LLM-PCPP: A novel robot prioritized coverage path planning approach for complex environments

Zhaofeng Hu[1] **and Ci-Jyun Liang**[1]

[1]Department of Civil Engineering, Stony Brook University, USA

zhaofeng.hu@stonybrook.edu, ci-jyun.liang@stonybrook.edu

Abstract -

This study introduces a novel prioritized coverage path planning method that combines the efficiency of Large Language Models (LLMs) with traditional optimization techniques to enhance the identification and traversal of key points of interest (POIs) on semantic maps of complex environments. By leveraging LLMs, the proposed approach dynamically identifies POIs, addressing inaccuracies and inefficiencies associated with manual selection. The LLM-A* algorithm is then utilized to compute the shortest paths between POIs, significantly reducing operational and storage parameters and thereby minimizing computational overhead. Finally, the traversal problem is formulated as a Traveling Salesman Problem for optimal route planning. This method demonstrates substantial improvements in computational efficiency, particularly benefiting resource-constrained robotic systems, and offers a scalable solution for efficient path planning in complex environments.

Keywords -

Prioritized Coverage Path Planning ; Large Language Model; A* Algorithm;

1 Introduction

The use of autonomous robots and drones [1] in task-specific environments has gained significant attention due to their ability to optimize operations, reduce human error, and improve safety [2]. In outdoor construction sites, indoor warehouses, and other complex environments, efficient path planning for robotic systems is essential to achieve task goals effectively. In robot navigation, coverage path planning (CPP) focuses on ensuring complete area coverage [3]. However, these uniform approaches fail to account for the heterogeneous complexity of real-world environments, leading to redundant scans in simple regions and insufficient focus on critical areas.

To address these challenges, prioritized CPP has emerged as a promising solution for selectively covering critical points of interest (POIs) while minimizing resource consumption [3]. However, identifying these POIs is a challenging and error-prone process, often reliant on manual input or predefined heuristics, which limits scalability and introduces inaccuracies. Moreover, computational constraints in resource-limited devices, such as drones or small mobile robots, require more efficient algorithms for coverage path planning.

This study introduces a novel prioritized CPP method that leverages Large Language Models (LLMs) for identifying POIs on semantic maps, combined with the LLM-A* [4] algorithm for optimizing paths between these points. By utilizing the rich contextual information embedded in semantic maps [5], our approach enables more accurate and dynamic identification of key regions of interest, addressing inefficiencies and errors associated with manual selection. LLMs analyze the semantic structure of the environment, allowing the system to prioritize areas based on task-specific relevance. The LLM-A* algorithm further enhances efficiency by reducing operational and storage requirements, making it particularly suitable for resource-constrained robots. After connecting the identified POIs through LLM-A*, the problem is modeled as a Traveling Salesman Problem (TSP) [6] to determine the optimal path.

2 Related Work

2.1 Coverage Path Planning

Coverage Path Planning (CPP) is a fundamental problem in robotics, with applications ranging from environmental monitoring to robotic cleaning and agricultural surveying. Traditional CPP methods aim to ensure complete area coverage [3]. Although these approaches are straightforward and widely used, they often fail to account for semantic information about the environment, leading to inefficiencies in resource allocation and redundant coverage in simple regions. To address these limitations, non-uniform CPP strategies have been proposed. For instance, Kazemdehbashi [7] introduced an adaptive grid-based decomposition algorithm that optimizes UAV coverage paths by dynamically adjusting grid size based on environmental complexity, reducing computation and coverage time. In addition, Girdhar et al. [8] proposed a curiosity-driven path planning approach, leveraging information-theoretic metrics to prioritize regions with higher informational value. Similarly, Sadat et al. [9] introduced a coverage strategy utilizing space-filling curves to achieve efficient exploration of regions with varying levels of interest, making it suitable for tasks requiring adaptive focus on high-priority areas. Manjanna et al. [10] proposed a reward-driven selective coverage path planning algorithm using a finite-horizon Markov Decision Process (MDP), which prioritizes high-value regions and generates time- and energy-efficient trajectories for multi-robot path planning.

While non-uniform CPP strategies demonstrate adaptability and regional prioritization, their reliance on predefined heuristics and manual metrics often limits their applicability in complex environments. Our approach addresses these limitations by utilizing LLMs for automated and dynamic identification of POIs from semantic maps. Additionally, the integration of the LLM-A* algorithm ensures computational efficiency and scalability, making it well-suited for diverse real-world scenarios.

2.2 Large Language Models in Path Planning

Large Language Models (LLMs), such as GPT [11], have been employed to generate task-specific goals, interpret natural language instructions, and provide context-aware decision-making for navigation tasks [12]. The integration of LLMs into robots has opened new avenues for enhancing path planning through their ability to process and interpret high-level semantic information. Latif [13] proposed a probabilistic path planning approach using GPT-3.5-turbo, showcasing its ability to optimize robot navigation in complex environments. The method outperformed traditional algorithms like A* and RRT in processing time, demonstrating the potential of LLM in terms of efficiency and cost savings. Meng et al. [4] introduced LLM-A*, a hybrid algorithm combining the global reasoning capabilities of large language models with the precise pathfinding of A*. The approach significantly reduces computational operations and memory usage while maintaining path validity, making it well-suited for large-scale path planning scenarios.

Furthermore, LLMs have been utilized in coverage path planning tasks. Kong et al. [14] developed a multi-layer framework that integrates LLMs into coverage path planning for mobile robots. Their approach enhances the spatial inference capabilities of LLMs and achieves efficient, reliable path planning by combining high-level reasoning with low-level control, demonstrating the potential for LLMs in embodied AI tasks. These studies demonstrate the potential of LLMs to go beyond traditional numerical optimization methods by incorporating contextual and semantic understanding into the path planning process. Despite these advancements, those methods are limited to path planning tasks between two points or simply covering path planning tasks without obstacles. Our study builds on these foundations by leveraging LLMs to identify POIs and integrating them into a prioritized coverage path planning framework. This framework can handle complex environments and effectively bridge the gap between semantic understanding and efficient navigation.

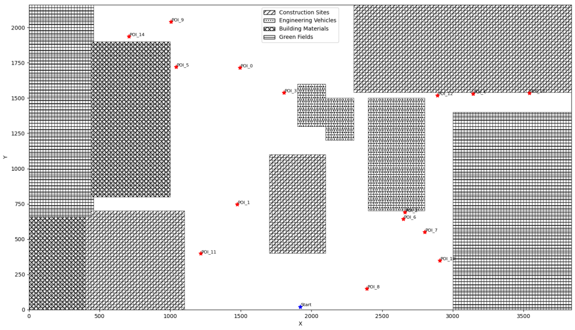
3 Methodology

Our proposed methodology integrates semantic information extraction, point of interest (POI) identification, and optimized path planning. First, the environment is represented as a 2D semantic grid map, combining spatial and contextual information. Points of Interest (POIs) are then identified using a Large Language Model (LLM), which leverages semantic reasoning to prioritize task-relevant regions. The LLM-A* algorithm computes efficient paths between POIs while maintaining scalability and computational efficiency. Finally, the problem is formulated as a Traveling Salesman Problem (TSP), with the optimal route derived through heuristic and constraint-based optimization.

3.1 Semantic Information Grid Map Construction

To enable computational processing and algorithmic path planning, the first step is to convert the real-world environment into a 2D semantic information grid map. This discretization captures the spatial features of the environment while simplifying the problem for efficient computation. Data representing the real-world environment is collected through various sources, such as satellite imagery, LiDAR scans, or high-resolution drone footage (Figure 1a). This data provides essential spatial and structural details, including boundaries, obstacles, and open areas. Subsequently, data processing is performed to construct the grid map, ensuring that the number and size of the grids align with the resolution of the original image to maintain high-fidelity modeling, as shown in Figure 1b. Each cell in the grid represents a discrete area in the real-world environment. The processed data is converted into a grid-based matrix representation. Each cell in the matrix is assigned a binary value indicating its navigability. For example, cells representing obstacles are marked as non-navigable, while open spaces are marked as traversable. We convert the obstacles to rectangular shapes for computational efficiency, but they can be represented as complex geometries. This grid map forms the basis for subsequent steps, enabling spatial computations and path planning algorithms. In addition, this grid map is accompanied by semantic information, i.e., natural language descriptions indicating the location and type of obstacles, which will be provided to the LLM for reasoning.

(a) A real environment captured by a drone.



(b) Semantic grid map.

Figure 1. Real Environment and Semantic Grid Map.

3.2 Point of Interest (POI) Identification

The process begins with identifying Points of Interest (POIs) within the grid map. These POIs represent critical locations that must be prioritized during path planning and are determined using the semantic and contextual capabilities of a Large Language Model (LLM). By employing prompt engineering [15] to provide the LLM with a detailed description of the grid map and task-specific objectives, such as locating operational zones, high-risk areas, or strategic points, the model processes the input and leverages its contextual understanding to identify areas of significance based on task requirements.

The LLM then generates a list of potential POIs, which are subsequently validated against predefined criteria to ensure their relevance and alignment with the overall objectives. For example, POIs may include equipment storage areas, hazard zones, or task-specific targets. Once validated, the identified POIs are annotated on the 2D grid map, as illustrated in Figure 1b. This annotated visualization serves as a tool to verify the accuracy of the LLM's output and provides a clear depiction of the prioritized areas, laying the foundation for the subsequent path planning processes.

3.3 Distance Calculation Using LLM-A*

The next step is to compute the shortest paths between all pairs of POIs using the LLM-A* algorithm. This hybrid approach improves computational efficiency and reduces memory usage while maintaining path validity. The LLM-A* algorithm works by integrating LLM-generated waypoints as intermediate targets to guide the A* search process.

The POIs and the obstacles in the grid map are first defined as input states. The LLM-A* algorithm generates a sequence of waypoints using global insights from the LLM, providing a coarse roadmap through the environment. Waypoints divide the entire environment into distinct regions for partitioned processing, followed by path computation using the standard A* algorithm. This method of regional division significantly optimizes the computational cost of heuristic path planning algorithms [16]. Additionally, with the introduction of LLM for waypoint identification, the waypoints gain functional coverage capabilities. While the robot's path length may slightly increase, it enables the robot to cover more points on the map that are rich in semantic information.

The algorithm builds a distance matrix by calculating the shortest paths between all pairs of POIs. Each pairwise calculation leverages the LLM-A*'s ability to avoid unnecessary node expansions, significantly reducing the number of operations and storage requirements compared to traditional A*. By incorporating LLM-generated global insights with the deterministic precision of A*, LLM-A* enables the efficient and accurate computation of the distance matrix, laying the groundwork for the subsequent Traveling Salesman Problem (TSP) optimization. This hybrid approach addresses the limitations of standalone A* and LLM-based methods, ensuring both scalability and robustness in large-scale path planning scenarios.

3.4 Traveling Salesman Problem (TSP) Solution

To determine the optimal path visiting all identified POIs exactly once and returning to the starting point, the Traveling Salesman Problem (TSP) is formulated using the distance matrix from 3.3. Google OR-Tools is employed to solve the TSP, utilizing a hybrid approach that balances computational efficiency with solution quality. The initial solution is generated using the *PATH_CHEAPEST_ARC* heuristic, which iteratively selects the minimum-cost edges to construct a feasible route. This strategy provides a fast approximation to the TSP. Further optimization is achieved through constraint-based refinement, ensuring the path adheres to task-specific requirements, such as visiting all POIs exactly once and returning to the starting point. This approach leverages heuristic initialization and constraint optimization to achieve near-optimal solutions while maintaining computational efficiency.

4 Expirement

4.1 Experimental Setup

To evaluate the effectiveness and efficiency of the proposed approach, we conducted experiments in simulated environments with varying levels of complexity, including one virtual indoor scene and two real outdoor construction site scenes. The experiments were designed to evaluate the performance of the LLM-A* algorithm and its integration into the overall system for solving the TSP.

Real outdoor construction site scenarios were represented using 2D grid maps generated with different resolutions and varying distributions of obstacles. The maps ranged from simple layouts with minimal obstacles to highly complex configurations with dense barriers. POIs were placed strategically across the maps to simulate real-world task requirements.

The simulation environment was configured to reflect realistic constraints, including image resolution, number of POIs (5 to 15 per map), and obstacle density (low 10%, medium 30%, and high 50%).

The experiments were implemented in Python, with the LLM-A* algorithm and TSP solver integrated into the workflow. The LLM used for global reasoning was GPT-3.5, accessed via API. For TSP optimization, a genetic algorithm was employed. The intermediate targets for LLM A* were determined in LLM and fed to A* Python code independently.

4.2 Evaluation Metrics

The proposed approach is evaluated using three key metrics: path length, operations, and storage.

Path Length: This metric measures the total distance of the computed path. While shorter paths are typically desirable in many scenarios, for prioritized coverage path planning (PCPP) tasks, a slightly longer path may be beneficial as it allows for the exploration of more semantically rich areas, aligning with the task's objectives.

Operations: The number of computational steps required to compute the path is recorded. This metric reflects the algorithm's efficiency in terms of processing effort and computational complexity.

Storage: The memory usage during the pathfinding process is measured to evaluate the algorithm's scalability and suitability for resource-constrained systems. Lower storage requirements indicate better optimization of the search process.

4.3 Experimental Details

4.3.1 Experiment 1: Complex Outdoor Construction Site Scenario

The first experiment evaluates the performance of the proposed approach in a complex outdoor construction site scenario derived from drone footage. The scene was converted into a high-resolution semantic grid map to simulate realistic navigation challenges. This real-world scenario and related grid map are shown in Figure 1. The resolution of the grid map is 3840×2160 with a high obstacle density, representing 50% of the grid cells. ChatGPT-4o served as the agent for generating semantically meaningful POIs and providing heuristic guidance for the LLM-A* algorithm. Table 1 summarizes the key parameters of this experimental setup.

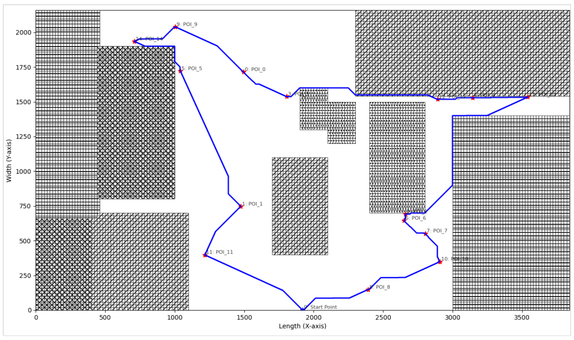
Table 1. Details of Experiment 1: Complex Outdoor Scenario

Parameter	Value
Grid Map Resolution	3840 × 2160
Obstacle Density	High (50%+)
Obstacle Types	Green Fields, Construction Sites, Equipment, Building Materials
Task Guidance	Avoid the obstacles, ensure coverage of most areas
LLM Agent for POIs	ChatGPT-4o
LLM Agent for LLM-A*	ChatGPT-4o
Number of POIs	14

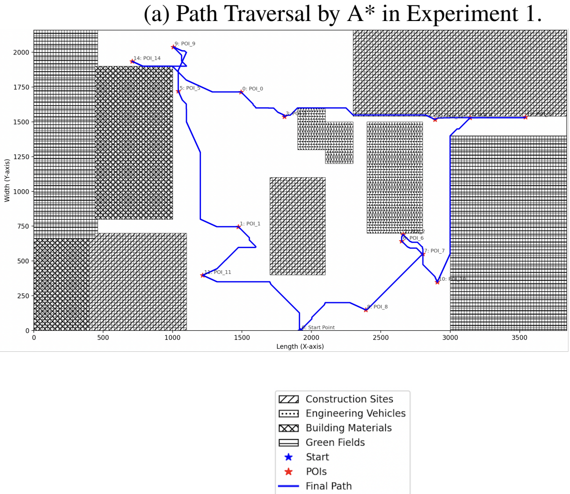
Figure 2 and Table 2 illustrate the comparative performance of A* and LLM-A* in Experiment 1. Figure 2 presents the specific paths taken by each algorithm. The paths highlight LLM-A*'s focus on semantically rich exploration over strict distance minimization. Table 2 provides quantitative insights into the results, demonstrating that while the total path length for LLM-A* increased by 13.6%, it significantly reduced operations by 41.1% and storage requirements by 36.9%. Therefore, Figure 2 and Table 2 highlight LLM-A*'s effective balance between computational efficiency and semantic exploration, making it an appropriate choice for prioritized coverage path planning tasks in complex environments.

Table 2. Comparison of A* and LLM-A* in Experiment 1

Metric	A*	LLM-A*
Total Path Length (TSP)	9270.45	10530.60 ↑ 13.6%
Total Operations	72343616	42636892 ↓ 41.1%
Total Storage	44987486	28383237 ↓ 36.9%



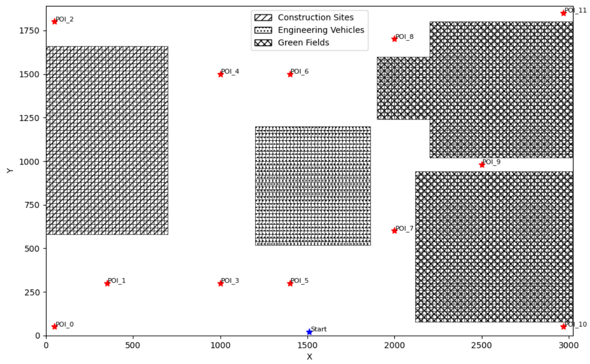
(a) Path Traversal by A* in Experiment 1.



(b) Path Traversal by LLM-A* in Experiment 1.

Figure 2. Path Planning Comparison in Experiment 1: A* vs LLM-A*.

(a) Simple real environment in Experiment 2.



(b) Semantic grid map with POIs in Experiment 2.

Figure 3. Real Environment and Semantic Grid Map in Experiment 2.

4.3.2 Experiment 2: Simple Outdoor Construction Site Scenario

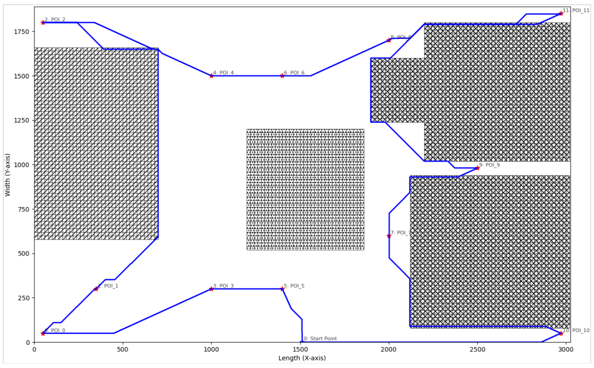
The second experiment evaluates the performance of the proposed approach in a simple outdoor construction site scenario derived from drone footage. The scene was transformed into a semantic grid map with a lower resolution compared to Experiment 1, representing a simpler environment for navigation. Points of Interest (POIs) were generated using ChatGPT-o1, while the LLM-A* algorithm employed ChatGPT-4o for heuristic guidance. The scenario of experiment 2 and the related grid map with POIs are in Figure 3.

This experiment involved a grid map with a resolution of 3024×1890 and a medium obstacle density, representing 30% of the grid cells. The goal was to evaluate the algorithm's performance in less complex scenarios, focusing on efficiency and scalability. Table 3 summarizes the key parameters of this experimental setup.

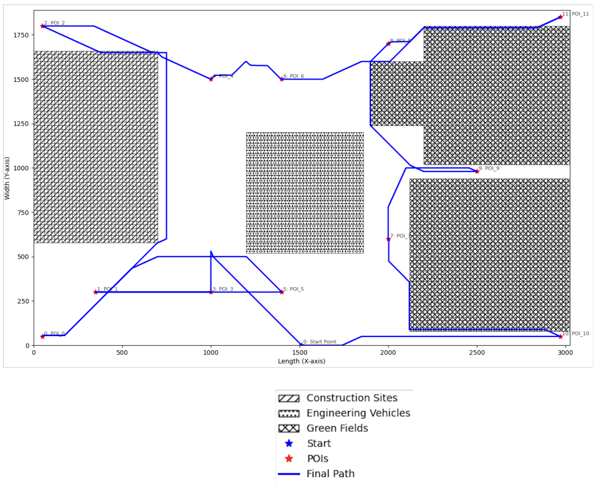
Table 3. Details of Experiment 2: Simple Outdoor Scenario

Parameter	Value
Grid Map Resolution	3024 × 1890
Obstacle Density	Medium (30%)
Obstacle Types	Green Fields, Construction Sites, Equipment, Building Materials
Task Guidance	Avoid obstacles, ensure coverage of most areas
LLM Agent for POIs	ChatGPT-o1
LLM Agent for LLM-A*	ChatGPT-4o
Number of POIs	11

Figure 4 and Table 4 illustrate the specific paths and performance metrics for A* and LLM-A* in Experiment 2. Figure 4 highlights the distinct paths taken by each algorithm, showing LLM-A*'s preference for semantically rich exploration, which resulted in a 23.9% increase in total path length as detailed in Table 4. At the same time, LLM-A* demonstrated superior computational efficiency with a 34.2% reduction in operations and a 34.0% reduction in storage requirements. These findings demonstrate that LLM-A* continues to balance computational efficiency and semantic exploration even in simpler environments.



(a) Path Traversal by A* in Experiment 2.



(b) Path Traversal by LLM-A* in Experiment 2.

Figure 4. Path Planning Comparison in Experiment 2: A* vs LLM-A*.

Table 4. Comparison of A* and LLM-A* in Experiment 2

Metric	A*	LLM-A*
Total Path Length	13562.50	16300.53 ↑ 23.9%
Total Operations	60556220	39873931 ↓ 34.2%
Total Storage	39494015	26063827 ↓ 34.0%

4.3.3 Experiment 3: Virtual Indoor Warehouse Scenario

The third experiment evaluates the performance of the proposed approach in a virtual indoor warehouse scenario. The scene was designed with a resolution of 3024×1890 and included medium obstacle density (30%). The environment featured four types of obstacles: shelves, automated sorting stations, cold storage areas, and hazardous areas. To evaluate the comprehension capabilities of LLMs and expand potential application scenarios, we conducted task guidance tests in this environment focusing not only on obstacle avoidance and maximizing map coverage but also on executing specific task instructions tailored to the setting, such as ensuring detailed exploration around the cold storage area for proper monitoring. LLMs were prompted to understand the task and generate additional Points of Interest (POIs) near the cold storage area. The virtual scenario of experiment 3 only has a designed grid map with POIs, which is in Figure 5.

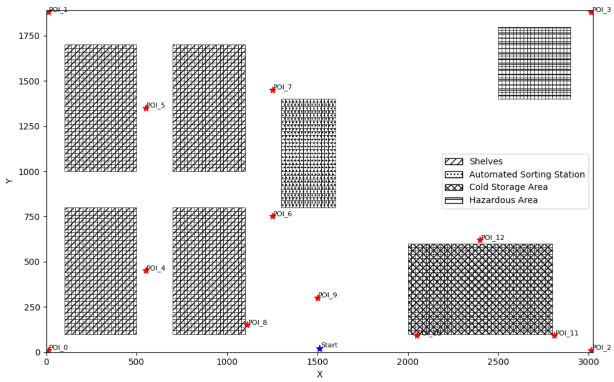


Figure 5. Semantic grid map with POIs in Experiment 3.

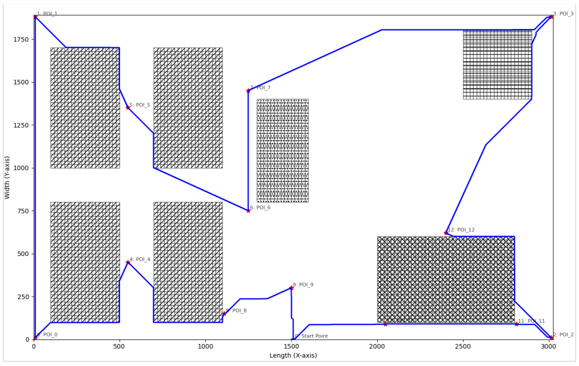
ChatGPT-o1 was used to generate semantically relevant POIs, and ChatGPT-4o provided heuristic guidance for LLM-A*. Table 5 summarizes the key parameters of this experimental setup.

Table 5. Details of Experiment 3: Virtual Indoor Warehouse Scenario

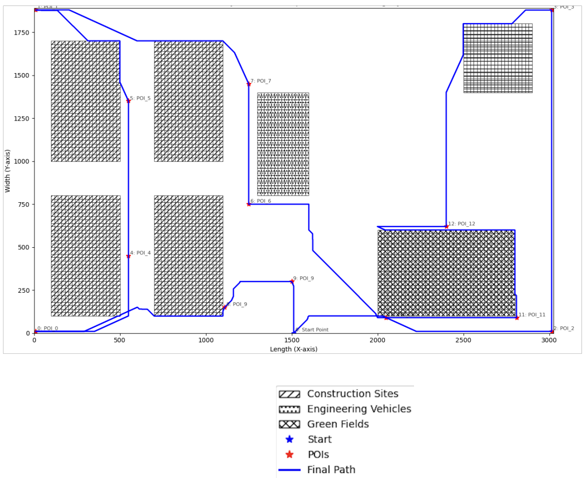
Parameter	Value
Grid Map Resolution	3024 × 1890
Obstacle Density	Medium (30%)
Obstacle Types	Shelves, Sorting Stations, Cold Storage, Hazardous Areas
Task Guidance	Emphasis on cold storage
LLM Agent for POIs	ChatGPT-o1
LLM Agent for LLM-A*	ChatGPT-4o
Number of POIs	78

Figure 6 and Table 6 present the comparative performance of A* and LLM-A* in Experiment 3, conducted within a virtual environment with specific task guidance. Figure 6 illustrates the distinct paths traversed by each algorithm, showcasing LLM-A*'s approach to semantically rich exploration, which aligns with the task's guided objectives. As shown in Table 6, LLM-A* increased the total path length (TSP) by 18.7% (from 13208.39 to 15682.17), reflecting its prioritization of meaningful interaction over strict path minimization.

At the same time, LLM-A* significantly reduced computational demands, with a 41.8% reduction in total operations and a 30.6% reduction in storage requirements. These improvements demonstrate the algorithm's ability to optimize resources while fulfilling specific guidance task requirements in a controlled virtual setting. Overall, Experiment 3 also highlights LLM-A*'s strength in balancing computational efficiency with task-specific semantic exploration.



(a) Path Traversal by A* in Experiment 2.



(b) Path Traversal by LLM-A* in Experiment 3.

Figure 6. Path Planning Comparison in Experiment 3: A* vs LLM-A*.

Table 6. Comparison of A* and LLM-A* in Experiment 3

Metric	A*	LLM-A*
Total Path Length	13208.39	15682.17 ↑ 18.7%
Total Operations	61625226	35904779 ↓ 41.8%
Total Storage	35970025	24992699 ↓ 30.6%

4.3.4 Comparative Analysis

The results provide a comprehensive evaluation of LLM-A*'s performance in varying environments and task setups. While LLM-A* consistently yields longer overall paths (ranging from 13.6% to 23.9% increases), it significantly reduces computational operations (by 34.2% to 41.8%) and lowers storage requirements (by 30.6% to 36.9%) compared to standard A*. These results underscore the advantage of incorporating semantic guidance into path planning: the algorithm selectively emphasizes task-relevant regions (e.g., general guidance for avoiding obstacles in outdoor settings and specific guidance for monitoring cold storage in warehouses), effectively accessing large segments of the search space.

Furthermore, the slight increase in total path length should not be considered a negative outcome. This increase is akin to an exploration phase, effectively expanding the overall coverage area of the map and fulfilling the requirements of PCPP tasks for semantically rich regions. Therefore, this additional path length enhances the algorithm's capability to cover complex environments. In terms of efficiency, LLM-A* consistently demonstrated significant reductions in both computational operations and storage requirements. These results highlight its strong adaptability and scalability, particularly in resource-constrained systems.

Moreover, the performance of LLM-A* remains consistent across all experiments, demonstrating robustness in various levels of environmental complexity. Notably, the performance slightly improved in Experiment 1, which involved complex environments, and in Experiment 3, which featured medium-density environments with specific task guidance. This suggests that the algorithm is not only adaptable to different environments but also excels when faced with customized task requirements.

5 Conclusion

This paper presented a novel Prioritized Coverage Path Planning (PCPP) approach—referred to as LLM-PCPP—that integrates large language models with conventional path planning techniques. By leveraging LLMs to identify context-rich Points of Interest (POIs) and employing LLM-A* to compute efficient paths between these points, the proposed method significantly reduces computational overhead and memory usage. The experimental results across three distinct scenarios (complex outdoor, simple outdoor, and virtual indoor) demonstrated that, while path lengths increased moderately (13.6% to 23.9%), LLM-PCPP consistently reduced the number of operations (34.2% to 41.8%) and memory requirements (30.6% to 36.9%) compared to standard approaches. These findings underscore the advantage of incorporating semantic cues into coverage path planning by LLM.

Future work will focus on exploring collaborative multi-robot systems and developing complex task-specific guidance, such as determining the farthest points for advanced robots to explore in unknown environments. These efforts will further validate the robustness and scalability of LLM-PCPP, positioning it as a promising solution for advanced coverage applications in resource-constrained and semantically diverse settings. In addition, the POIs found by LLM will be validated in the future. This can be done by consulting with field experts. Finally, benchmarking with the state-of-the-art CPP methods will be conducted to evaluate our PCPP method.

References

[1] Ci-Jyun Liang, Thai-Hoa Le, Youngjib Ham, Bharadwaj RK Mantha, Marvin H. Cheng, and Jacob J. Lin. Ethics of artificial intelligence and robotics in the architecture, engineering, and construction industry. *Automation in Construction*, 162: 105369, 2024. doi:10.1016/j.autcon.2023.105369.

[2] Justin M Haney and Ci-Jyun Liang. A literature review on safety perception and trust during human–robot interaction with autonomous mobile robots that apply to industrial environments. *IISE Transactions on Occupational Ergonomics and Human Factors*, 12(1-2):6–27, 2024. doi:10.1080/24725838.2023.2283537.

[3] Anirudh Krishna Lakshmanan, Rajesh Elara Mohan, Balakrishnan Ramalingam, Anh Vu Le, Prabahar Veerajagadeshwar, Kamlesh Tiwari, and Muhammad Ilyas. Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot. *Automation in Construction*, 112:103078, 2020. doi:10.1016/j.autcon.2020.103078.

[4] Silin Meng, Yiwei Wang, Cheng-Fu Yang, Nanyun Peng, and Kai-Wei Chang. LLM-A*: Large language model enhanced incremental heuristic search on path planning. *arXiv preprint*, arXiv:2407.02511, 2024. Available at https://arxiv.org/abs/2407.02511.

[5] Zhengyi Chen, Hao Wang, Keyu Chen, Changhao Song, Xiao Zhang, Boyu Wang, and Jack CP Cheng. Improved coverage path planning for indoor robots based on bim and robotic configurations. *Automation in Construction*, 158:105160, 2024. doi:10.1016/j.autcon.2023.105160.

[6] Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi. The traveling salesman problem. In *Handbooks in Operations Research and Management Science*, volume 7, pages 225–330. Elsevier, 1995.

[7] Sina Kazemdehbashi. Adaptive grid-based decomposition for uav-based coverage path planning in maritime search and rescue. *arXiv preprint*, arXiv:2412.00899, 2024. Available at https://arxiv.org/abs/2412.00899.

[8] Yogesh Girdhar, David Whitney, and Gregory Dudek. Curiosity based exploration for learning terrain models. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 578–584. IEEE, 2014. doi:10.1109/ICRA.2014.6906948.

[9] Seyed Abbas Sadat, Jens Wawerla, and Richard Vaughan. Fractal trajectories for online non-uniform aerial coverage. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2971–2976. IEEE, 2015. doi:10.1109/ICRA.2015.7139617.

[10] Sandeep Manjanna, Nikhil Kakodkar, and Gregory Dudek. Multi-robot path planning for selective coverage. In *ICRA 2016 Workshop on Fielded Multi-robot Systems Operating on Land, Sea, and Air*, Stockholm, Sweden, 2016.

[11] Josh Achiam et al. Gpt-4 technical report. *arXiv preprint*, arXiv:2303.08774, 2023. Available at https://arxiv.org/abs/2303.08774.

[12] Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems*, 36, 2024.

[13] Ehsan Latif. 3p-llm: Probabilistic path planning using large language model for autonomous robot navigation. *arXiv preprint*, arXiv:2403.18778, 2024. Available at https://arxiv.org/abs/2403.18778.

[14] Xiangrui Kong et al. Embodied ai in mobile robots: Coverage path planning with large language models. *arXiv preprint*, arXiv:2407.02220, 2024. Available at https://arxiv.org/abs/2407.02220.

[15] Jules White et al. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint*, arXiv:2302.11382, 2023. Available at https://arxiv.org/abs/2302.11382.

[16] Adi Botea, Martin Müller, and Jonathan Schaeffer. Near optimal hierarchical path-finding. *Journal of Game Development*, 1(1):1–30, 2004.