Interactive Navigation of Quadruped Robots in Challenging Environments using Large Language Models

Kangjie Zhou
College of Engineering
Peking University

Yao Mu
Department of Computer Science
The University of Hong Kong

Pengying Wu
College of Engineering
Peking University

Han Gao
College of Engineering
Peking University

Chang Liu*
College of Engineering
Peking University

Abstract

Robotic navigation in complex environments remains a critical research challenge. Notably, quadrupedal navigation has made significant progress due to the terrain adaptivity and movement dexterity of quadruped robots. However, traditional navigation tasks confine the robot to a predefined free space and focus on obstacle avoidance, limiting their applicability in more challenging environments, such as scenarios lacking feasible paths to the target. We propose an interactive navigation approach that leverages agile quadrupedal movements to adapt to diverse terrains and interact with environments, changing the workspace to tackle challenging navigation tasks in open and complex environments. We present a primitive tree for high-level task planning with large language models (LLMs), facilitating effective reasoning and task decomposition for long-horizon tasks. The tree structure allows for dynamic node addition and pruning, enabling adaptive responses to new observations and enhancing both robustness and real-time performance during navigation. For low-level motion planning, we adopt reinforcement learning to pre-train a skill library containing complex locomotion and interaction behaviors for task execution. Furthermore, we introduce a cognition-based replanning method consisting of the advisor and arborist to react to real-time egocentric observations. The proposed method has been validated in multiple simulated scenarios, demonstrating its effectiveness in diverse scenarios and real-time adaptivity in partially observable conditions.

1 Introduction

Navigation in diverse and complex environments is a pivotal challenge in robotics, necessitating innovative approaches to ensure effective and adaptable decision-making and planning. Recent research has demonstrated the potential of robotic platforms to traverse cluttered environments and navigate to the desired position, utilizing wheeled mobile robots [1, 2] for obstacle avoidance and quadrupedal robots [3, 4] for maneuvering across various terrains. Despite these advancements, conventional navigation strategies primarily focus on collision avoidance within a given free workspace, lacking applicability in more challenging scenarios where no collision-free paths to destinations exist.

To address these limitations, we introduce a novel interactive navigation framework, which utilizes the agility of quadrupedal robots to enable the robot to adapt to diverse terrains and interact with objects to reach impossible goals by traditional navigation methods, while reacting to environmental

*Corresponding author.

Preprint. Under review.

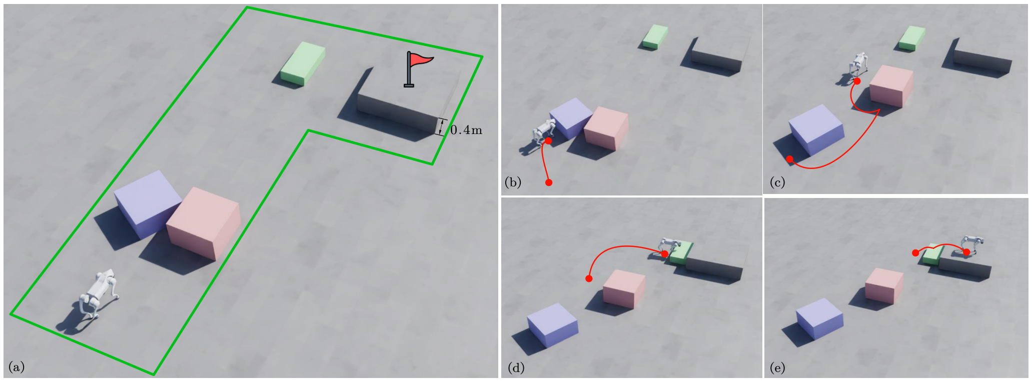

Figure 1: **Interactive navigation task in a challenging environment.** (a) illustrates a task scenario where the robot needs to navigate to the top of a platform with a height of 0.4m, while remaining within the feasible area outlined by the green line. (b) The robot initially attempts to push the blue box but finds it immovable. (c) Subsequently, the robot tries to push the red box and successfully passes through the narrow corridor. (d) Upon discovering no direct path to the platform, the robot assesses the green box and decides to use it as a stair to assist in reaching the platform. (e) By creating this new terrain, the robot successfully navigates to the platform.

feedback swiftly. For example, as displayed in Fig. 1, the robot needs to navigate to the top of a platform with a height of 0.4m and stay in the feasible area outlined by the green line. Obstacles are positioned in the narrow corridor, and the platform is too high for the robot to climb, since we assume the robot can only ascend surfaces where the height difference is less than 0.25m. By leveraging interactable objects within the environment, such as pushing movable objects aside and utilizing boxes as stairs, our approach creates viable pathways through interaction and allows for rapid replans to adapt to new environmental information, thus facilitating the accomplishment of challenging navigation tasks in open and complex environments.

Interactive navigation problem can essentially be regarded as a task and motion planning (TAMP) problem. The high-level task planning involves task decomposition to assist in completing navigation tasks, such as selecting which objects to interact with and determining the sequence of interactions. Low-level motion planning considers how to execute the high-level task by a controller to accomplish the task. Recent work has utilized pre-trained atomic skill libraries combined with large language models (LLMs) for task planning to address TAMP problems [5, 6, 7]. However, these methods require global scene information from an omniscient perspective, which limits their applicability in navigation tasks with partially observable information from an egocentric perspective. Closed-loop task planning can respond to new observations through iterative replanning [8, 9, 10]. However, previous works commonly focus on failure recovery, overlooking the holistic and effective utilization of new information. The primary challenge we aim to address is how to create real-time, effective plans for navigation tasks in open and interactive environments while rapidly responding to newly acquired information.

We propose a hierarchical framework for interactive navigation tasks. We present a primitive tree for high-level task planning using the LLM, which represents task decomposition as a tree structure with primitive skills, enabling promising reasoning and rapid adaptation to new information by modifying the decision tree and enhancing the robustness and effectiveness of task execution. As for low-level control, we develop a comprehensive skill library with reinforcement learning (RL), equipping the robot with the competencies needed for complex locomotion and interaction in intricate environments. Finally, we introduce a cognition-based replanning mechanism that handles new egocentric observations, facilitating informed real-time decision-making. The proposed replanning module consists of an advisor and an arborist, where the advisor analyzes environmental observations to identify potential plan adjustments, while the arborist automatically reconstructs the decision tree for subsequent plan selection. We validate our method across multiple scenarios, showcasing its efficacy in tackling challenging navigation tasks and its capacity for real-time adaptation to novel observations. Through this innovative framework, we aim to extend the boundaries of robotic navigation, offering an effective and computationally efficient solution for interactive navigation in open-world environments.

2 Related Work

2.1 Quadruped Robot Navigation

Quadruped robots have demonstrated agile capabilities and terrain traversability in diverse complex environments in recent researches [11, 12, 13, 14, 15]. Based on the superior agility and adaptivity, quadruped robot navigation has made significant progress through various kinds of approaches, including sampling-based method [16, 17], learning-based method [18, 19, 20]. Recently, benefiting from the powerful understanding and reasoning ability of pre-trained large models, abundant works utilized the LLM and vision-language models (VLM) to understand the semantic environment information for more flexible terrain traverse, so as to tackle navigation tasks [21, 22, 23]. However, current approaches often assume the existence of navigation paths, neglecting the possibility that such paths may not exist in real cluttered scenarios. Zhang et al. [24] presented an interactive navigation framework that instructs robots to navigate in an environment with traversable obstacles with LLMs. However, they aim to adapt to the environment passively rather than actively utilizing interactive objects for task completion. Wu et al. [5] and Xu et al. [25] proposed using LLMs to interact with environments and employ tools to address complex long-horizon tasks with quadruped robots more effectively. However, they assume complete scene descriptions with omniscient information and perform open-loop planning, which is impractical for navigation tasks where the robot obtains real-time egocentric observation incrementally.

2.2 Closed-Loop Task Planning with LLMs

Due to superior performance in language understanding and commonsense reasoning, LLMs are widely adopted to efficiently solve closed-loop task planning, which refers to generating a series of intermediate steps to achieve the specific goal while modifying the plan based on new observations. Many works incorporate an iterative manner for closed-loop task planning, which replans based on the description or analysis of real-time observations [7, 9, 10, 26, 27, 28, 29]. However, these methods typically generate a single plan for execution, which often contains errors or infeasible decomposition due to model uncertainties. This increases the need for replanning and hinders computationally efficient planning. Tree-based structure for LLM reasoning and planning enhances the solution quality by generating multiple plans and select the optimal tree path with state evaluation [8, 30, 31, 32]. However, these methods either lack a replanning mechanism or trigger replanning solely upon encountering failures and concentrate primarily on error correction, overlooking a comprehensive understanding and effective utilization of interactive environments.

3 Method

Our framework consists of three primary components: LLM-based task planning, motion planning with pre-trained skills, and cognition-based replanning, as shown in Fig. 2. With global user instructions and partially observable object information serving as inputs, the task planning module uses the LLM to construct a primitive tree for reasonable task decomposition and convenient adjustments based on new information, enabling the robot to obtain the optimal high-level primitive skeleton. Leveraging a pre-trained skill library consisting of robust locomotion and interaction skills with RL, the motion planning module executes specific low-level actions following the skeleton to complete the task. Finally, a novel cognition-based replanning mechanism interprets new observations from an egocentric perspective to determine whether replanning is necessary, allowing for swift reaction to incremental environmental cognition.

3.1 Task Planning with LLMs

The high-level task planning framework encompasses three LLM-based roles: primitive tree construction, node evaluation, and skeleton selection. Given the user instructions and object description, inspired by [8], we first utilize the LLM to propose multiple potential planning strategies, where each plan consists of several robot's primitive skills. Unlike other approaches that use the LLM to generate a single task plan directly, multiple alternative plans proposed by the LLM are more robust to unexpected errors, such as misunderstandings by the LLM, limited scene observation, and environmental uncertainties. After plans are generated, by merging nodes with identical historical

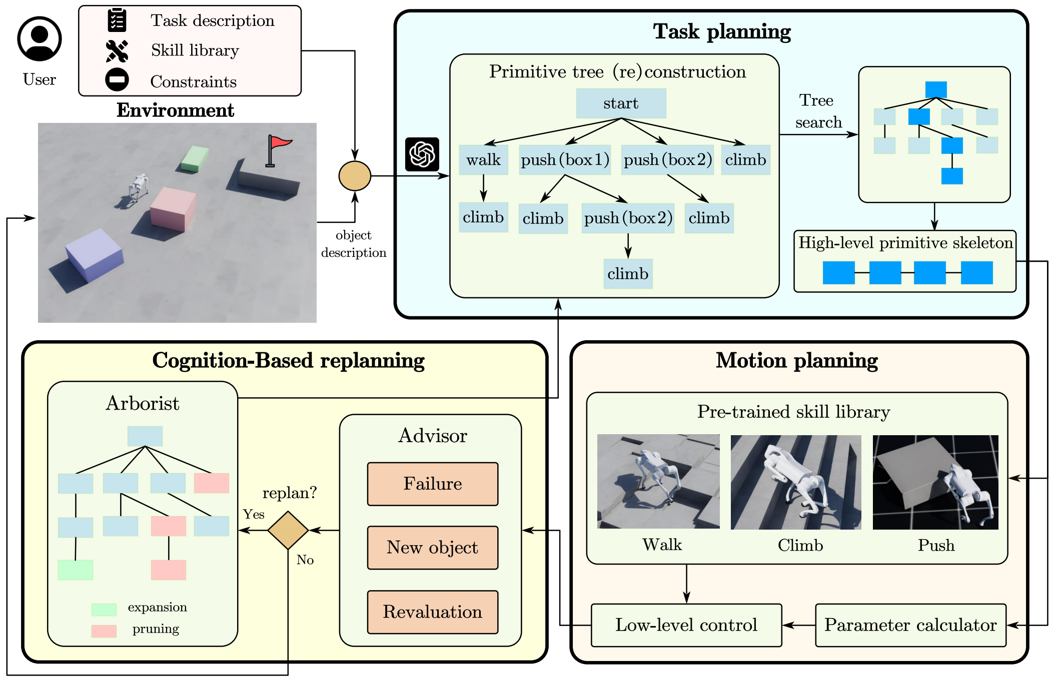

Figure 2: An overview of our interactive navigation approach.

traces, we construct a primitive tree where each node represents a primitive skill from the robot's skill library.

To evaluate the proposed plans, we utilize another LLM as an evaluator to assess the immediate reward $r(n)$ of each node n in the primitive tree, which considers the difficulty of primitive skills and the contribution to the specific task. For example, if the robot interacts with objects like pushing movable obstacles to create a free path for navigation, this skill is essential to completing the task, but the interaction is also difficult to complete. If one skill is non-executable, the subsequent nodes including this node will be deleted to ensure the feasibility of the plan. If a trajectory in the primitive tree reaches the target point at the end, an additional bonus is given to the leaf node as the terminal reward. Finally, we apply a backup method to update the cumulative reward $Q(n)$ of nodes with

$$Q(n) = r(n) + \gamma \sum_{n_c \in children(n)} Q(n_c), \tag{1}$$

where γ is the discounted factor. By iteratively selecting the child nodes with maximal cumulative reward, we derive the optimal high-level primitive skeleton. By representing task planning strategies with a tree structure, the LLM can iteratively reason through potential solutions. This approach also allows for the existence of alternative strategies, enabling rapid replanning upon receiving new environmental information. This aspect will be specifically addressed in cognition-based replanning.

3.2 Motion Planning with Reinforcement Learning

To robustly execute the decomposed high-level tasks, we train a primitive skill library comprising locomotion and interaction skills using RL for low-level motion planning. We first train locomotion policies to develop fundamental movement capabilities on diverse terrains. Building upon the locomotion policies, we employ a hierarchical RL framework to train more complex strategies, such as pushing policy for moving objects to designated positions and walking policy for navigation with collision avoidance. All these skills are trained using PPO [33] in the IsaacLab simulation environment [34].

For locomotion skill training, we formulate the training procedure as a velocity-tracking pattern similar to [35], which encourages robots to follow the linear and angular velocity command while adapting to arbitrary terrain. Furthermore, we employ curriculum learning to incrementally increase terrain difficulty, thereby developing a robust locomotion policy. With pre-trained locomotion policy, we train mid-term strategies with the hierarchical RL method as a pose-tracking task, which generates velocity commands for the pre-trained locomotion policy to follow. Specifically, the pushing policy aims to push objects to the goal pose, and the walking policy navigates the robot to the target pose while realizing obstacle avoidance. After training these skills, we encapsulate them into APIs for high-level task planning. Based on the calculated parameters, we allow the execution of corresponding tasks with low-level skills.

3.3 Cognition-Based Replanning

Unlike previous LLM-based planning approaches that assume omniscient scene description, navigation tasks typically involve unknown environments, where the robot continuously acquires new observations with an egocentric perspective. For this reason, we develop a cognition-based replanning mechanism to analyze the new observations and efficiently replan based on new environmental cognition. This mechanism consists of two LLM-based roles: a LLM advisor and a LLM arborist.

The LLM advisor analyzes new environmental information and the current plan to determine whether replanning is necessary based on these cognition updates. We let the advisor conduct environmental cognition in three aspects, including failure, new objects, and revaluation. Failure indicates that the current plan has encountered some errors, new objects represent the discovery of a new interactive object that may be useful, and revaluation means reassessing the plan based on updated information about the environment. Unlike previous works that only replan based on failure, the other two types of cognition are also important to perform tasks with an incrementally updated understanding of the environment. For example, when you discover a new object, you need to analyze its geometric and semantic information and decide to utilize it if it is more helpful to your task. As for revaluation, the robot's understanding of the object information under observation uncertainties will be updated, which will affect the evaluation of the current plan. Based on these three types of cognition, the advisor will decide whether to replan and replan suggestions.

If the advisor thinks replan is necessary, the suggestion will be sent to the LLM arborist, which modifies the primitive tree structure, such as adding nodes for new objects and pruning nodes for failure occurrence. With a tree structure, it is convenient to add or prune nodes and bring computationally efficient replanning procedure, making real-time robotics task execution possible. Then backup is conducted within the new tree and a new plan skeleton is selected to perform with pre-trained skills. Through this replanning approach, the robot can respond to new environmental information in real-time, adjusting its planning strategies accordingly. By analyzing new observations, the robot can assess their impact on task completion and decide whether adjustments to the current plan are required. This process helps avoid the computational cost of unnecessary replanning.

4 Simulation

4.1 Quantitative Simulation on Task Planning

To demonstrate the effectiveness of the proposed high-level task planning approach, we first conduct quantitative simulations to compare the performance of different baselines in task planning during navigation. We design four different scenarios in the same simulation environment created in IsaacLab, which is shown in Fig. 3. The goal for the quadruped robot is to navigate to the platform with 0.4m height, and we assume that the robot can only climb to the surface within 0.25m height difference without considering the advanced parkour skills shown in previous work. The robot must reason how to use the objects in the environment to create a feasible plan to reach the high platform.

To focus on the comparison of the quality of high-level task planning results, we assume a perfect low-level motion controller and directly set the robot and environment state in the simulation environment based on the high-level plan and related parameters. We artificially designed the feasibility judgment conditions for each skill, such as the robot can only climb objects of limited height, push objects in the same plane as the robot, and so on. After the reset, the robot updates the observation information from the current new state. We use a simplified perception module that obtains pre-defined object information with a limited sensing range. When the object appears in the sensing range of the robot and is not occluded, the simulator will give the robot object information, substituting using VLM for object description.

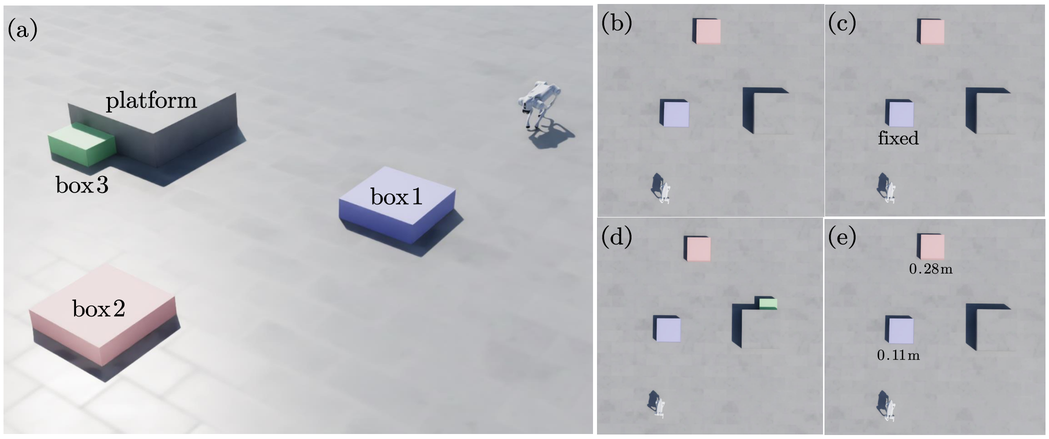

Figure 3: **Simulation environment for quantitative comparison of task planning.** (a) demonstrates the simulation environment consists of different boxes and the target platform. The height of the blue, red, and green boxes are 0.21m, 0.18m, and 0.20m, respectively. (b)-(e) illustrate four scenarios for interactive navigation tasks.

Table 1: Comparison of different methods on interactive navigation tasks.

Scenario	Method	SR	Plan length	Replan times	Cost (10^{-2}\$)
Scenario 1	w/o replan	7/10	4.3 ± 0.7	-	**2.25** ± 0.38
	Step-based replan	**10/10**	**3.9** ± 0.5	2.9 ± 0.5	8.66 ± 1.06
	Failure-based replan	9/10	5.9 ± 2.0	**0.3** ± 0.5	3.16 ± 1.57
	Cognition-based replan	8/10	5.0 ± 1.4	**0.3** ± 0.6	3.12 ± 1.88
	Ours	**10/10**	4.1 ± 0.7	**0.3** ± 0.6	10.69 ± 0.55
Scenario 2	w/o replan	2/10	**4.0** ± 0.0	-	**2.24** ± 0.33
	Step-based replan	**10/10**	4.8 ± 0.5	3.8 ± 0.5	10.47 ± 1.26
	Failure-based replan	9/10	6.0 ± 1.9	0.9 ± 0.5	4.29 ± 1.57
	Cognition-based replan	9/10	6.0 ± 2.0	0.8 ± 0.6	4.34 ± 1.81
	Ours	9/10	4.8 ± 0.9	**0.7** ± 0.5	11.11 ± 1.02
Scenario 3	w/o replan	7/10	4.1 ± 0.8	-	**2.23** ± 0.33
	Step-based replan	**10/10**	**3.7** ± 0.3	2.7 ± 0.3	8.08 ± 1.10
	Failure-based replan	9/10	5.7 ± 2.2	**0.2** ± 0.4	2.71 ± 1.28
	Cognition-based replan	**10/10**	3.9 ± 0.5	1.0 ± 0.0	4.88 ± 0.28
	Ours	**10/10**	4.3 ± 0.4	1.0 ± 0.0	11.59 ± 0.63
Scenario 4	w/o replan	1/10	**6.0** ± 0.0	-	**2.26** ± 0.26
	Step-based replan	4/10	7.0 ± 1.4	7.8 ± 1.8	19.61 ± 4.17
	Failure-based replan	2/10	7.0 ± 1.0	1.3 ± 0.6	5.16 ± 1.34
	Cognition-based replan	4/10	7.8 ± 1.1	1.4 ± 0.7	5.91 ± 1.54
	Ours	**7/10**	6.4 ± 1.0	**0.3** ± 0.5	11.61 ± 0.77

We compare the proposed method to four baselines: w/o replan, step-based replan, failure-based replan, and cognition-based replan. All these baselines only use the LLM to generate one plan and execute it directly. w/o replan refers to no replan during execution. Step-based replan represents the robot performing replan after each step based on new observations. Failure-based replan is a common replanning scheme for closed-loop task planning, which replans when encountering errors during task completion. Cognition-based replan is essentially a baseline for ablation study, which only removes the tree structure for task planning.

We use four metrics to evaluate the performance of different methods: success rate (SR), plan length, replan times, and token expenditure (Cost). SR refers to whether the robot reaches the top of the platform within ten steps. Plan length measures the number of executed steps to complete the task when the task is successful. Replan time refers to the replanning times for each method. Cost is the

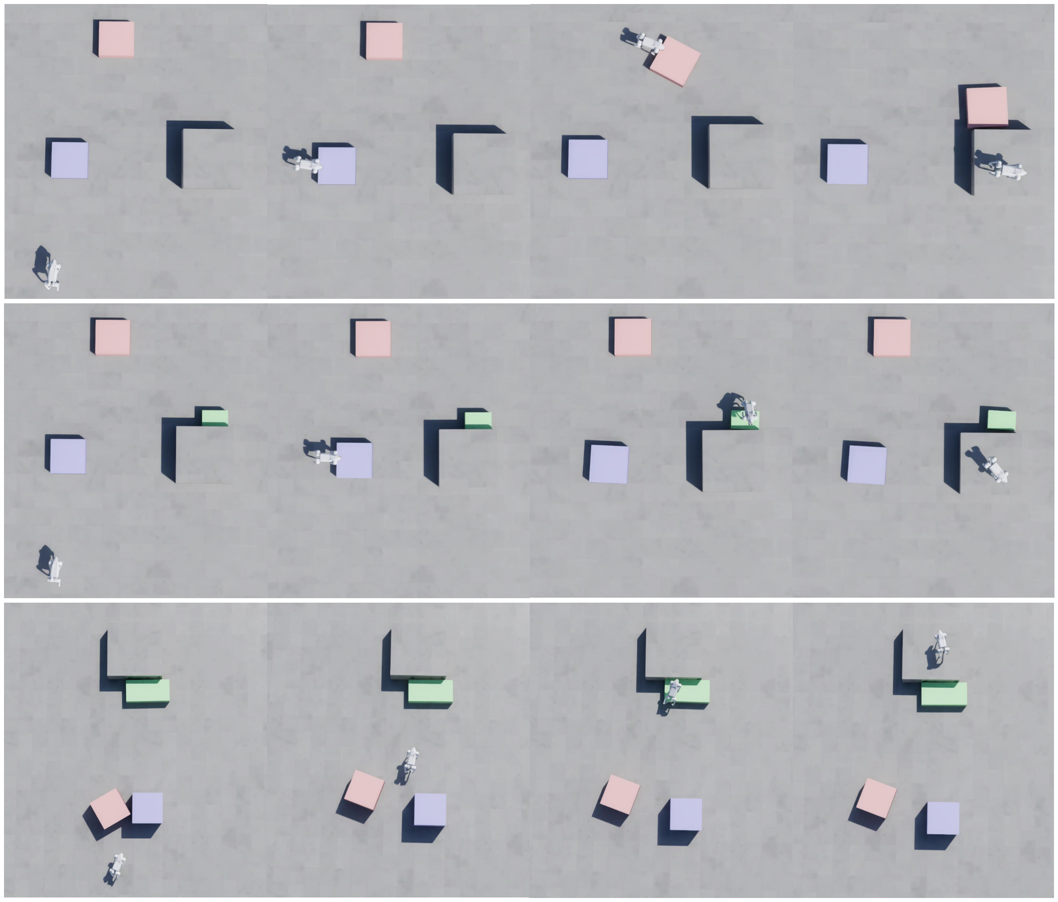

Figure 4: **Qualitative simulation for interactive navigation tasks.** Each row depicts a distinct scenario: the first features an immovable blue box, the second includes an initially invisible green box, and the third simulates high obstacles blocking the path.

economic cost of using LLM based on the pricing provided by OpenAI[2]. We use GPT-4o [36] as the LLM and conduct 10 runs over each scenario and average the evaluation metrics.

As shown in Table 1, the proposed method exhibits distinct advantages over baselines in different scenarios. Overall, our method demonstrates a high success rate and reduced plan length across four scenarios. Compared to the step-based method that requires replanning after each step, our method replans adaptively based on environmental information to reduce unnecessary replans and enhance planning efficiency while ensuring successful scenario completion. Although failure-based replanning can respond to failures and perform error correction, it lacks the intelligent understanding of new objects. For instance, in scenario 3, the robot fails to recognize that it can directly climb onto the platform via the green box upon detecting the new box, resulting in increased plan length. In contrast, our cognition-based replanning approach enables the robot to comprehend and utilize new objects effectively, thus avoiding the time-consuming step of executing a pushing policy. Furthermore, compared to three replan methods in baselines that generate a single plan for execution, our tree-structured planning approach demonstrates more effective reasoning for long-horizon tasks and generates more robust plans. As depicted in scenario 4, our method shows significant advantages in success rates and requires fewer replanning attempts. Additionally, the arborist allows for convenient modifications to the tree structure based on advisor recommendations after analyzing new information, thereby enhancing replanning efficiency.

4.2 Qualitative Simulation for Interactive Navigation

As this research is on process currently, we discuss some qualitative results for complete interactive navigation tasks including high-level task planning and low-level task execution. We display qualitative results in three scenarios in Fig. 4, showcasing how the robot performs planning with an egocentric perspective under partially observable information, how the robot adjusts the current plan based on the real-time feedback, and finally efficiently and adaptively completes navigation tasks.

In Scenario 1, there are two boxes and the blue box is immovable. Initially, the robot observes both boxes and considers pushing one or both as an intermediate step to complete the task. Since the blue box is closer, the robot first decides to push it towards the platform. However, upon discovering that the blue box is immovable, the advisor analyzes the failure situation, and the arborist prunes the node involving pushing the blue box. The robot then replans and decides to push the red box to complete the task.

In Scenario 2, there are three movable boxes, but the robot is initially unaware of the green box due to occlusion. Similar to the high-level planning procedure in scenario 1, the robot first decides to push the blue box. During this process, the robot discovers the green box and, through advisor analysis, realizes that the green box is near the platform and can be used to climb onto it directly. Consequently, the robot stops pushing the blue box and moves directly to the green box to climb onto the platform.

In Scenario 3, the robot is blocked in a narrow passage by two boxes that are too tall, and it needs to consider whether these boxes can help it climb onto the platform, despite their height being too high. The robot first decides to push one of the boxes to create a viable path and successfully moves the red box. Subsequently, it perceives a green box near the platform that can serve as a step, so it directly climbs onto the platform using the green box.

In summary, our approach can accomplish interactive navigation tasks in various scenarios while quickly reacting to real-time environmental cognition to improve task efficiency. With the adaptivity to partially observable environments, including unknown obstacles and their properties, the proposed method can effectively address challenging task planning in open-world complex environments.

5 Limitation Discussion

While our approach has demonstrated promising performance in interactive navigation tasks, several limitations remain to be addressed. Firstly, the success rate in highly challenging scenarios needs improvement. This can be attributed to two main factors: at the high level, the LLM fails to propose a reasonable plan, making it difficult to achieve an optimal solution in subsequent steps. At the low level, the lack of robustness in the pre-trained skills and the integration between different skills affects successful task execution. Secondly, current work simplifies the perception model and involves relatively simple task scenarios. However, by leveraging powerful VLM for scene understanding, we believe our framework can better adapt to open-world environments and make more informed decisions. Additionally, the skill library for the quadruped robot is relatively limited. By incorporating a robotic arm like [37, 38], our proposed framework could accomplish more complex tasks, such as mobile manipulation. Finally, although we have demonstrated improved performance in navigation tasks within partially observable environments, the tree construction incurs significant time and token costs. Albeit the proposed cognition-based replanning mechanism can quickly respond to new environmental information, overall computational efficiency needs enhancement, potentially through simplification operation in high-level task planning.

6 Conclusion

We introduce an LLM-based interactive navigation approach using a quadruped robot, which actively understands and utilizes interactive objects in partially observable environments to tackle challenging navigation tasks where feasible paths may not exist. Our hierarchical framework leverages the LLM for high-level task planning and employs pre-trained skills with RL for low-level motion planning. With tree-structured task decomposition, our method not only enhances the planning quality for long-horizon complex tasks but also allows for convenient modifications in specific tree parts when

[2]https://openai.com/pricing

real-time new observations are obtained. During task execution, the quadruped robot uses pre-trained agile locomotion and interaction skills to engage with the environment and create feasible paths to accomplish the task. We present a cognition-based replanning method for the intelligent understanding of incremental information, enabling robots to quickly adapt to open and complex environments and efficiently complete tasks. This instructive framework has the potential to enhance robotic navigation capabilities, offering an effective and computationally efficient solution for interactive navigation in open-world environments.

7 Acknowledgment

This work was supported in part by the Beijing Nova Program (20220484056, 20240484498) and the National Natural Science Foundation of China (62203018).

References

[1] Dhruv Shah, Michael Robert Equi, Błażej Osiński, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. In *Conference on Robot Learning*, pages 2683–2699. PMLR, 2023.

[2] Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. Voronav: Voronoi-based zero-shot object navigation with large language model. *arXiv preprint arXiv:2401.02695*, 2024.

[3] David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

[4] Lei Han, Qingxu Zhu, Jiapeng Sheng, Chong Zhang, Tingguang Li, Yizheng Zhang, He Zhang, Yuzhen Liu, Cheng Zhou, Rui Zhao, et al. Lifelike agility and play in quadrupedal robots using reinforcement learning and generative pre-trained models. *Nature Machine Intelligence*, pages 1–12, 2024.

[5] Yutao Ouyang, Jinhan Li, Yunfei Li, Zhongyu Li, Chao Yu, Koushil Sreenath, and Yi Wu. Long-horizon locomotion and manipulation on a quadrupedal robot with large language models. *arXiv preprint arXiv:2404.05291*, 2024.

[6] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[7] Peiyuan Zhi, Zhiyuan Zhang, Muzhi Han, Zeyu Zhang, Zhitian Li, Ziyuan Jiao, Baoxiong Jia, and Siyuan Huang. Closed-loop open-vocabulary mobile manipulation with gpt-4v. *arXiv preprint arXiv:2404.10220*, 2024.

[8] Mengkang Hu, Yao Mu, Xinmiao Yu, Mingyu Ding, Shiguang Wu, Wenqi Shao, Qiguang Chen, Bin Wang, Yu Qiao, and Ping Luo. Tree-planner: Efficient close-loop task planning with large language models. *arXiv preprint arXiv:2310.08582*, 2023.

[9] Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*, 2023.

[10] Marta Skreta, Zihan Zhou, Jia Lin Yuan, Kourosh Darvish, Alán Aspuru-Guzik, and Animesh Garg. Replan: Robotic replanning with perception and language models. *arXiv preprint arXiv:2401.04157*, 2024.

[11] Siddhant Gangapurwala, Mathieu Geisert, Romeo Orsolino, Maurice Fallon, and Ioannis Havoutis. Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control. *IEEE Transactions on Robotics*, 38(5):2908–2927, 2022.

[12] Tairan He, Chong Zhang, Wenli Xiao, Guanqi He, Changliu Liu, and Guanya Shi. Agile but safe: Learning collision-free high-speed legged locomotion. *arXiv preprint arXiv:2401.17583*, 2024.

[13] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023.

[14] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450. IEEE, 2024.

[15] Xuxin Cheng, Ashish Kumar, and Deepak Pathak. Legs as manipulator: Pushing quadrupedal agility beyond locomotion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5106–5112. IEEE, 2023.

[16] Lorenz Wellhausen and Marco Hutter. Rough terrain navigation for legged robots using reachability planning and template learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6914–6921. IEEE, 2021.

[17] Lorenz Wellhausen and Marco Hutter. Artplanner: Robust legged robot navigation in the field. *arXiv preprint arXiv:2303.01420*, 2023.

[18] Nikita Rudin, David Hoeller, Marko Bjelonic, and Marco Hutter. Advanced skills by learning locomotion and local navigation end-to-end. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2497–2503. IEEE, 2022.

[19] Shiyu Feng, Ziyi Zhou, Justin S Smith, Max Asselmeier, Ye Zhao, and Patricio A Vela. Gpf-bg: A hierarchical vision-based planning framework for safe quadrupedal navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1968–1975. IEEE, 2023.

[20] David Hoeller, Lorenz Wellhausen, Farbod Farshidian, and Marco Hutter. Learning a state representation and navigation in cluttered and dynamic environments. *IEEE Robotics and Automation Letters*, 6(3):5081–5088, 2021.

[21] Ye Wang, Yuting Mei, Sipeng Zheng, and Qin Jin. Quadrupedgpt: Towards a versatile quadruped agent in open-ended worlds. *arXiv preprint arXiv:2406.16578*, 2024.

[22] Shaoting Zhu, Derun Li, Yong Liu, Ningyi Xu, and Hang Zhao. Cross anything: General quadruped robot navigation through complex terrains. *arXiv preprint arXiv:2407.16412*, 2024.

[23] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023.

[24] Zhen Zhang, Anran Lin, Chun Wai Wong, Xiangyu Chu, Qi Dou, and KW Samuel Au. Interactive navigation in environments with traversable obstacles using large language and vision-language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7867–7873. IEEE, 2024.

[25] Mengdi Xu, Peide Huang, Wenhao Yu, Shiqi Liu, Xilun Zhang, Yaru Niu, Tingnan Zhang, Fei Xia, Jie Tan, and Ding Zhao. Creative robot tool use with large language models. *arXiv preprint arXiv:2310.13065*, 2023.

[26] Zeyi Liu, Arpit Bahety, and Shuran Song. Reflect: Summarizing robot experiences for failure explanation and correction. *arXiv preprint arXiv:2306.15724*, 2023.

[27] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[28] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR, 2022.

[29] Shu Wang, Muzhi Han, Ziyuan Jiao, Zeyu Zhang, Ying Nian Wu, Song-Chun Zhu, and Hangxin Liu. Llmˆ 3: Large language model-based task and motion planning with motion failure reasoning. *arXiv preprint arXiv:2403.11552*, 2024.

[30] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[31] 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.

[32] Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

[33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[34] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023.

[35] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[36] GPT-4o. https://openai.com/index/hello-gpt-4o/.

[37] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023.

[38] Nishanth Kumar, Tom Silver, Willie McClinton, Linfeng Zhao, Stephen Proulx, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Jennifer Barry. Practice makes perfect: Planning to learn skill parameter policies. *arXiv preprint arXiv:2402.15025*, 2024.