HAMMR: A Human-Aligned Multi-Agent Framework for Language-Guided Robotic Manipulation

Demonstration Track

Roopesh Kumar Shanmugasundaram
University of Jyväskylä
Jyväskylä, Finland
roopesh.k.shanmugasundaram@jyu.fi

Niko Mäkitalo
University of Jyväskylä
Jyväskylä, Finland
niko.k.makitalo@jyu.fi

ABSTRACT

We present HAMMR, a human-aligned multi-agent framework for language-guided robotic manipulation, enabling explainable planning, step-by-step action justification, risk assessment, and mandatory human approval, achieving 89% success on RLBench tasks.

KEYWORDS

Multi-Agent Systems; Explainable AI; Autonomous Manipulation

ACM Reference Format:
Roopesh Kumar Shanmugasundaram and Niko Mäkitalo. 2026. HAMMR: A Human-Aligned Multi-Agent Framework for Language-Guided Robotic Manipulation: Demonstration Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/BHTF7700

1 INTRODUCTION

Large language models are increasingly transforming robotics by enabling natural language understanding, high-level task planning, decision making, and intuitive human–robot interaction [12]. Early systems such as SayCan [1] grounds language in action by using LLMs to propose action sequences and value functions to evaluate their feasibility, while Inner Monologue [5] extends this paradigm with closed-loop planning through continual perception and self-reflection. Code-as-Policies [9] and RobotGPT [7] translate instructions into executable programs or action code, emphasizing programmatic control. RT-2 [2] enables generalized robotic control through end-to-end vision–language–action models, whereas VoxPoser [4] emphasizes interpretability by building structured 3D scene representations. Progress in LLM reasoning frameworks, particularly ReAct [13], enables agents to interleave reasoning and action through iterative thought–action–observation cycles. ROSA [10] integrates this paradigm into robotic systems as embodied agents. Extending beyond single-agent systems, multi-agent architectures enable collaborative specialization, improving robustness on complex tasks. SMART-LLM [8] enables collaborative multi-agent reasoning by decomposing tasks and orchestrating robots based on their skills, while MALMM [11] introduces a Planner–Coder–Supervisor framework for zero-shot manipulation.

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/BHTF7700

While these approaches leverage LLMs for high-level planning [1, 5] and low-level control [9, 11], they prioritize autonomous execution over human understanding. Decision-making processes remain largely opaque, preventing users from inspecting plans, assessing safety, or intervening before actions occur. As highlighted by Raptis et al. [12], transparency and explainability remain under-explored in LLM-driven robotics, limiting deployment in safety-critical environments. Article 13 of the EU AI Act formalizes this requirement by mandating that high-risk AI systems provide sufficient transparency for deployers to interpret system outputs and exercise human oversight [3].

To address these gaps, we introduce **HAMMR (Human-Aligned Multi-Agent Manipulation with Reasoning)**[1]. It is designed to place humans explicitly in the decision loop by requiring interpretable planning and mandatory pre-execution approval before any physical action occurs. The framework extends existing multi-agent paradigms, by incorporating step-by-step plan justifications, task-level risk classification, and parameter-level user control. Autonomous execution is retained after approval, enabling efficient task completion without sacrificing transparency, interpretability, or alignment with human intent.

2 MOTIVATION

Figure 1 illustrates HAMMR's intended deployment within a smart factory setting, where heterogeneous robotic systems and sensors operating alongside human workers under a central supervisor. The supervisor issues natural-language commands ranging from global directives to coordinated multi-agent tasks. HAMMR enables selective human oversight: routine low-risk actions execute autonomously, while novel or safety-critical tasks generate explainable plans for supervisor review and approval, ensuring transparent and safe collaboration.

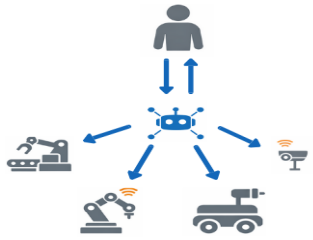

Figure 1: HAMMR in Smart Factory

[1]Demo video: https://youtu.be/DE1d4G4PUCk; Code repository: https://github.com/RoopsHub/rlbench-multi-agent

3 FRAMEWORK OVERVIEW

HAMMR enforces a strict two-phase workflow that distinctly separates human oversight from autonomous execution. The framework consists of four agents: a central **Planner Agent**, which orchestrates the overall workflow, and three specialized agents, **Sensing, Perception, and Motion**, that are executed sequentially and leverage high-level MCP tools to accomplish users request (Figure 2). MCP enhances modularity by supporting multiple simulators (e.g., Gazebo, CoppeliaSim) and enabling direct transfer to real robots, while providing an abstraction layer that decouples LLM reasoning from the underlying robot implementations. We implement HAMMR using RLBench [6], a benchmark suite for robot manipulation built on CoppeliaSim, featuring multi-modal observations including RGB images, depth maps, and point clouds. This work focuses on single-arm manipulation tasks, laying the groundwork for future deployments with heterogeneous robotic systems.

3.0.1 Phase 1: Explainable Planning. The Planner Agent receives natural language commands (e.g., "pick up the red cube") and identifies the task category by analyzing linguistic patterns. Each category maps to a predefined motion sequence template(e.g., this task follows: open gripper→approach cube→grasp→maintain closed gripper→move to target). The agent instantiates this template with step-by-step justifications (e.g., "move to 15cm above object to ensure collision-free descent") and performs automatic risk assessment: tasks without grasping are LOW risk, single-object manipulation is MEDIUM risk, multi-object tasks requiring state tracking are HIGH risk. Plans include adjustable parameters like approach height and grasp offset. Users modify these conversationally (e.g, "increase grasp offset to 0.02m), which triggers validation and updates to the plan. Critically, the orchestrator operates without tool access during planning, enforcing plan-execution separation. Only upon explicit user approval does autonomous execution begin. Once approved, execution proceeds automatically through three specialized agents in strict sequential order.

3.0.2 Phase 2: Autonomous Execution. Sensing Agent initiates the pipeline by invoking the load_task(task_name) MCP tool to initialize the RLBench task, followed by the get_camera_observation() MCP tool to capture multi-modal data: RGB image, depth map, point cloud, camera intrinsics, and camera-to-base pose transformation. The agent retrieves automatically extracted detection prompts ("red cube . red sphere") and ground-truth positions for validation. Outputs are structured as JSON and passed to the Perception Agent.

Perception Agent performs vision-language grounding via the detect_object_3d() MCP tool, which encapsulates: (1) GroundingDINO inference yielding 2D bounding boxes, (2) LAB color space verification to correct/verify object colors, (3) extraction of corresponding 3D points from depth data, and (4) coordinate transformation to the robot base frame. For single-object task, it returns position_3d: [x, y, z] with confidence. For multi-object tasks with compound prompts, it returns an objects[] array with positions and confidences.

Finally, Motion Agent executes approved motions using the move_to_position() and control_gripper() MCP tools,which abstracts inverse kinematics and trajectory generation. The agent maintains gripper state awareness and halts immediately upon failure.

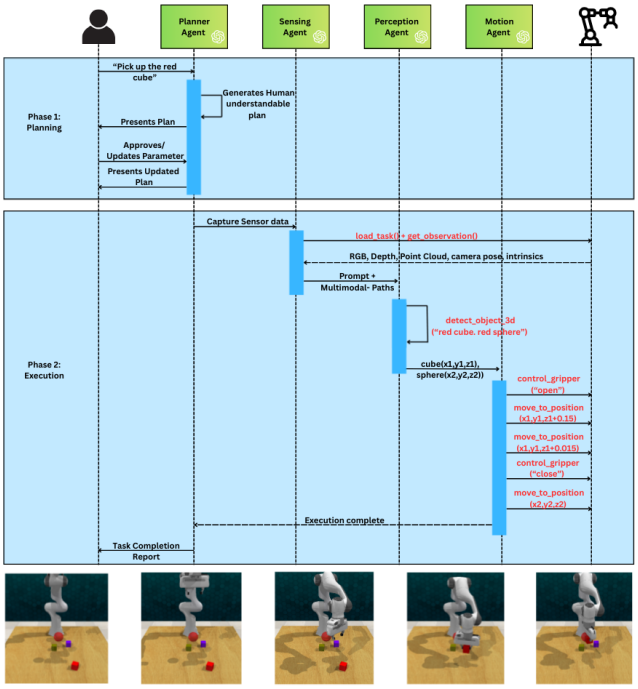

Figure 2: HAMMR multi-agent workflow demonstrating a red cube grasp-and-lift task

Table 1: HAMMR task performance across 15 trials per task (75 total episodes).

Task	*Complexity*	*Success*	*Failure Cause*
ReachTarget	Simple	15/15	-
PushButton	Simple	15/15	-
PickAndLift	Medium	12/15	Grasp misalignment
PutRubbishInBin	Medium	12/15	Perception noise
StackBlocks	Long-horizon	13/15	Placement instability
Overall		**67/75**	**89%**

4 EVALUATION

HAMMR is evaluated on five RLBench manipulation tasks spanning simple, medium, and long-horizon complexity categories across 75 trials (15 per task), using GPT-5-mini as the reasoning model. Table 1 reports task-wise results, achieving 89% overall success, with failures attributed solely to execution-level perception noise or motion inaccuracies rather than planning errors. Direct comparison with the closely related MALMM [11] was not possible due to the lack of publicly available replication resources; architecturally, HAMMR extends MALMM with step-wise justifications, risk classification, and operator-adjustable parameters. Future work will focus on benchmarking against related multi-agent frameworks and extending HAMMR to heterogeneous multi-robot settings, first in simulation and subsequently on real-world systems to evaluate scalability and deployment robustness under human oversight.

REFERENCES

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. arXiv:2204.01691 [cs.RO] http://arxiv.org/abs/2204.01691

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. arXiv:2307.15818 [cs.RO] http://arxiv.org/abs/2307.15818

[3] European Parliament and Council of the European Union. 2024. Article 13: Transparency and Provision of Information to Deployers. Regulation (EU) 2024/1689 of the European Parliament and of the Council on Artificial Intelligence. https://artificialintelligenceact.eu/article/13/

[4] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. 2023. VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models. arXiv:2307.05973 [cs.RO] http://arxiv.org/abs/2307.05973

[5] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. 2022. Inner Monologue: Embodied Reasoning through Planning with Language Models. arXiv:2207.05608 [cs.RO] http://arxiv.org/abs/2207.05608

[6] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. 2020. RLBench: The Robot Learning Benchmark & Learning Environment. *IEEE Robotics and Automation Letters* 5, 2 (2020), 3019–3026. https://doi.org/10.1109/LRA.2020.2974707

[7] Yixiang Jin, Dingzhe Li, Yong A, Jun Shi, Peng Hao, Fuchun Sun, Jianwei Zhang, and Bin Fang. 2023. RobotGPT: Robot Manipulation Learning from ChatGPT. arXiv:2312.01421 [cs.RO] http://arxiv.org/abs/2312.01421

[8] Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. 2024. SMART-LLM: Smart Multi-Agent Robot Task Planning using Large Language Models. arXiv:2309.10062 [cs.RO] http://arxiv.org/abs/2309.10062

[9] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as Policies: Language Model Programs for Embodied Control. arXiv:2209.07753 [cs.RO] http://arxiv.org/abs/2209.07753

[10] Rob Royce, Marcel Kaufmann, Jonathan Becktor, Sangwoo Moon, Kalind Carpenter, Kai Pak, Amanda Towler, Rohan Thakker, and Shehryar Khattak. 2025. Enabling Novel Mission Operations and Interactions with ROSA: The Robot Operating System Agent. arXiv:2410.06472 [cs.RO] http://arxiv.org/abs/2410.06472

[11] Harsh Singh, Rocktim Jyoti Das, Mingfei Han, Preslav Nakov, and Ivan Laptev. 2025. MALMM: Multi-Agent Large Language Models for Zero-Shot Robotic Manipulation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 20386–20393. https://doi.org/10.1109/IROS60139.2025.11247340

[12] Jiaqi Wang, Enze Shi, Huawen Hu, Chong Ma, Yiheng Liu, Xuhui Wang, Yincheng Yao, Xuan Liu, Bao Ge, and Shu Zhang. 2025. Large language models for robotics: Opportunities, challenges, and perspectives. *Journal of Automation and Intelligence* 4, 1 (2025), 52–64. https://doi.org/10.1016/j.jai.2024.12.003

[13] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629 [cs.CL] http://arxiv.org/abs/2210.03629