Embodied AI in Mobile Robot Simulation with EyeSim: Coverage Path Planning with Large Language Models

Xiangrui Kong[1][a], Wenxiao Zhang[2][b], Jin Hong[2][c] and Thomas Bräunl[1][d]

[1]*Department of Electrical, Electronic and Computer Engineering, University of Western Australia, Crawley, Australia*
[2]*Department of Computer Science and Software Engineering, University of Western Australia, Crawley, Australia*

Keywords: Natural Language Processing, Mobile Robots, Path Planning, Indoor Navigation.

Abstract: In recent years, Large Language Models (LLMs) have demonstrated remarkable capabilities in understanding and solving mathematical problems, leading to advancements in various fields. We propose an LLM-embodied path planning framework for mobile agents, focusing on solving high-level coverage path planning issues and low-level control. Our proposed multi-layer architecture uses prompted LLMs in the path planning phase and integrates them with the mobile agents' low-level actuators. To evaluate the performance of various LLMs, we propose a coverage-weighted path planning metric to assess the performance of the embodied models. Our experiments show that the proposed framework improves LLMs' spatial inference abilities. We demonstrate that the proposed multi-layer framework significantly enhances the efficiency and accuracy of these tasks by leveraging the natural language understanding and generative capabilities of LLMs. Experiments conducted in our EyeSim simulation demonstrate that this framework enhances LLMs' 2D plane reasoning abilities and enables the completion of coverage path planning tasks. We also tested three LLM kernels: *gpt-4o*, *gemini-1.5-flash*, and *claude-3.5-sonnet*. The experimental results show that *claude-3.5* can complete the coverage planning task in different scenarios, and its indicators are better than those of the other models. We have made our experimental simulation platform, **EyeSim**, freely available at https://roblab.org/eyesim/.

1 INTRODUCTION

The application of Large Language Models (LLMs) has grown exponentially, revolutionizing various fields with their advanced capabilities (Hadi et al., 2023). Modern LLMs have evolved to perform various tasks beyond natural language processing. When integrated into mobile agents, these LLMs can interact with the environment and perform tasks without the need for explicitly coded policies or additional model training. This capability leverages the extensive pre-training of LLMs, enabling them to generalize across tasks and adapt to new situations based on their understanding of natural language instructions and contextual cues.

Embodied AI refers to artificial intelligence systems integrated into physical entities, such as mobile robots, that interact with the environment through sensors and actuators (Chrisley, 2003). The integration of LLMs with embodied AI in applications such as autonomous driving (Dorbala et al., 2024) and humanoid robots (Cao, 2024) demonstrates their potential. However, the application of LLMs in controlling mobile robots remains challenging due to issues such as end-to-end control gaps, hallucinations, and path planning inefficiencies. LLMs possess the capability to solve mathematical problems, which directly aids in path planning methods (Gu, 2023).

Path planning and obstacle avoidance are critical for the effective operation of mobile robots, ensuring safe and efficient navigation in dynamic environments (Hewawasam et al., 2022). Coverage path planning is a typical method employed in various research areas, such as ocean seabed mapping (Galceran and Carreras, 2012), terrain reconstruction (Torres et al., 2016), and lawn mowing (Hazem et al., 2021). Traditional path planning methods include algorithms such as A* (Warren, 1993), D* (Ferguson and Stentz, 2005), and potential field methods (Barraquand et al., 1992). Given a global map, a path-planning method can be framed as a mathematical problem solvable by

[a] https://orcid.org/0000-0001-5066-1294
[b] https://orcid.org/0009-0000-5196-8562
[c] https://orcid.org/0000-0003-1359-3813
[d] https://orcid.org/0000-0003-3215-0161

(a) Standard lawnmower   (b) Square spiral

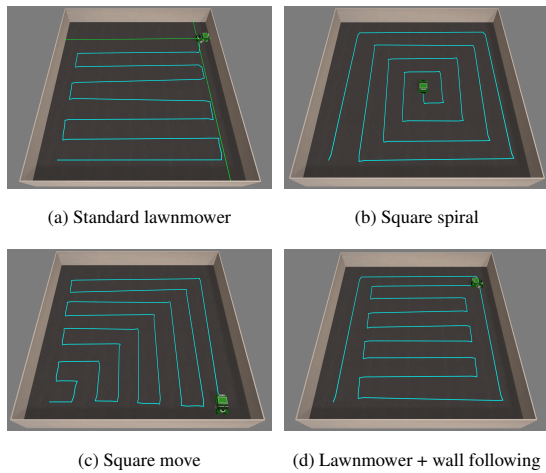(c) Square move   (d) Lawnmower + wall following

Figure 1: Comparison of Path Planning Patterns generated by prompted LLMs.

LLMs. In this context, we simplify some traditional path-planning methods and test LLMs in our mobile robot simulator. LLMs demonstrate their ability to solve mathematical problems collaboratively (Zhang et al., 2024). The **EyeSim VR** is a multiple mobile robot simulator with VR functionality based on game engine Unity 3D that allows experiments with the same unchanged EyeBot programs that run on the real robots (Bräunl, 2020), which is capable of simulating all major functionalities in RoBIOS-7. We have made our experimental simulation platform, **EyeSim**, freely available at https://roblab.org/eyesim/.

This paper presents a multi-layer coverage path planner based on existing multimodal large language models. It involves the static low-dimensional deconstruction of unstructured maps, abstracting spatial relationships into mathematical problems for reasoning and solving by prompted LLMs. The reasoning accuracy of the LLM is enhanced through multi-turn dialogues and multimodal interactions. The inferred results from the LLM are combined with the control interface, enabling the mobile agent to control the robot in real time for path planning. Simulation experiments demonstrate that LLMs possess path-planning capabilities in unstructured static maps.

2 RELATED WORKS

2.1 LLMs in Mobile Robots

Currently, LLMs are involved in various aspects of mobile robots, including code writing, model training, action interpretation, and task planning. LLMs can process new commands and autonomously re-

compose API calls to generate new policy code by chaining classic logic structures and referencing third-party libraries (Liang et al., 2023). LLMs have also been used to automatically generate reward algorithms for training robots to learn tasks such as pen spinning (Ma et al., 2024). PaLM-E, an embodied language model trained on multi-modal sentences combining visual, state estimation, and textual input encodings, demonstrates the versatility and positive transfer across diverse embodied reasoning tasks, observation modalities, and embodiments (Driess et al., 2023). LLMs have shown promise in processing and analyzing massive datasets, enabling them to uncover patterns, forecast future occurrences, and identify abnormal behaviour in a wide range of fields (Su et al., 2024). VELMA is an embodied LLM agent that generates the next action based on a contextual prompt consisting of a verbalized trajectory and visual observations of the environment (Schumann et al., 2024). Sharma et al. propose a method for using natural language sentences to transform cost functions, enabling users to correct goals, update robot motions, and recover from planning errors, demonstrating high success rates in simulated and real-world environments (Sharma et al., 2022).

There is also some research applying LLMs in zero-shot path planning (Chen et al., 2025). The 3P-LLM framework highlights the superiority of the GPT-3.5-turbo chatbot in providing real-time, adaptive, and accurate path-planning algorithms compared to state-of-the-art methods like Rapidly Exploring Random Tree (RRT) and A* in various simulated scenarios (Latif, 2024). Singh et al. describe a programmatic LLM prompt structure that enables the generation of plans functional across different situated environments, robot capabilities, and tasks (Singh et al., 2022). Luo et al. demonstrate the integration of a sampling-based planner, RRT, with a deep network structured according to the parse of a complex command, enabling robots to learn to follow natural language commands in a continuous configuration space (Kuo et al., 2020). ReAct utilizes LLMs to generate interleaved reasoning traces and task-specific actions (Yao et al., 2022). These methods typically use LLMs to replace certain components of mobile robots. The development of a hot-swapping path-planning framework centred around LLMs is still in its early stages.

2.2 Path Planning Method

Path planning for mobile robots involves determining a path from a starting point to a destination on a known static map (Ab Wahab et al., 2024). Obstacle avoidance acts as a protective mechanism for

the robot, enabling interaction with obstacles encountered during movement. Low-level control connects algorithms to different types of system agents, such as UAVs, UGVs, or UUVs (Ali and Israr, 2024). In addition to the A* and D* algorithms mentioned in the previous chapter, path planning algorithms include heuristic optimization methods based on pre-trained weights, such as genetic algorithms (Castillo et al., 2007), particle swarm optimization (Dewang et al., 2018), and deep reinforcement learning (Panov et al., 2018). These pre-trained methods do not directly rely on prior knowledge but utilize data to pretrain weights. The obstacle avoidance problem addresses dynamic obstacles encountered during movement, ensuring the safety of the mobile agent. Mainstream methods include the Artificial Potential Field.

The coverage path planning problem is a branch of path planning problems. Compared with point-to-point path planning, a coverage waypoint list needs to cover the given area as much as possible (Di Franco and Buttazzo, 2016). Classically, decomposing a given map based on topological rules and then applying a repeatable coverage pattern is a common way to solve this issue following the divide-and-conquer algorithm (Palacios-Gasós et al., 2018; Petitjean, 2002; Smith, 1985). In this way, a known map is required to start, whereas the Traveling Salesman Problem (TSP), an optimization problem that seeks to determine the shortest possible route for a salesman to visit a given set of cities exactly once and return to the original city, offers another solution to solve it in a node graph (Hoffman et al., 2013). Figure 1 presented showcases a comparison of four distinct path-planning patterns employed in robotic navigation. The first pattern, labelled as a standard lawnmower (Figure 1a), utilizes a standard back-and-forth sweeping motion to ensure comprehensive coverage of the area. The second pattern, square spiral (Figure 1b), depicts a robot following an inward spiral trajectory, efficiently covering the space in a continuous inward motion. The third pattern, square move (Figure 1c), illustrates a robot navigating in a sequential inward square formation, progressively moving towards the centre. Finally, the lawnmower after wall following (Figure 1d) combines two approaches: initially, the robot adheres to the perimeter of the wall following area, and subsequently, it adopts a lawnmower pattern to cover the remaining interior space. This comparative analysis of path planning strategies highlights the versatility and application-specific advantages of each method in ensuring thorough area coverage in robotic navigation tasks.

3 METHODOLOGY

As depicted in Figure 2, our method is divided into three main sections: global planning, waypoint evaluation, and navigation. In *global planning* phase, a coverage planning task in a given map is decomposed into a cell map, and the additional requirement is designed using natural language with a simplified format to decompose LLM responses. During the *waypoint evaluation* phase, the LLM responses are further evaluated before execution. The theoretical coverage rate and the theoretical shortest path distance are calculated in this phase. Once the desired path passes the evaluation, the planned warpaint list transitions to the navigation phase. In *navigation* phase, the mobile agent simply travels through them one by one and triggers the safety mechanism if the sensor shows a threshold distance between the robot and an unknown obstacle.

3.1 Global Planning

We design a waypoint generation prompt with natural language describing 2D grid maps like a chessboard to simplify the inference difficulty of LLMs. During the global phase, a prompt contains the size of the grid map, current location, and response format. We assume the LLM generates the desired waypoint list with a required format which is a local position sequence separated with a bar sign. In order to evaluate the performance and excitability of the planned path, the desired waypoint list is visualised and calculated in the phrase of waypoint evaluation. Considering the robot's kinematic limitation, we prompt a description of mobile agents including equipped sensors, driving commands, and basic status. We experimented with various settings to describe robot behaviors in conversations with ChatGPT. However, we observed that these changes in description had minimal impact on the output responses. We use OpenAI GPT-4o services (Achiam et al., 2023), a multimodal efficient model for inference and reasoning. The temperature parameter with the range from 0 to 2 is set as 0.6 with our prompt for a consistent planned path. Lower values for temperature result in more consistent outputs, while higher values generate more diverse and creative results.

3.2 Waypoint Evaluation

The response from the LLMs can occasionally be incorrect, leading us to design a waypoint evaluator to mitigate hallucinations. Initially, the desired waypoint list is visualized on a 2D map, providing a clear
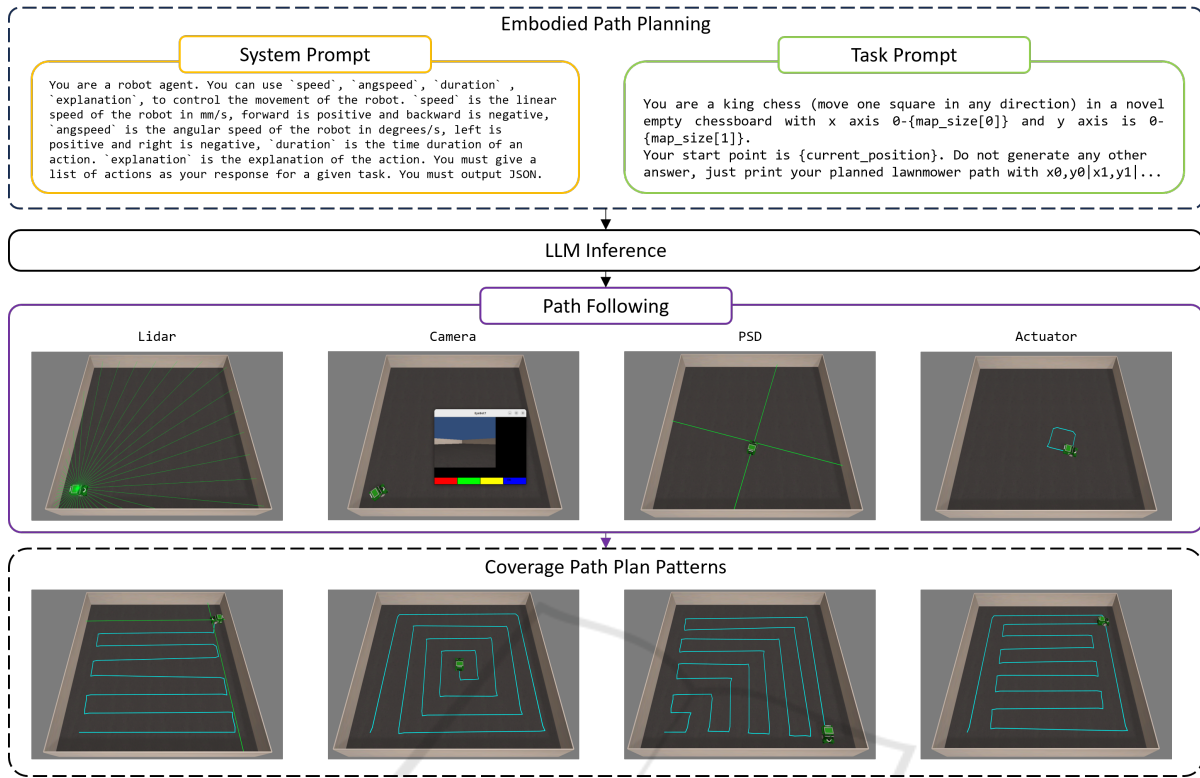
Figure 2: Multi-layer embodied path planning framework.

and precise layout of the proposed route. The shortest path and the number of turns are then calculated mathematically to ensure efficiency and feasibility. Paths that do not meet the required criteria are rejected and not converted into a driving command list. The designed dialogue system initiates as soon as the agent receives the task command and map, continuing until a waypoint list passes the evaluation. This ensures that only optimal routes are considered for execution. Once the mobile agent begins driving, the task cannot be altered, guaranteeing consistency and reliability in task completion.

Algorithm 1 begins by initializing key parameters: the maximum number of iterations N, the evaluation

---

Data: N, θ, p_t, s_0
Result: $\mathcal{P} \leftarrow \{p_t, s_0\}$
initialization;
while $n < N$ **do**
  $\quad$ $\mathbf{W} \leftarrow \Phi(\mathcal{P})$ // LLM inference ;
  $\quad$ $r, \tau \leftarrow \mathcal{E}(\mathbf{W})$;
  $\quad$ **if** $r, \tau > \theta$ **then**
  $\quad\quad$ | $\quad$ $\mathbf{W}$;
  $\quad$ **end**
end

Algorithm 1: Initialization.

threshold θ, the target position p_t, and the starting position s_0. A prompt $\mathcal{P}$ is created, containing the task description and current position, which is then used by the LLM to generate waypoints. The LLM inference function Φ produces a list of waypoints W based on this prompt, taking into account the grid map, current location, and required response format. As the algorithm iterates, it evaluates the generated waypoints using the evaluation function $\mathcal{E}$, which calculates the shortest path r and the number of turns τ. If the calculated path metrics r and τ exceed the predefined threshold θ, the waypoint list is considered feasible and returned. This loop continues until a valid waypoint list is identified or the maximum number of iterations is reached. The algorithm ensures that only optimal routes are considered, thus providing a robust framework for waypoint generation and evaluation. This process incorporates global planning and rigorous waypoint evaluation to leverage LLM capabilities while ensuring safe and reliable path execution for mobile agents.

3.3 Waypoint Navigation

After evaluating the waypoint list, the mobile agent begins to iterate through the waypoints. Due to po-

```
while w_i in W do
    s ← O ;
    s' ← W ;
    a ← Γ(s, s') //Choose a method ;
    Δ ← ||s − s'|| ;
    if Δ < d then
        continue
    end
end
```

Algorithm 2: Execution.

tential sensor errors and the intricacies of the path-following method, it is essential for the mobile agent to appropriately select the following method. Simple waypoint following methods such as the dog curve and turn-and-drive can be employed to navigate the waypoints with a fixed distance. These methods enable the mobile agent to follow the sequence of waypoints with smooth and accurate navigation along the route.

In our approach, we decompose this procedure using a status transform matrix that maps the next driving command based on the current heading, current position, and the next waypoint. This matrix allows for dynamic adjustment and precise control during navigation. Additionally, the designed safety system ensures the execution is safe by preventing collisions with unknown obstacles. This is achieved using a position-sensitive detector and LIDAR beams, which continuously monitor the environment and provide real-time feedback for obstacle avoidance.

A algorithm 2 iterates over each waypoint w_i in the list W. The current position s is updated using odometry data O, and the next waypoint s' is converted from the waypoint list W. The following method is chosen based on the action command a, which is determined by the selected path following function $\Gamma(s, s')$. The distance Δ between the current position s and the next waypoint s' is calculated. If the distance Δ is less than a predefined threshold d, the algorithm continues to the next waypoint.

4 EXPERIMENT

4.1 Implement Details

This framework has been implemented on *EyeBot simulator* (Bräunl, 2023). The EyeBot simulator with virtual reality EyeSim VR is a multiple mobile robot simulator with VR functionality based on game engine Unity 3D that allows experiments with the same unchanged EyeBot programs that run on the real robots. We adjust the environmental values based on

the task map from 5×5 to 11×11. In each map, the mobile agent is at a random starting position and runs the proposed method in 10 episodes, and all performance metrics are averaged. Three large language models are evaluated in the experiment including *gpt-4o*, *gemini-1.5-flash* and *claude-3.5-sonnet* with the same system prompt and default temperature shown in Figure 2.

4.2 Metrics

We referenced the metrics from (Anderson et al., 2018) and (Zhao et al., 2021), including success rate, average distance, and coverage rate. The success rate indicates whether the paths generated by LLMs can cover the designated area. Average distance represents the average path length of the mobile robot, while coverage rate is a metric specific to coverage methods, used to assess the completeness of coverage path planning algorithms.

In traditional navigation evaluation standards, task termination is determined by the distance between the agent and the target point, which is effective for path planning problems with clearly defined start and end points. However, for coverage path planning algorithms, the generated paths do not have a clear endpoint, and the coverage path is autonomously decided by the LLM. Therefore, we have added a coverage rate metric to the comprehensive evaluation standards referenced from the cited sources. Inspired by Success weighted Path Length (SPL) from (Anderson et al., 2018), we will refer to the following measure as CPL, short for **C**overage weighted by (normalized inverse) **P**ath **L**ength:

$$CPL = \frac{1}{N} \sum_{i=1}^{N} \frac{A_i}{\bar{A}_i} \frac{l_i}{max(p_i, l_i)} \qquad (1)$$

where N means the number of test episodes. A_i and $\bar{A}_i$ indicate the area of the coverage path and the area of the mission area, respectively. The ratio of A_i and $\bar{A}_i$ is expressed as the **C**overage **R**ate (CR), which is used to evaluate the completeness of the path. The l_i means the theoretical shortest path distance from the mobile agent start point, and the p_i is the **P**ath **L**ength (PL) of the moving path by the agent.

4.3 Results and Analysis

The performance and time analysis are shown in Table 1 and Table 2. All three models demonstrate the ability to plan a coverage path in a square space with a random start position. However, as the map size increases, the coverage rate decreases by approximately 5% to 10%, though all models maintain

Table 1: Zero-shot coverage path planning performance using multiple LLM services in various environments.

Map Size	GPT-4o			Gemini-1.5			Claude-3.5		
	CPL↑	PL↓	CR↓	CPL	PL	CR	CPL	PL	CR
5×5	0.95	34.2	96.4	0.87	44.5	87.8	**0.99**	37.2	100
7×7	0.86	56.9	86.7	0.81	61.1	82.0	**0.97**	65.9	97.6
11×11	0.78	116	79.7	0.67	124	68.1	**0.98**	147	97.7

Table 2: Preceding and execution time analysis.

Map Size	GPT-4o			Gemini-1.5			Claude-3.5		
	T	T_i	T_d	T	T_i	T_d	T	T_i	T_d
5×5	84.6	2.93	81.5	107	3.20	104	85.9	**2.81**	83.1
7×7	129	4.94	124	125	3.67	121	130	**3.18**	127
11×11	169	9.47	160	157	**5.56**	151	184	5.84	178

a coverage rate above 65%. As shown in Table 1, the model *claude-3.5-sonnet* exhibits the best performance among the three models in terms of coverage rate and weighted path length. Changes in map size do not significantly affect the coverage rate and weighted path for the model *gemini-1.5-flash*. Conversely, the model *gpt-4o* achieves a higher coverage rate with smaller map sizes, but this rate decreases as the map size increases. As the map size grows, the actual path length increases more rapidly than the weighted path length, indicating that the planned paths include repeated visits to the same cells based on the random start position.

The differences in path length are attributed to the coverage rate of the planned path and the mobile agent's hardware capabilities, such as sensors and actuators. Since the evaluation processes locally with a short time cost (less than 300ms), we sum the inference time and the evaluation time as T_i. T and T_d represent the total time spent and the driving part time cost, respectively.

Model *claude-3.5-sonnet* performs best and exhibits the fastest inference time in the experiment, planning fully coverage waypoints in various environments. Model *gpt-4o* shows stable performance across different map sizes, demonstrating robustness and reliability. However, it is noted that the model's performance declines slightly as the map size increases, which could be attributed to the complexity of managing larger spaces and more waypoints. Model *gemini-1.5-flash*, on the other hand, maintains consistent performance regardless of map size, although it occasionally introduces extra line break marks in its responses, which could be due to formatting issues within the LLM's output generation process.

Additionally, the path length differences highlight the varying capabilities of the mobile agents' hardware, such as sensor accuracy and actuator precision,

which directly impact the execution of the planned paths. The evaluation process, which includes both inference and validation, ensures that the paths are not only feasible but also optimized for efficiency.

Overall, the *claude-3.5-sonnet* model excels in both performance and speed, making it ideal for scenarios requiring rapid and thorough coverage. The *gpt-4o* model offers balanced performance with stability across various map sizes, making it a versatile choice. The *gemini-1.5-flash* model, despite minor formatting issues, proves to be reliable with consistent performance. These insights can guide the selection of appropriate LLM services for specific coverage path planning tasks in mobile robotics.

5 CONCLUSIONS

We propose a novel embodied framework for mobile agents, incorporating weighted evaluation metrics for the specific task of coverage path planning. A key factor of the framework is the use of zero-shot prompts to simplify LLM inference during the initial phase. This approach leverages the power of LLMs to generate effective waypoints without the need for extensive training data, thus streamlining the path-planning process. During the navigation phase, we introduced a robust safety mechanism for mobile agents to avoid obstacles. This mechanism ensures that the mobile agents can navigate safely and efficiently in dynamic environments. Our experiments demonstrate that current LLMs have the capability to function as an embodied AI brain within mobile agents for specific tasks, such as area coverage, when guided by appropriately designed prompts.

The competition among LLM companies has significantly advanced the field, freeing researchers from the traditional labelling-training-validation loop in AI research. This shift allows for more focus on inno-

vative applications and real-world deployment of AI technologies. Future research will focus on evaluating path-planning problems in more realistic scenarios and simulation environments. This includes integrating more complex environmental variables and constraints to further evaluate and enhance the robustness of the proposed framework. Additionally, exploring the scalability of LLMs in diverse and larger-scale applications will be crucial in advancing the practical deployment of embodied AI systems in mobile robotics.

REFERENCES

Ab Wahab, M. N., Nazir, A., Khalil, A., Ho, W. J., Akbar, M. F., Noor, M. H. M., and Mohamed, A. S. A. (2024). Improved genetic algorithm for mobile robot path planning in static environments. *Expert Systems with Applications*, 249:123762.

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

Ali, Z. A. and Israr, A. (2024). *Motion Planning for Dynamic Agents*. BoD–Books on Demand.

Anderson, P., Chang, A., Chaplot, D. S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., et al. (2018). On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*.

Barraquand, J., Langlois, B., and Latombe, J.-C. (1992). Numerical potential field techniques for robot path planning. *IEEE transactions on systems, man, and cybernetics*, 22(2):224–241.

Bräunl, T. (2020). *Robot adventures in Python and C*. Springer.

Bräunl, T. (2023). *Mobile Robot Programming: Adventures in Python and C*. Springer International Publishing.

Cao, L. (2024). Ai robots and humanoid ai: Review, perspectives and directions. *arXiv preprint arXiv:2405.15775*.

Castillo, O., Trujillo, L., and Melin, P. (2007). Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. *Soft Computing*, 11:269–279.

Chen, Y., Han, Y., and Li, X. (2025). Fastnav: Fine-tuned adaptive small-language- models trained for multipoint robot navigation. *IEEE Robotics and Automation Letters*, 10(1):390–397.

Chrisley, R. (2003). Embodied artificial intelligence. *Artificial intelligence*, 149(1):131–150.

Dewang, H. S., Mohanty, P. K., and Kundu, S. (2018). A robust path planning for mobile robot using smart particle swarm optimization. *Procedia computer science*, 133:290–297.

Di Franco, C. and Buttazzo, G. (2016). Coverage path planning for uavs photogrammetry with energy and reso-
lution constraints. *Journal of Intelligent & Robotic Systems*, 83:445–462.

Dorbala, V. S., Chowdhury, S., and Manocha, D. (2024). Can llms generate human-like wayfinding instructions? towards platform-agnostic embodied instruction synthesis. *arXiv preprint arXiv:2403.11487*.

Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., and Florence, P. (2023). Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*.

Ferguson, D. and Stentz, A. (2005). The field d* algorithm for improved path planning and replanning in uniform and non-uniform cost environments. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-19*.

Galceran, E. and Carreras, M. (2012). Efficient seabed coverage path planning for asvs and auvs. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 88–93. IEEE.

Gu, S. (2023). Llms as potential brainstorming partners for math and science problems. *arXiv preprint arXiv:2310.10677*.

Hadi, M. U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., Akhtar, N., Wu, J., Mirjalili, S., et al. (2023). A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Hazem, S., Mostafa, M., Mohamed, E., Hesham, M., Mohamed, A., Lotfy, E., Mahmoud, A., and Yacoub, M. (2021). Design and path planning of autonomous solar lawn mower. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 85369, page V001T01A016. American Society of Mechanical Engineers.

Hewawasam, H., Ibrahim, M. Y., and Appuhamillage, G. K. (2022). Past, present and future of path-planning algorithms for mobile robot navigation in dynamic environments. *IEEE Open Journal of the Industrial Electronics Society*, 3:353–365.

Hoffman, K. L., Padberg, M., Rinaldi, G., et al. (2013). Traveling salesman problem. *Encyclopedia of operations research and management science*, 1:1573–1578.

Kuo, Y.-L., Katz, B., and Barbu, A. (2020). Deep compositional robotic planners that follow natural language commands.

Latif, E. (2024). 3p-llm: Probabilistic path planning using large language model for autonomous robot navigation.

Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., and Zeng, A. (2023). Code as policies: Language model programs for embodied control.

Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., and Anandkumar, A.

(2024). Eureka: Human-level reward design via coding large language models.

Palacios-Gasós, J. M., Sagües Blazquiz, C., and Montijano Muñoz, E. (2018). *Multi-Robot Persistent Coverage in Complex Environments*. PhD thesis, PhD thesis, Universidad de Zaragoza.

Panov, A. I., Yakovlev, K. S., and Suvorov, R. (2018). Grid path planning with deep reinforcement learning: Preliminary results. *Procedia computer science*, 123:347–353.

Petitjean, S. (2002). A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys (CSUR)*, 34(2):211–262.

Schumann, R., Zhu, W., Feng, W., Fu, T.-J., Riezler, S., and Wang, W. Y. (2024). Velma: Verbalization embodiment of llm agents for vision and language navigation in street view. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18924–18933.

Sharma, P., Sundaralingam, B., Blukis, V., Paxton, C., Hermans, T., Torralba, A., Andreas, J., and Fox, D. (2022). Correcting robot plans with natural language feedback.

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. (2022). Progprompt: Generating situated robot task plans using large language models.

Smith, D. R. (1985). The design of divide and conquer algorithms. *Science of Computer Programming*, 5:37–58.

Su, J., Jiang, C., Jin, X., Qiao, Y., Xiao, T., Ma, H., Wei, R., Jing, Z., Xu, J., and Lin, J. (2024). Large language models for forecasting and anomaly detection: A systematic literature review.

Torres, M., Pelta, D. A., Verdegay, J. L., and Torres, J. C. (2016). Coverage path planning with unmanned aerial vehicles for 3d terrain reconstruction. *Expert Systems with Applications*, 55:441–451.

Warren, C. W. (1993). Fast path planning using modified a* method. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 662–667. IEEE.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2022). React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.-W., Gao, P., et al. (2024). Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? *arXiv preprint arXiv:2403.14624*.

Zhao, M., Anderson, P., Jain, V., Wang, S., Ku, A., Baldridge, J., and Ie, E. (2021). On the evaluation of vision-and-language navigation instructions.