ICRA2026 Late Breaking Results Poster presented at
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

GLaMP: A Grounded Language Model-based Multi-agent System for Long-Horizon Robotic Task Planning in Industrial Settings

Hongpeng Chen, David Navarro-Alarcon, Pai Zheng

***Abstract*—This paper proposes GLaMP, a grounded language model–based multi-agent system for long-horizon robotic task planning in industrial settings. GLaMP uses a vision-language model (VLM) agent to infer a hierarchical task graph from manuals, grounds multimodal observations into planning domain definition language (PDDL) predicates to maintain symbolic consistency, and employs an large language model (LLM) behavior-tree planner for interpretable plan generation and execution. Typed-symbolic feedback enables failure-aware re-grounding and fallback replanning. Experiments on five industrial tasks show that GLaMP consistently outperforms representative baselines in success rate, planning latency, and execution time.**

I. Introduction

This paper proposes GLaMP, a grounded language model–based closed-loop multi-agent framework where VLM/LLM agents collaborate through a bidirectional perception–grounding–planning loop. The key insight is that bidirectional feedback is essential for preventing predicate drift in long-horizon industrial tasks. The contributions are: (1) a closed-loop multi-agent framework with bidirectional information flow for long-horizon industrial planning; (2) a VLM-driven hierarchical task-graph induction method with nested compositional and part-equivalence relations; and (3) a predicate-consistent planning-to-execution paradigm using reactive behavior trees with multimodal re-grounding and adaptive parameter regeneration to prevent predicate drift.

II. Methodology

As illustrated in Fig. 1, GLaMP is a closed-loop multi-agent framework composed of three collaborative agents with bidirectional information flow. The **task understanding agent** leverages a VLM to induce a hierarchical task graph from manuals and diagrams, capturing nested compositional dependencies and part-equivalence relations for reusable task decomposition. The **perception agent** grounds real-time multimodal observations into PDDL predicates to maintain symbolic–physical alignment, and performs runtime predicate verification and failure classification, enabling upstream re-grounding rather than a strictly sequential pipeline. The **planning agent** compiles instructions into reactive behavior trees validated against PDDL preconditions and effects, with re-grounding and parameter regeneration enabled before fallback replanning.

H. Chen, D. Navarro-Alarcon, and P. Zheng are with The Hong Kong Polytechnic University, Hong Kong.

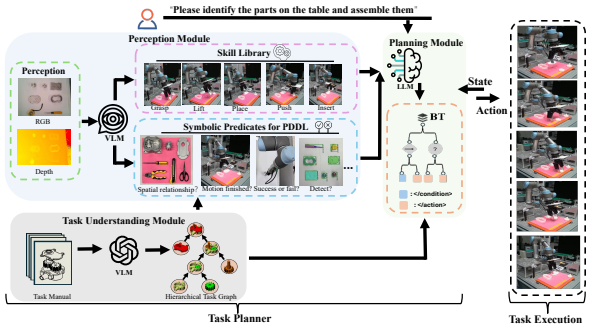

Fig. 1. Architecture of the GLaMP framework

A. Manual-Grounded Hierarchical Task-Graph Induction

This module constructs a symbolic directed acyclic graph $G = (O, H, R)$ encoding compositional dependencies and part-equivalence relations from manuals and diagrams. The induction proceeds in two VLM-driven stages via chain-of-thought prompting. In Stage I, the VLM processes labeled scene images alongside manual illustrations to identify each component's name, functional role, and equivalence class for identical parts. Set-of-Marks and Grounding DINO are used for automatic component labeling to enhance recognition accuracy. In Stage II, the VLM sequentially analyzes manual pages to infer step-wise assembly hierarchy, where each non-leaf node represents a subtask composed of disjoint child components. Equivalence edges allow interchangeable parts across subtrees, enabling subplan reuse. The resulting task graph is compiled into a PDDL domain defining the skill set, preconditions, effects, and goals.

B. Closed-Loop Multimodal Predicate Grounding

The perception agent integrates a VLM for semantic parsing, Grounded SAM [1] for segmentation and 6D pose estimation, and AnyGrasp [2] for grasp synthesis. Continuous perceptual observations are mapped to discrete PDDL predicates such as $\texttt{at}(o_i, loc_j)$ and $\texttt{graspable}(o_i)$ that serve as action preconditions.

Critically, the perception agent is invoked after every action to verify expected symbolic effects through VLM-based visual question answering. When inconsistencies are detected, failures are classified into typed cases—*insertion/precision* (misalignment or excessive force) and *grasp/hold* (unstable grasp or slippage)—to guide recovery. This closed-loop verification forms the bidirectional feedback that distinguishes GLaMP from one-way pipelines.

C. LLM-Enabled Predicate-Consistent Planning and Execution

The planning agent uses an LLM to compile high-level instructions into reactive behavior trees. Each leaf node corresponds to an instantiated PDDL skill with typed preconditions and effects. Sequence, fallback, and action nodes encode execution logic with runtime adaptability. All skills include velocity and clearance as safety-critical parameters.

A tiered recovery strategy handles failures: adaptive parameter regeneration retries up to $F_{\max}$ times before triggering fallback. For insertion errors, guarded motion reduces velocity and force thresholds; for grasp failures, a new pose is sampled within a shrinking radius. Multi-modal success detection combines VLM-based visual reasoning with proprioceptive feedback. Only when all regeneration attempts are exhausted does the behavior tree invoke the fallback branch.

III. Experiments

A. Experimental Setup

The system consists of a UR5 robotic manipulator equipped with a Robotiq 2F-85 parallel gripper and an Intel RealSense L515 LiDAR RGB-D camera mounted on the end-effector. The agentic LLM/VLM is accessed via the OpenAI GPT-4o API. Five benchmark experiments were conducted: *Pick and Place*, *Pick and Lift*, *Component Storage*, *Interlocking Block Assembly*, and *Mechanical Parts Assembly*, designed as long-horizon tasks involving multiple interdependent subtasks. Each task was executed 50 times, and the same prompt templates were used across all tasks.

B. Ablation Studies

Ablation studies on three long-horizon tasks confirm that all components are essential (see Fig. 2). Removing the task understanding module causes the largest performance drop because the planner loses the hierarchical task graph and fails to decompose tasks correctly. Removing behavior tree reactivity leads to clear degradation because open-loop execution cannot recover from grasp/pose/insertion errors online. Removing upstream feedback causes stale predicates and grounding mistakes to accumulate, leading to unrecoverable deadlocks.

C. Task Execution and Comparison Results

GLaMP achieves success rates of 92% on simple pick-and-place tasks, 88–90% on medium-complexity tasks, and 82% on the most challenging mechanical parts assembly. Error analysis reveals that perception errors dominate in simple tasks, task understanding errors emerge at intermediate complexity, and planning errors dominate at higher complexity. Planning latency ranges from 12 s for simple tasks to 26 s for complex ones, constituting a small fraction of the overall task cycle in targeted flexible manufacturing scenarios.

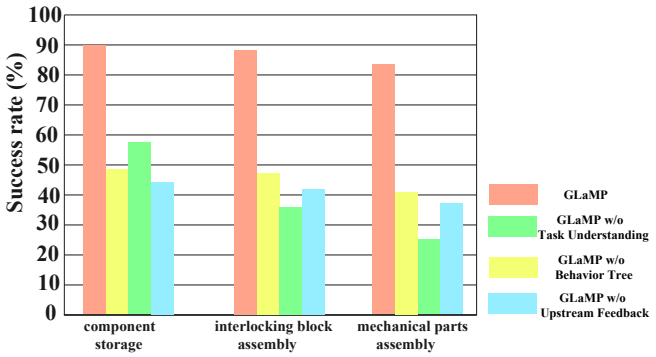

Fig. 2. Ablation study of GLaMP: success rates under removing task understanding, behavior tree, or upstream feedback.

Table I compares GLaMP against some representative baselines. GLaMP consistently achieves the highest success rates, outperforming IALP [3] by 5–12% and ALBP-BT [4] by 10–20%. The improvement is attributed to hierarchical behavior tree–based execution with reactive re-planning and robust fallback strategies. GLaMP also demonstrates competitive or lower planning latency due to its modular PDDL-based prompt generation, and shorter execution times due to fewer planning retries enabled by visual-symbolic alignment.

TABLE I
Performance comparison across three long-horizon tasks. SR: Success Rate (%), PT: Planning Time (s), ET: Execution Time (s).

Method	Comp. Storage			Interlock. Asm.			Mech. Parts		
	SR	PT	ET	SR	PT	ET	SR	PT	ET
GLaMP	**90.1**	**21.0**	**175**	**88.1**	**24.0**	**185**	**82.4**	**25.0**	**188**
IALP [3]	85.5	24.8	181	75.0	27.5	189	74.5	28.5	192
ALBP-BT [4]	79.8	22.4	177	70.3	23.6	182	63.9	25.0	186

IV. Conclusion

This work presents GLaMP, a grounded language model–based multi-agent framework that bridges perceptual–symbolic discrepancies for long-horizon industrial manipulation. Experiments on five tasks show that GLaMP consistently outperforms representative baselines in success rate, planning latency, and execution time.

References

[1] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, *et al.*, "Grounded sam: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024.

[2] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3929–3945, 2023.

[3] F. Wang, S. Lyu, P. Zhou, A. Duan, G. Guo, and D. Navarro-Alarcon, "Instruction-augmented long-horizon planning: Embedding grounding mechanisms in embodied mobile manipulation," 2025.

[4] J. Wang, A. Laurenzi, and N. Tsagarakis, "Autonomous behavior planning for humanoid loco-manipulation through grounded language model," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 10 856–10 863.