of finding the object of interest in a container

TABLE I
AVG. COST FOR 200 HOUSEHOLD ENVIRONMENTS

Planner	Avg. Cost (metres)
Myopic Optimistic	16.1
Myopic Learned with GPT-4o	16.8
LSP-LLM with GPT-4o (ours)	**11.7**

within a particular room. The output of the LLM is then parsed to obtain the probability. We repeat this process until all probabilities needed for planning are determined. We provide a full example prompt for our system in Sec.V.

We note that planning via Eq. (1) can scale poorly as the number of potential containers grows. As a cost-saving measure, we use a heuristic to pick a subset of the best 8 containers including the 8 containers nearest to the agent, and then the 8 highest likelihood of the remaining containers. These are used as the action set for the robot. Whenever a container is searched and found not to contain the object of interest, the robot chooses the 8 best of the remaining unsearched containers. This process of searching and replanning continues until the object is found or all containers are exhausted.

IV. SIMULATED EXPERIMENTS IN PROCTHOR

We conduct simulated experiments in 200 distinct household environments drawn from the PROCTHOR [23] dataset, which consists of procedurally generated homes for use in rearrangement-style tasks. In our experiments, we presume that the robot has access to the underlying occupancy grid representation and is provided the room layout and containers that may contain objects of interest, but not what is inside those containers, as it must search them to reveal the objects. In this work, we evaluate the performance of three distinct planning strategies:

LSP-LLM (ours) Our LLM-informed planner, as described in Sec. III-B, which uses probability values obtained from an LLM to define parameters for a model-based planner.

Myopic Optimistic (non-learned baseline) This planning strategy optimistically assumes that containers in the map may contain the object of interest. This assumption results in a greedy planning strategy in which the robot systematically searches the nearest unexplored container.

Myopic Learned (informed baseline) This planning strategy leverages the probabilities produced from the LLM, yet navigates directly to the container with the highest probability, rather than using it to inform planning. As such, this strategy is also greedy.

In Table I we show the average performance of each planning approach among the 200 maps where our LSP-LLM planner outperforms all other planners. From Fig. 2 we observe that our proposed LLM-based planner using predictions from GPT-4o shows a cost reduction of 27.07% and 30.29% compared to the Myopic Optimistic baseline and Myopic LLM-informed baseline, respectively making our proposed method the best among all.

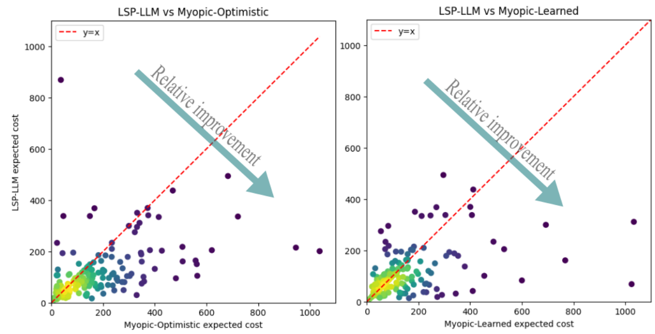

Fig. 2. Results for average cost for 200 random maps using **GPT-4o**. Our LSP-LLM planner outperforms both the Myopic learned planner and Myopic optimistic planner by 27.1% and 30.3% respectively

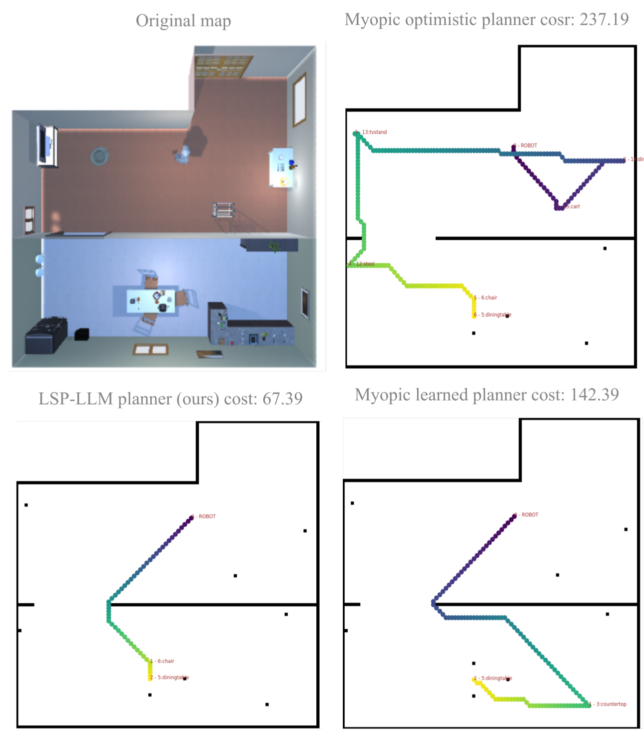

Fig. 3. Trajectory of the robot for each of the planners on a map where the target object is a pan on a dining table

In Fig. 3 LSP-LLM planner shows the most efficient path planning around containers with minimal deviation, indicating a strong ability to anticipate the environment. The Myopic Learned planner, though more efficient than the optimistic planner, still exhibits some unnecessary exploration. The Myopic Optimistic planner blindly explores the nearest containers with respect to the robot's present location, leading to excessive detours and inefficiency in locating the target object.

Fig. 4 compares the performance of three planners on three more maps where a robot searches for a target object. In comparison to the other planners, our LSP-LLM planner frequently makes quick progress toward finding the object,

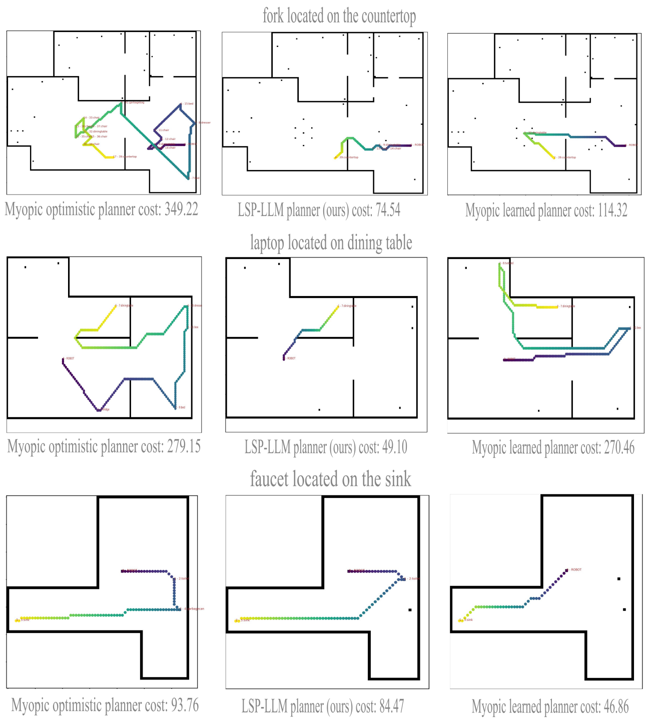

Fig. 4. Trajectory of the robot for each of the planners on three different maps where the target object where the target object is a fork located on the countertop, laptop located on a dining table, and faucet located on the sink respectively

occasionally stopping at other containers near to its path along the way. The Myopic Learned planner shows planning costs that are lower than those of the Myopic Optimistic planner but higher than the LSP-LLM planner, the Myopic Optimistic planner explores excessively and incurs substantially higher costs. The results show that for the Myopic Optimistic planner, good performance requires more than simply making reasonable predictions about object location. Our approach benefits from both model-based planning and the predictions from the LLM, and so is able to outperform the competitive baselines.

In comparison to the Myopic Optimistic planner, the LSP-LLM planner in the final map achieves relatively low cost, though slightly underperforms the Myopic Learned planner. This shows that even if the LSP-LLM is not always optimal it is still efficient and reaches the target object due to the model-based planner coupled with the LLM having real-world knowledge.

V. Conclusion and Future work

In this research, we present an LLM-based object search approach. We extract predictions of finding particular target objects in different containers of different rooms in an apartment from GPT-4o. We then feed these prediction values into the LSP planner to find the cost output and compare it with cost outputs from other planners. From our findings, our proposed LLM-based approach outperforms other planners. In future work, we aim to conduct more experiments on different household environments and compare the results yielded by our LLM-based approach with the other baseline approaches.

Appendix: Sample GPT-4o Instruction

This section includes the sample of the entire prompt and LLM *output.*

Input Instruction:

You are serving as part of a system in which a robot needs to find objects located around a household. Here is a schema that describes the connectivity of rooms in the house: The apartment contains: a bedroom. bedroom contains: bed, dresser, diningtable, chair, chair, armchair, safe, desk, dogbed.

You will be asked to estimate the probability (a value between 0% and 100%) of where objects are located in that house, leveraging your considerable experience in how human occupied spaces are located. You must produce a numerical value and nothing else, as it is important to the overall functioning of the system. Here is an example exchange for an arbitrary house:

User: "What is the likelihood that I find eggs in the refrigerator in the kitchen." You: "90%"

The logic here is that there is a high likelihood that a typical refrigerator in the kitchen contains eggs, but it is not guaranteed as not all refrigerators have eggs.

Here is your prompt for today: "What is the likelihood that the desk in the bedroom contains a tabletopdecor?"

Output: 25%

References

[1] V. Dorbala, J. Mullen, and D. Manocha, "Can an embodied agent find your "cat-shaped mug"? llm-based Zero-Shot object navigation," *IEEE Robotics and Automation Letters*, vol. 9, pp. 4083–4090, 2023.

[2] K. Rana, J. Haviland, S. Garg, *et al.*, "SayPlan: Grounding large language models using 3d scene graphs for scalable robot task planning," in *Proceedings of The 7th Conference on Robot Learning*, 2023.

[3] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "ZSON: Zero-Shot Object-Goal navigation using multimodal goal embeddings," in *Neural Information Processing Systems*, 2022.

[4] S. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "CoWs on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation," in *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[5] V. N. Devarakonda, R. G. Goswami, A. U. Kaypak, N. Patel, R. Khorrambakht, P. Krishnamurthy, and F. Khorrami, "OrionNav: Online planning for robot autonomy with context-aware llm and open-vocabulary semantic scene graphs," 2024. [Online]. Available: https://arxiv.org/abs/2410.06239

[6] S. Kambhampati, "Can large language models reason and plan?" *Annals of the New York Academy of Sciences*, vol. 1534, no. 1, pp. 15–18, 2024.

[7] K. Valmeekam, A. Olmo, S. Sreedharan, and S. Kambhampati, "PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change," in *Neural Information Processing Systems*, 2022.

[8] S. Hao, Y. Gu, H. Ma, J. Hong, Z. Wang, D. Wang, and Z. Hu, "Reasoning with language model is planning with world model," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2305.14992

[9] S. Sharan, F. Pittaluga, G. VijayKumarB., and M. Chandraker, "LLM-Assist: Enhancing closed-loop planning with language-based reasoning," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2401.00125

[10] G. J. Stein, C. Bradley, and N. Roy, "Learning over subgoals for efficient navigation of structured, unknown environments," in *Proceedings of The 2nd Conference on Robot Learning*, vol. 87, 2018, pp. 213–222.

[11] 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*, 2023. [Online]. Available: https://arxiv.org/abs/2307.05973

[12] K. Shirai, C. Beltran-Hernandez, M. Hamaya, A. Hashimoto, S. Tanaka, K. Kawaharazuka, K. Tanaka, Y. Ushiku, and S. Mori, "Vision-language interpreter for robot task planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 2051–2058.

[13] X. Zhang, Y. Ding, S. Amiri, H. Yang, A. Kaminski, C. Esselink, and S. Zhang, "Grounding classical task planners via vision-language models," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2304.08587

[14] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2Motion: from natural language instructions to feasible plans," *Autonomous Robots*, vol. 47, pp. 1345–1365, 2023.

[15] Y. Ding, X. Zhang, S. Amiri, *et al.*, "Integrating action knowledge and llms for task planning and situation handling in open worlds," *Autonomous Robots*, vol. 47, pp. 981–997, 2023.

[16] L. Guan, K. Valmeekam, S. Sreedharan, and S. Kambhampati, "Leveraging pre-trained large language models to construct and utilize world models for model-based task planning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 79 081–79 094, 2023.

[17] S. Kambhampati, K. Valmeekam, L. Guan, *et al.*, "LLMs can't plan, but can help planning in llm-modulo frameworks," *ArXiv*, vol. abs/2402.01817, 2024.

[18] A. Irpan, A. Herzog, A. T. Toshev, A. Zeng, A. Brohan, B. A. Ichter, B. David, C. Parada, C. Finn, C. Tan, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on Robot Learning*, no. 2022, 2022.

[19] A. Rajvanshi, K. Sikka, X. Lin, B. Lee, H. Chiu, and A. Velasquez, "SayNav: Grounding large language models for dynamic planning to navigation in new environments," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 34, 2024, pp. 464–474.

[20] G. Kim, T. Kim, S. Kannan, V. Venkatesh, D.-H. Kim, and B.-C. Min, "DynaCon: Dynamic robot planner with contextual awareness via llms," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2309.04077

[21] X. Ye and Y. Yang, "Efficient robotic object search via hiem: Hierarchical policy learning with intrinsic-extrinsic modeling," *IEEE Robotics and Automation Letters*, vol. 6, pp. 4425–4432, 2020.

[22] Y. Li, A. Debnath, G. J. Stein, and J. Kosecka, "Comparison of model-free and model-based learning-informed planning for pointgoal navigation," *ArXiv*, vol. abs/2212.08801, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254853956

[23] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, J. Salvador, K. Ehsani, W. Han, E. Kolve, A. Farhadi, A. Kembhavi, and R. Mottaghi, "ProcTHOR: Large-scale embodied ai using procedural generation," *ArXiv*, vol. abs/2206.06994, 2022.