Instruction-Augmented Long-Horizon Planning: Embedding Grounding Mechanisms in Embodied Mobile Manipulation

Instruction-Augmented Long-Horizon Planning: Embedding Grounding Mechanisms in Embodied Mobile Manipulation

Fangyuan Wang[1,2], **Shipeng Lyu**[1, 2], **Peng Zhou**[3], **Anqing Duan**[4], **Guodong Guo**[2*], **David Navarro-Alarcon**[1*]

[1] The Hong Kong Polytechnic University [2] Ningbo Institute of Digital Twin, Eastern Institute of Technology
[3] Great Bay University [4] Mohamed Bin Zayed University of Artificial Intelligence
gdguo@eitech.edu.cn, dnavar@polyu.edu.hk

Abstract

Enabling humanoid robots to perform long-horizon mobile manipulation planning in real-world environments based on embodied perception and comprehension abilities has been a longstanding challenge. With the recent rise of large language models (LLMs), there has been a notable increase in the development of LLM-based planners. These approaches either utilize human-provided textual representations of the real world or heavily depend on prompt engineering to extract such representations, lacking the capability to quantitatively understand the environment, such as determining the feasibility of manipulating objects. To address these limitations, we present the Instruction-Augmented Long-Horizon Planning (IALP) system, a novel framework that employs LLMs to generate feasible and optimal actions based on real-time sensor feedback, including grounded knowledge of the environment, in a closed-loop interaction. Distinct from prior works, our approach augments user instructions into PDDL problems by leveraging both the abstract reasoning capabilities of LLMs and grounding mechanisms. By conducting various real-world long-horizon tasks, each consisting of seven distinct manipulatory skills, our results demonstrate that the IALP system can efficiently solve these tasks with an average success rate exceeding 80%. Our proposed method can operate as a high-level planner, equipping robots with substantial autonomy in unstructured environments through the utilization of multi-modal sensor inputs.

Website — https://nicehiro.github.io/IALP

Introduction

Embodied decision-making has been regarded as a critical role for humanoid robots tasked with executing long-horizon mobile manipulation tasks, which require prolonged navigation of the environment and interaction with objects. The recent advancement of large language models (LLMs) has significantly enhanced their ability to perceive, comprehend, and plan actions with their surroundings, thus promoting the successful completion of long-horizon mobile manipulation tasks. However, most of the existing research lacks the grounding knowledge of the real-world environment, such as spatial positioning and feasible grasping poses. They either rely on simulated settings or predefined contextual descriptions, with few investigations into real-world understanding and representation through the use of LLMs. This deficiency in grounded understanding impedes robots from properly interacting with complex objects/structures, thus, impairing their application in unknown environments. Given these challenges, an imperative question arises: Can we effectively extract and utilize environmental representations, both abstract and grounded, to empower humanoid robots to plan long-horizon tasks in real-world environments?

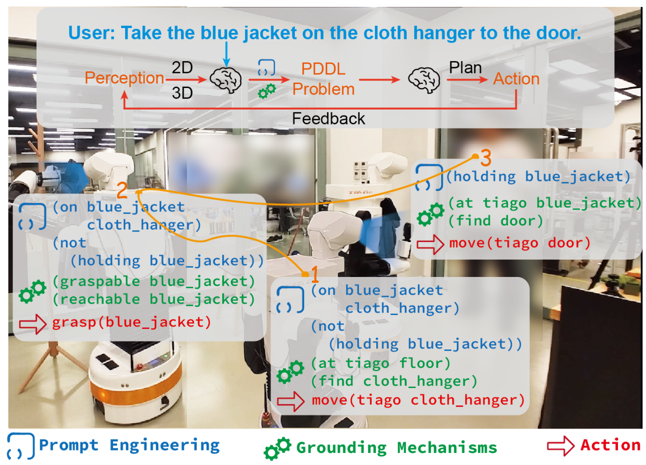

Figure 1: The IALP system leverages the reasoning capabilities of LLMs and grounding mechanisms to enrich the task representation, enabling plan feasible and optimal actions in the real world.

An approach to providing robots with embodied intelligence involves utilizing off-the-shelf foundation models such as GPT-4 (Achiam et al. 2023), CLIP (Radford et al. 2021), and SAM (Kirillov et al. 2023), thus enabling them to do long-horizon planning. Classical methods (Wake et al. 2023; Lin et al. 2023; Huang et al. 2023; Ahn et al. 2024) typically operate within simulation environments or assume the availability of textual descriptions of the environment, heavily relying on the reasoning capabilities of visual language models (VLMs) through prompt engineering. However, these methods face significant limitations in real-world

*Corresponding authors

environments where extracting key features, such as the precise position of objects, from the real world is challenging.

Furthermore, mobile manipulation in robots has gained increasing popularity. Previous studies have employed both models trained via transformer-based structures (Brohan et al. 2022, 2023) and training-free approaches (Liu et al. 2024; Wang et al. 2024; Ahn et al. 2024). Nonetheless, these approaches either lack long-horizon planning capabilities (Brohan et al. 2022, 2023; Liu et al. 2024) and/or interactive planning abilities that enable the robot to adjust its behavior based on feedback (Liu et al. 2024; Wang et al. 2024; Ahn et al. 2024). We anticipate leveraging the implicit world knowledge of LLMs and grounding mechanisms, such as voxel map building and grasping pose generation, to decode task-relevant information from the environment. This information can be transformed into structured descriptions that are recognized by the LLM planner, enabling the execution of long-horizon tasks in a closed-loop manner.

To this end, we introduce the **I**nstruction **A**ugmented **L**ong-Horizon **P**lanning (**IALP**) system: a flexible and extendable framework to solve long-horizon tasks in real-world environments. The IALP system (i) augments the user instruction with the task-related knowledge extracted from promptable and grounding mechanisms-based predicates using Planning Domain Description Language (PDDL) (Fox and Long 2003; Garrett, Lozano-Pérez, and Kaelbling 2020), and (ii) generates feasible and optimal actions to complete long-horizon tasks in a *close-loop* manner (see Figure 1). Specifically, our new method first augments the user's instruction by generating a PDDL problem based on the user's input and the robot's perception, which includes both 2D and 3D data. Then, the system employs the LLM planner to interactively generate feasible and optimal actions, encompassing both navigation and manipulation, to accomplish the entire task. The system's extendability allows for the design and inclusion of additional predicates, both promptable and grounding mechanisms, to meet various needs. The contributions are summarized as follows:

- We introduce IALP, an adaptive framework that utilizes both abstract reasoning abilities (from LLMs) and grounding mechanisms to extract task-relevant representations from the environment.
- We propose a method that enables robots to generate feasible and optimal actions based on real-time feedback in a closed-loop interaction with the environment.
- We deploy the entire system in real-world scenarios across several long-horizon tasks, comprising skills such as moving, picking, placing, and more. The results demonstrate that our algorithm achieves over 80% planning accuracy.

Related Works

Pretrained Foundation Models for Task Planning

Preliminary studies (Kambhampati et al. 2023; Valmeekam et al. 2023; Guan et al. 2023; Ao et al. 2024) have demonstrated that off-the-shelf large language models (LLMs) have the capability to solve long-horizon tasks as effective high-level planners. Traditional planning systems (e.g., those based on PDDL) employ efficient search algorithms to identify correct or even optimal plans, but are constrained by predefined structured task descriptions. Recent studies (Liu et al. 2023; Silver et al. 2022) have leveraged the capabilities of LLMs and PDDL planners, where LLMs are employed to generate structured task descriptions, and PDDL planners subsequently used to find optimal solutions.

With the integration of multimodal inputs, such as visual and auditory data, several studies (Geng et al. 2023; Zhang et al. 2024; Huang et al. 2023) have heavily relied on the reasoning and in-context learning capabilities of visual language models (VLMs) to extract spatial relations and object attributes for planning purposes. However, we argue that while VLMs excel at reasoning about the spatial relations in 2D images, they are limited in their ability to extract precise spatial information, such as an object's position and pose within the environment. Rather than relying on predefined planning priors, we leverage not only VLMs' implicit world knowledge but also grounding mechanisms to extract representations that encode task-relevant information, such as checking if the generated grasping pose is feasible to reach.

Mobile Manipulation in Open Worlds

Mobile manipulation and navigation in open-world environments necessitate open-vocabulary capabilities and strong adaptive skills in robots. Despite recent advances in pretraining foundation models, which have spurred increased research interest, this remains an unresolved challenge (Yenamandra et al. 2023). To address this issue, (Brohan et al. 2022, 2023) propose end-to-end pipelines by training transformer-based vision-language models. On the other hand, (Liu et al. 2024; Wang et al. 2024; Ahn et al. 2024) introduce training-free approaches utilizing pretrained foundation models. OK-Robot (Liu et al. 2024) can perform pick-and-place tasks in open-world settings without any training by leveraging state-of-the-art vision-language models (VLMs) such as CLIP (Radford et al. 2021) and OWL-ViT (Minderer et al. 2022) for object detection, navigation and manipulation. However, it lacks the ability to reason about long-horizon tasks and is limited to pick-and-place operations. In contrast, (Wang et al. 2024; Ahn et al. 2024) use distilled spatial geometry and 2D observations with VLMs to guide the morphology selection of robots. Their proposed planning frameworks are one-shot, meaning they plan the entire task at the beginning without receiving feedback and use LLMs only for morphology selection rather than for generating primitive actions. Prior studies either lack long-horizon planning capabilities (Brohan et al. 2022, 2023; Liu et al. 2024) or interactive planning abilities (Wang et al. 2024; Ahn et al. 2024).

Our work constructs the PDDL problem description based on the observation of the current state using embodied perception. These textual descriptions are then fed into LLMs. After executing the generated action, the current observation changes, prompting the reconstruction of the PDDL problem. This process is iteratively executed, enabling our method to achieve both long-horizon planning and interactive planning capabilities.

Problem Formulation

We aim to address long-horizon mobile manipulation tasks that necessitate both symbolic and geometric reasoning from a natural language instruction i and the initial state of the environment. We assume an open-world setting, wherein the robot operates without prior knowledge of task-relevant objects or other ground truth information. Instead, it relies solely on data acquired through its embodied perception system and a prebuilt voxel map for navigation.

We utilize an LLM planner with a library of skills $L^{\psi} = \{\psi_1, \psi_2, \cdots, \psi_n\}$. Each skill ψ is associated with a single-timestep Markov Decision Process (MDP) (Sutton and Barto 2018; Lin et al. 2023), which is denoted as $(\mathcal{S}, \mathcal{A}, R, T, \rho)$. $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition function, R is the reward function, and ρ is the distribution of the initial state. Each action a is a skill $\psi \in L^{\psi}$, which is a parameterized primitive $\phi(\alpha)$. When an action is executed, a parameter α is sampled from the LLM planner and fed to its primitive $\phi(\alpha)$, which executes a series of motor commands on the robot. If the action succeeds, it receives a binary reward of r.

The Planning Objective

The objective is to find a sequence of actions $\{a_1, \cdots, a_H\}$, denoted as $a_{1:H}$, that can achieve the given instruction i. We use an interactive planning method that considers action feedback and proposes several candidates at each time step. Thus, the objective at time step $t \in \{1 : H\}$ can be expressed as the joint probability of skill sequence $a_{t:H}$ and binary rewards $r_{t:H}$ given the instruction i and the current state s_t:

$$\begin{aligned} p(a_{t:H}, r_{t:H}|i, s_t) &= p(a_{t:H} \mid i, s_t) p(r_{t:H} \mid i, s_t, a_{t:H}) \\ &= \prod_{x=t}^{H} p(a_x \mid i, s_{t:x}, a_{t:x-1}) \prod_{x=t}^{H} p(r_x \mid s_{t:x}, a_{t:x}), \end{aligned} \tag{1}$$

where the former term represents the probability that the action sequence $a_{t:H}$ will satisfy the instruction i from a symbolic perspective. We define the action sequence **optimality** score $S_{op} = \prod_{x=t}^{H} p(a_x \mid i, s_{t:x}, a_{t:x-1})$, where the probability of the next skill a_x is considered in terms of the current state s_t, predicted $s_{t+1:x}$ and planned $a_{t:x-1}$ and is thus independent of prior rewards $r_{1:x-1}$. It captures the utility of the action sequence $a_{t:H}$ with respect to satisfying the instruction i on current state s_t.

The later term of Equation 1 represents the probability that the action sequence $a_{t:H}$ achieve rewards $r_{t:H}$ when executed from the state s_t, which is conditionally independent of the instruction i. We consider the success probability of the action sequence $\prod_{x=t}^{H} p(r_x \mid s_{t:x}, a_{t:x})$ as **feasibility** score S_{fb}. The sequence of actions $\{a_t, \cdots, a_H\}$ is feasible if the robot can approach each action successfully, i.e., the reward sequence $r_{t:H}$ are all equal to 1. If even one skill fails, then the entire action sequence fails.

Methodology

As illustrated in Figure. 2, we propose the Instruction-Augmented Long-Horizon Planning (IALP) system to infer actions for solving long-horizon tasks in a close-loop manner. Firstly, the system augments the user instruction into a PDDL problem through promptable and grounding mechanism-based predicates. These predicates utilize 2D and depth information obtained from embodied sensors, along with a primitive skill library L^{ψ}, to represent the current state. Based on augmented instruction, we utilize the LLM planner to generate K action candidates, from which the optimal action is selected based on the highest token probability. The system's extensibility and flexibility are demonstrated by the capability to incorporate additional promptable and grounding mechanism-based predicates as required by the task.

Promptable	on, in, holding, opened
Grounding Mechanism	at, find, graspable, placeable, detected, reachable

Table 1: Promptable and grounding mechanism-based predicates.

Instruction Augmentation

To provide efficient and sufficient task-related knowledge for the planner, we leverage off-the-shelf visual foundation models, specifically the SAM and CLIP, among others, accessed via designated APIs. This approach facilitates the construction of the PDDL problem by augmenting the textual human instruction i with embodied sensors. The LLM-based planner requires a domain description D that defines a set of predicates (e.g., `reachable`) and actions (e.g., `move`), as well as a problem description P for each task. The problem description defines the task by specifying the `init` state and `goal` state using a list of predicates defined in the domain file.

Predicates. We consider utilizing a predicate library $\mathcal{L}^{\chi} = \{\chi^1, \cdots, \chi^M\}$, as shown in Table 1, to describe the current observation and feasibility feedback. Each predicate is a binary-valued function over objects. For example, `find` represents a binary-valued function, with `book` serving as the object argument for predicate `(find book)`. This library consists of four promptable predicates that can be addressed through prompt engineering based on the reasoning ability of state-of-the-art LLMs, such as `holding` and `on`, and six predicates determined by grounding mechanisms, such as `find` and `reachable`. For promptable predicates, LLMs will generate "True" or "False" based on textual or multi-modal inputs prompted by designed prompts. For example, if the robot successfully takes the book from the table, LLMs might respond "False" to `(on book table)` and "True" to `(holding book)`. Grounding mechanism-based predicates are primarily employed for generating feasibility feedback, as detailed in Section Feasibility.

Actions. The robot is equipped with a primitive action library $\mathcal{L}^{\psi}$, which is inherently imperfect and frequently cause unforeseen situations. We list seven actions in Table 2 each associated with a preconditions set ψ^p and effects ψ^e of executing the action. An action is feasible to execute if

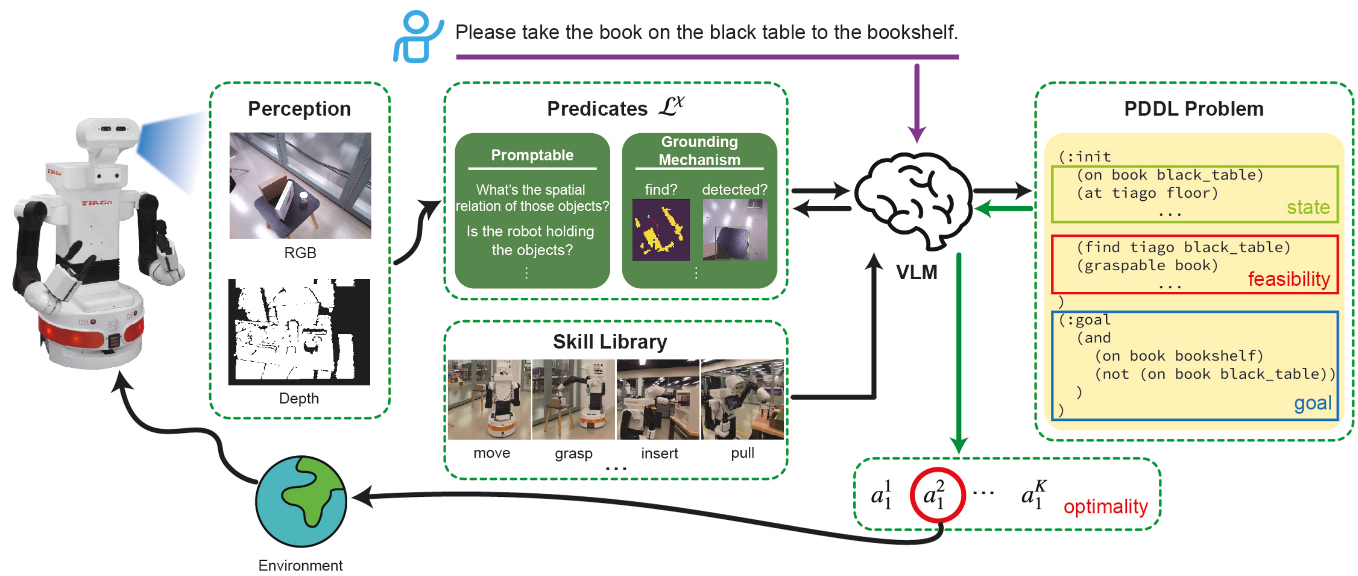

Figure 2: The proposed IALP system is designed to complete long-horizon mobile manipulation tasks in real world environment. Firstly, it constructs a PDDL problem for the task at current state by utilizing promptable and ground truth predicates derived from embodied perception and the skill library. Then, the robot executes the actions generated and selected by the LLM planner based on the constructed PDDL problem. The system operates in a closed-loop manner until the task is completed.

Actions	Preconditions ψ^p	Effects ψ^e
`move`	`(find obj)`	`(at obj)`
`scan`	`(at obj)` `(not (detected obj))`	`(detected obj)`

`grasp`	`(at obj)`
	`(detected obj)`
	`(reachable obj)`
	`(graspable obj)`
		`(holding obj)`

`place`	`(at pla)`
	`(holding obj)`
	`(peaceable pla)`
	`(reachable pla)`
		`(not (holding obj))`
		`(on obj pla)`

`lift`	`(holding obj)`	`(holding obj)`
`pull`	`(holding obj)`	`(opened obj)`

`insert`	`(at pla)`
	`(holding obj)`
	`(peaceable pla)`
	`(reachable pla)`
		`(not (holding obj))`
		`(on obj pla)`

Table 2: Preconditions and effects of actions in PDDL. `obj` and `pla` refer to the name of the object and peaceable space involved in the action.

and only if all the predicates in its precondition set are evaluated as "True". A reward of 1 is received if the robot successfully executes with the effects representing the expected state changes. For example, `move(obj)` denotes the moving action with a primitive function ϕ: `move` and its parameter α: `obj`. This action can be executed only if there's at least one path from the current position to the `obj`.

Current state. Unlike the definition in the original PDDL paper, which uses `:init` as the initial state of the task, we utilize it to represent the current state s_t at every timestep t. We initially employ LLMs to identify task-specific objects relevant to the current task. For instance, given the instruction "Take the blue jacket on the cloth hanger to the door", the objects "blue jacket", "cloth hanger" and "door" are extracted. Subsequently, utilizing these identified objects and the provided instruction, we apply the list of promptable predicates (Table 1) to derive knowledge about the current state. This includes determining the spatial relationships among the objects and the status (e.g., whether an object is being held or if a drawer is open) of the manipulated object.

Goal state. Given an instruction i, a set of objects o to be used in the current task is first generated by the LLM. Based on predicates library $\mathcal{L}^X$, we use the LLM to predict a symbolic goal g, which is a set of predicates grounded over objects o in the scene, to satisfy the instruction.

Feasibility

We employ an interactive planning approach, which involves replanning at each time step to allow the planner to adapt even if previous actions failed. Thus, here we consider the feasibility score of planning single time step, i.e., $S_{fb} = p(r_t|s_t, a_t)$. We aim to maximize the probability $p(1|s_t, a_t)$ of achieving a reward of 1 at state s_t by performing action a_t, as determined by the LLM planner. The entire action sequence $a_{1:H}$ fails, denoted by $S_{fb} = 0$, if at least one action fails, i.e., $r_t = 0, \exists t \in \{1 : H\}$. Two primary factors influence the probability $p(1|s_t, a_t)$. First, the feasibility of executing the action a_t at state s_t, such as whether the object to be manipulated can be grasped. Second, the likelihood of the action a_t succeeding when it is feasible. For instance, a robot cannot move toward a blue jacket if it cannot identify a feasible path. When there's a feasible path, navigation errors can still cause the action to fail.

To address the first factor, we aim to enhance the probability of action feasibility by providing more detailed feasibility feedback. We introduce six feasibility predicates, comprising two navigation predicates and four manipulation predicates, to maximize the feasibility score S_{fb} thereby increasing the likelihood that the generated actions can be successfully executed.

Navigation feasibility. Upon receiving an action to move toward an object, the robot must ensure the existence of at least one viable path from its current position to the target object. We utilize the predicate `find` to represent this determination and `at` to record the robot's current position.

To achieve this, we constructed a voxel map to provide a static representation of the current environment, employing the natural language mapping approach proposed by (Chen et al. 2023; Liu et al. 2024). This method maps visual-language embeddings of objects to their corresponding positions. We then use the SAM to generate masks for all potential objects (extracted from the current state) within the environment. From those masks, we extract corresponding crops c, including 2D image crops c_{RGB} and depth crops c_D. Object embeddings e_o are obtained via CLIP to encode image crops c_{RGB}, while the corresponding position (x_o, y_o, z_o) are derived from depth crops c_D. For a given object language description l_o, we first convert it to a semantic vector e'_o using the CLIP text encoder. The spatial position of an object language description l_o can be extracted by sampling from the voxel map, where the dot product between the image-encoded vector e_o and text-encoded vector e'_o is maximized. The A^* algorithm is employed to identify a path from the current position to the target position. For `at` predicate, we assume the robot will reach the target position after executing the action to move to the object.

Manipulation feasibility. We employ predicates `detected`, `graspable`, `placeable` and `reachable` to evaluate whether an object can be manipulated.

When the robot arrives at the object, we first ensure that the object is within the camera's field of view, denoted by predicate `detected`. To achieve this, we utilize the LangSAM (Medeiros 2023) model, which extracts the desired object's mask based on the object language description l_o and the image captured by the robot. For determining the predicates `graspable` and `placeable`, we employ the grasping and placing heuristics proposed by (Liu et al. 2024) which utilize foundational VLMs to verify and generate graspable poses and placeable areas. A pre-trained GraspNet (Fang et al. 2023) is used to generate several potential grasps for the robot, with LangSAM masks filtering the object-related grasps. The optimal placeable location for various surfaces is determined by first segmenting and aligning the point cloud, then calculating median coordinates and identifying the maximum height of the placeable object. For predicate `reachable`, we query these grasp poses in the pre-computed reachability map, where the reachability of any SE(3) end-effector pose of a robot is determined (Jauhri, Lueth, and Chalvatzaki 2024; Zacharias, Borst, and Hirzinger 2007; Vahrenkamp, Asfour, and Dillmann 2013).

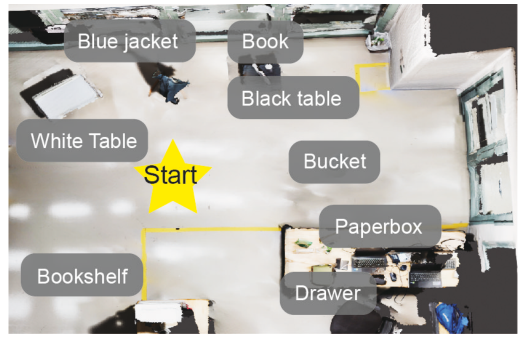

Figure 3: The room environment we used in our tasks. The position of the yellow star is the initial state of the robot.

We exclude any grasps that cannot be reached in the current state by computing a grasp reachability index for each candidate grasp. This map is created by executing forward kinematics for numerous joint configurations and recording the visitations and manipulability of the 6D voxels visited by the end-effector.

Optimality

We aim to maximize the optimality score S_{op} of the action sequence $a_{t:H}$ at time step t. Given our interactive planning method, which involves planning actions at every time step, the objective of the optimality score for planning at a single time step in state s_t is to maximize the probability $p(a_t|i, s_t)$ of next action a_t. This probability is independent of prior rewards $r_{1:t-1}$ and actions $a_{1:t-1}$. Given the PDDL problem P of a specific task, domain D, and user instruction i, we first query the LLM planner to generate K candidates actions $\{a_t^1, \cdots, a_t^K\}$ at state s_t. We then compute the optimality scores S_{op} by summing the token log probabilities of each action's language description. These scores represent the likelihood that a_t^k is the optimal skill to execute from a language modeling perspective to satisfy instruction i. The action with the highest token probability is then selected as the optimal action, i.e.,

$$a_t^* = \arg\max p_t(a_t^k|i, D, P), k \in \{1, \cdots, K\}. \quad (2)$$

where p_t is token probability of action.

Experiments

Experimental Setup

We conducted five long-horizon tasks, each consisting of several skills commonly used in real-life scenarios, as detailed in Figure 3 and Table 3, to evaluate the performance of the proposed system in real-world settings. Each task was accompanied by a natural language instruction i, and the robot initiated from the initial state at the beginning of each task. The experiments were carried out using the Tiago++ (Pages, Marchionni, and Ferro 2016) mobile robot, which is equipped with wheels and two 7-degree-of-freedom arms. The robot features an integrated RGB-D camera on its head, which is utilized for perception throughout the

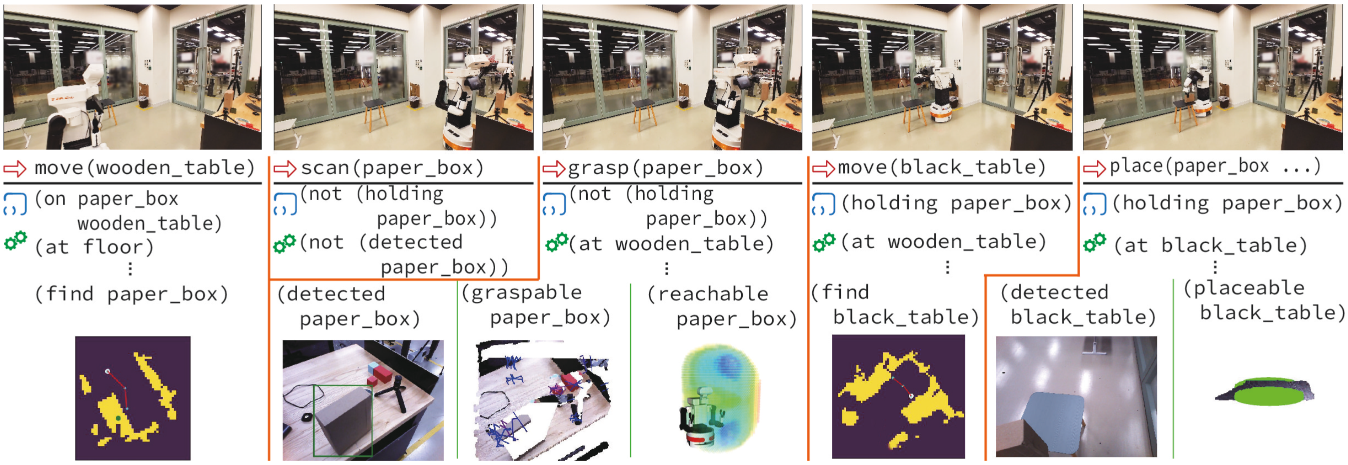

Figure 4: The states, feasibility feedback, and actions during the execution of long-horizon mobile manipulation tasks.

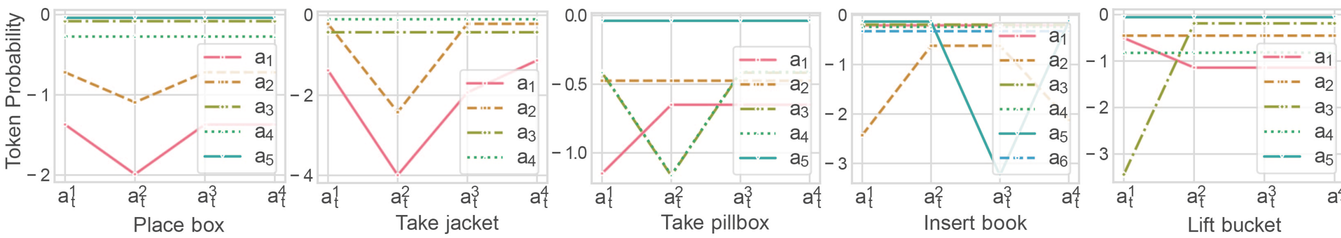

Figure 5: The token probability of action candidates generated by the LLM planner for five long-horizon tasks.

Task	Instruction
Place box	Pick the paper box on the wooden table and place it on the black table.
Take jacket	Take the blue jacket on the cloth hanger to the door.
Take pillbox	Take the pill box in the drawer to me.
Insert book	Take the book from the black table and put it on the bookshelf.
Lift bucket	Lift the blue bucket to the white table.

Table 3: The five long-horizon tasks and instructions employed in our experiments.

tasks. For motion planning of the robotic arms, we employed Pinocchio (Carpentier et al. 2019). High-level planning and promptable predicate checking were performed using the gpt-4o (Achiam et al. 2023) model. We used three computers: one for controlling the robot with the ROS Noetic system and others for generating navigation manipulation feedback. The robot employed the `adjust` action to modify its position, head tilt, and pane orientation several times to make the manipulated object feasible to grasp if certain manipulation feasibility predicates were not met. Seven prompts are used for promptable predicates identification, PDDL state and goal generation, and task planning. For other failed predicates, the robot could always use the `alert` action to report issues to a human operator.

Mobile Manipulation Tasks with the LLM Planner

Planning results. We investigated the capability of the LLM planner to perform zero-shot planning for long-horizon tasks based on the robot's perception and a pre-built voxel map. We recorded the actions and constructed PDDL problems for five long-horizon tasks, with the results for the "Take jacket" task illustrated in Figure 4. Given the instruction, "Pick the paper box on the wooden table and place it on the black table," and with the 2D and 3D images of the current state, we first construct the current state and goal state with predicates defined above. Then, we filter and integrate all possible predicate-object pairs, which are utilized to extract the knowledge of the environment, by inputting the designed prompt, available predicates and objects, and user instructions into the LLM planner. The promptable and grounding mechanism-based predicates are utilized to generate observations (e.g., the paper box is `on` the wooden table), and feasibility feedback (e.g., `find` a path to the paper box). For instance, when the robot is `at` the paper box (as determined by the visiting records) and the paper box is `detected` (as determined by LangSAM), `graspable` (as determined by filtering the grasping poses of the paper box generated by GraspNet), and `reachable` (determined by the reachability map), while the robot is not `holding` it, then the LLM planner will generate action primitive `grasp` with its parameter `paper_box` as the optimal action to execute. We

Tasks	Place box	Take jacket	Take pill box	Insert book	Lift bucket
Navigation	27	24	32	88	32
Manipulation	179	75	112	208	256
Planning	106	113	158	145	124

Table 4: The temporal duration (measured in seconds) used for the completion of tasks in a single successful execution.

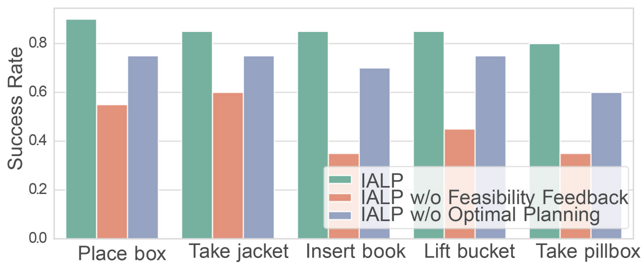

Figure 6: The success rate of IALP compared with that of IALP without feasibility feedback and without optimal selection, respectively.

list the actions and PDDL problems generated for the other four long-horizon tasks on the website.

Optimal actions. To evaluate the optimality of the actions generated by the LLM planner during the task, we collected the token probabilities of four action candidates $\{a_t^1, \cdots, a_t^4\}$ generated for each action a_t, where $t \in \{1 : H\}$, across five long-horizon tasks. The results are presented in Figure 5. The token probabilities exhibit variance due to the inherent uncertainty in the LLM's perception of the current state. Additionally, we observed that as the task progresses, the token probabilities of the four candidates converge, due to the reduction in task uncertainty as more information about the task and environment is provided to the LLM planner. Furthermore, we observed that as the task progresses, the token probabilities of candidate actions become increasingly consistent and higher, corresponding to a decrease in task uncertainty as more information about the task and the environment is provided to the LLM planner.

Temporal duration. We also recorded the navigation, manipulation, and planning times used to complete these long-horizon tasks in the real world, with the results presented in Table 4. These results demonstrate that our method can accomplish these tasks within a reasonable time.

Ablation Study on Feasibility and Optimality

To evaluate the impact of feasibility feedback and optimal selection on system performance, we conducted two ablation experiments, excluding either the feasibility feedback or optimal selection component separately to isolate their individual contributions to the overall system. All methods employed GPT-4o as the underlying planner. We generated 20 sets of observations for each task in different settings. The predicates in each set of observations were randomly assigned, i.e., set to True with a probability of 50%. We then

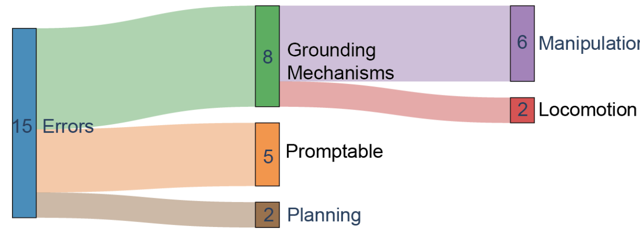

Figure 7: All failure cases of predicate checking in the real-world experiments across five long-horizon tasks.

recorded the success cases of the LLM planner generating executable actions, as shown in Figure 6. The results indicate that IALP achieves a success rate of over 80% in all long-term tasks. For the system without feasibility feedback (labeled as **IALP w/o Feasibility Feedback**), it encounters difficulty in generating feasible actions due to the removal of feasibility predicates such as `find` and `reachable` from the PDDL description. As a result, the success rate is substantially lower than that of other configurations. For the system without optimal selection, denoted as **IALP w/o Optimal Selection**, a relatively high success rate is still maintained because feasibility checks are applied to all action generation, which ensures a certain degree of effectiveness despite the absence of optimal selection.

Discussion

To investigate the types of failure cases in real-world experiments, we conducted 20 trials for each task within a real-world environment and recorded all occurring errors. All instances of predicate-checking failures were systematically aggregated and classified into three categories: planning, promptable, and grounding mechanisms failures. The results are presented in Figure 7. Planning failures occur when the planner fails to generate the correct action sequence. In contrast, promptable and grounding mechanism failures arise when predicates listed in Table 1 yield incorrect outputs, such as incorrect grasp poses for graspable objects. While 15 errors out of 100 may appear insignificant, they represent a considerable workload in real-world hardware experiments compared with numerical simulations due to factors such as hardware issues, noise, and physical limitations.

Conclusion

We propose IALP, a framework that leverages promptable and grounding mechanism-based predicates to construct an informative PDDL problem to represent task-relevant information of the current state. This enables the LLM planner to generate feasible and optimal actions in a closed-loop manner. Despite achieving the anticipated performance, our system has certain limitations. IALP is dependent on a predefined primitive action library, which restricts the range of task instructions and impedes the generalization of existing actions. Future work will focus on developing a more generalized action model capable of learning multiple skills and enhancing the dynamic reasoning ability of VLMs.

Acknowledgments

This work is supported in part by the Research Grants Council of Hong Kong under grant number C4042-23GF, and in part by the PolyU-EIT Collaborative PhD Training Programme under application number 220724983.

References

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*.

Ahn, M.; Dwibedi, D.; Finn, C.; Arenas, M. G.; Gopalakrishnan, K.; Hausman, K.; Ichter, B.; Irpan, A.; Joshi, N.; Julian, R.; et al. 2024. Autort: Embodied foundation models for large scale orchestration of robotic agents. *arXiv preprint arXiv:2401.12963*.

Ao, J.; Wu, F.; Wu, Y.; Swikir, A.; and Haddadin, S. 2024. LLM as BT-Planner: Leveraging LLMs for Behavior Tree Generation in Robot Task Planning. *arXiv preprint arXiv:2409.10444*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Carpentier, J.; Saurel, G.; Buondonno, G.; Mirabel, J.; Lamiraux, F.; Stasse, O.; and Mansard, N. 2019. The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, 614–619. IEEE.

Chen, B.; Xia, F.; Ichter, B.; Rao, K.; Gopalakrishnan, K.; Ryoo, M. S.; Stone, A.; and Kappler, D. 2023. Open-vocabulary queryable scene representations for real world planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 11509–11522. IEEE.

Fang, H.-S.; Wang, C.; Fang, H.; Gou, M.; Liu, J.; Yan, H.; Liu, W.; Xie, Y.; and Lu, C. 2023. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*.

Fox, M.; and Long, D. 2003. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research*, 20: 61–124.

Garrett, C. R.; Lozano-Pérez, T.; and Kaelbling, L. P. 2020. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the international conference on automated planning and scheduling*, volume 30, 440–448.

Geng, H.; Wei, S.; Deng, C.; Shen, B.; Wang, H.; and Guibas, L. 2023. Sage: Bridging semantic and actionable parts for generalizable articulated-object manipulation under language instructions. *arXiv preprint arXiv:2312.01307*.

Guan, L.; Valmeekam, K.; Sreedharan, S.; and Kambhampati, S. 2023. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36: 79081–79094.

Huang, S.; Jiang, Z.; Dong, H.; Qiao, Y.; Gao, P.; and Li, H. 2023. Instruct2act: Mapping multi-modality instructions to robotic actions with large language model. *arXiv preprint arXiv:2305.11176*.

Jauhri, S.; Lueth, S.; and Chalvatzaki, G. 2024. Active-perceptive motion generation for mobile manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 1413–1419. IEEE.

Kambhampati, S.; Valmeekam, K.; Marquez, M.; and Guan, L. 2023. On the role of large language models in planning. In *Tutorial presented at the International Conference on Automated Planning and Scheduling (ICAPS), Prague*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Lin, K.; Agia, C.; Migimatsu, T.; Pavone, M.; and Bohg, J. 2023. Text2Motion: From Natural Language Instructions to Feasible Plans. *Autonomous Robots*, 47(8): 1345–1365. ArXiv:2303.12153 [cs].

Liu, B.; Jiang, Y.; Zhang, X.; Liu, Q.; Zhang, S.; Biswas, J.; and Stone, P. 2023. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*.

Liu, P.; Orru, Y.; Paxton, C.; Shafiullah, N. M. M.; and Pinto, L. 2024. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*.

Medeiros, L. 2023. Lang segment anything. https://github.com/luca-medeiros/lang-segment-anything.

Minderer, M.; Gritsenko, A.; Stone, A.; Neumann, M.; Weissenborn, D.; Dosovitskiy, A.; Mahendran, A.; Arnab, A.; Dehghani, M.; Shen, Z.; et al. 2022. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, 728–755. Springer.

Pages, J.; Marchionni, L.; and Ferro, F. 2016. Tiago: the modular robot that adapts to different research needs. In *International workshop on robot modularity, IROS*, volume 290.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Silver, T.; Hariprasad, V.; Shuttleworth, R. S.; Kumar, N.; Lozano-Pérez, T.; and Kaelbling, L. P. 2022. PDDL planning with pretrained large language models. In *NeurIPS 2022 foundation models for decision making workshop*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Vahrenkamp, N.; Asfour, T.; and Dillmann, R. 2013. Robot placement based on reachability inversion. In *2013 IEEE International Conference on Robotics and Automation*, 1970–1975. IEEE.

Valmeekam, K.; Marquez, M.; Sreedharan, S.; and Kambhampati, S. 2023. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36: 75993–76005.

Wake, N.; Kanehira, A.; Sasabuchi, K.; Takamatsu, J.; and Ikeuchi, K. 2023. ChatGPT Empowered Long-Step Robot Control in Various Environments: A Case Application. *IEEE Access*, 11: 95060–95078.

Wang, J.; Dai, R.; Wang, W.; Rossini, L.; Ruscelli, F.; and Tsagarakis, N. 2024. HYPERmotion: Learning Hybrid Behavior Planning for Autonomous Loco-manipulation. *arXiv preprint arXiv:2406.14655*.

Yenamandra, S.; Ramachandran, A.; Yadav, K.; Wang, A.; Khanna, M.; Gervet, T.; Yang, T.-Y.; Jain, V.; Clegg, A. W.; Turner, J.; et al. 2023. Homerobot: Open-vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*.

Zacharias, F.; Borst, C.; and Hirzinger, G. 2007. Capturing robot workspace structure: representing robot capabilities. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3229–3236. Ieee.

Zhang, X.; Altaweel, Z.; Hayamizu, Y.; Ding, Y.; Amiri, S.; Yang, H.; Kaminski, A.; Esselink, C.; and Zhang, S. 2024. DKPROMPT: Domain Knowledge Prompting Vision-Language Models for Open-World Planning. *arXiv preprint arXiv:2406.17659*.