Task Planning for Mobile Manipulation in Retail Stores using Foundation Models with Iterative Re-planning

Vismay Vakharia*, Sanjana Garai, Rolif Lima, Nijil George, Vighnesh Vatsal, Kaushik Das

***Abstract*— Automation in industries such as retail, warehousing and logistics presents opportunities for greater throughput, cost reduction and mitigation of disruptions from labour shortages. Previously, such efforts have focused on back-room operations involving packing and sorting in relatively structured environments. With advances in robotic mobile manipulation hardware and foundation models, automation can now be applied to more variable and human-centric environments such as retail store shelves. In this work, we present a task-planning approach using Large Language Models (LLMs) and Vision-Language Models (VLMs) to address the restocking problem in retail scenarios such as supermarkets. We demonstrate this system on a custom omnidirectional mobile manipulation platform, with user-driven prompts and a feedback-based iterative re-planning approach for error correction. The end-to-end system is validated in a PyBullet simulation environment for pick-and-place tasks.**

I. INTRODUCTION

The modern retail industry is rapidly adopting robotics-enabled solutions, driven by the competitive nature of the sector and labour shortages, particularly in developed countries. Automating a retail store using mobile robots presents significant challenges, as evidenced by international robotics competitions such as the "Amazon Picking Challenge" [1] and "Future Convenience Store Challenge" [2]. The robots tasked with order picking, restocking, and organizing must decompose high-level goals into sequenced sub-tasks and generate efficient motion plans – all while navigating real-world uncertainty [3].

Traditionally, this has been achieved using Task and Motion Planning (TAMP) [4], [5] frameworks, that rely on symbolic reasoning that is manually integrated with continuous motion control. However, these methods are often domain-specific and rigid in their operation.

Recent work shows that Large Language Models (LLMs) can transform TAMP by replacing rigid rule-based systems with flexible, general-purpose reasoning [6], [7]. Pretrained on vast internet-scale text corpora using masked language modelling and autoregressive prediction objectives, LLMs can infer structured task sequences without the need for fine-tuning [8], adapt to environmental context, and even recover from failures without domain-specific engineering or retraining [9].

In this study, we explore this new paradigm using a custom-built, omnidirectional, dual-arm mobile manipulator [10], [11]. The system is tasked with 'order picking' in a simulated retail environment, where it must retrieve items from a given list in their respective quantities efficiently based on store layout. Our framework combines an LLM for high-level logical reasoning with a VLM for spatial understanding, using execution feedback to continually refine its plans. The proposed framework is tested in a PyBullet simulation environment.

The authors are with TCS Research, Tata Consultancy Services Ltd., Bengaluru - 560066, Karnataka, India. *Corresponding author, e-mail: vismay.vakharia@tcs.com

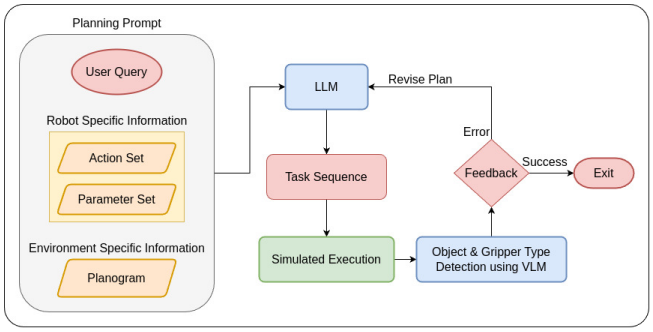

Fig. 1: Framework Architecture

II. METHOD

The proposed framework is initialized by the user inputting the query in natural language (English) to the LLM. Along with the query, a planogram containing the description of the environment and robot-specific information consisting of the feasible symbolic actions and their respective parameter sets (table I) are provided to the LLM.

The first stage of real-world task execution involves creating a plan that outlines a sequence of actions. The LLM is prompted to generate a task sequence from the robot's feasible actions to ensure that all the tasks are within the scope of the robot's capabilities while maintaining the context of the environment through the planogram. Each action takes specific parameters and performs the motion planning necessary to complete the task. Table I explains the robot's feasible actions and their respective parameters.

The omnidirectional mobile base is equipped with an autonomous navigation system including localization and path planning algorithms. The `navigate(table_location)` is a wrapper that interfaces the parametrized symbolic functions with a high-level motion planner that takes care of the navigation of the mobile base. A call of `navigate` makes the robot navigate to the desired target `table_location`.

In the `scan` function, the robot uses its head-mounted camera to capture an image and prompts VLM to find a `object_type` in the robot's workspace. The VLM's spatial reasoning comes in handy to identify which object would be ideal to grasp if multiple options are found and to determine the appropriate gripper type (rigid gripper or soft gripper) as well as the end-effector grasp approach depending on how tightly packed the objects are.

Action / Parameters	Description
`navigate(table_location)`	moves the robot base to the target `table_location`
`scan(object_type)`	captures an image and uses VLM to scan the environment; determines the object of `object_type` to grasp, the gripper type and grasp orientation
`pick(object)`	plans a trajectory, grasp the `object` and places it on the tray
`place(object)`	grasps the `object` from the tray and places on the drop table

TABLE I: Robot Feasible Actions and Parameters

The `pick` and `place` functions interface with motion planners for the two UR5e arms. Once the object, gripper, and approach are selected, the planner solves inverse kinematics and computes a trajectory. In `pick`, the object is retrieved from a shelf and placed on the robot's tray; in `place`, it's picked from the tray and placed in an appropriate slot identified using the VLM.

While the language models are excellent at generating task plans, they're not immune to mistakes. Our framework includes real-time checks to catch errors during execution and gives immediate feedback to the LLM/VLM for on-the-fly adjustments to the plan. This iterative process ensures smooth operation without any interruptions or manual interventions.

Since LLMs are inherently stochastic, they might generate invalid actions or incorrect parameters. While obvious errors—like syntax mistakes or out-of-bound values—are easy to catch due to the robot's limited and fixed capabilities, logical inconsistencies are trickier. For example, if the plan tells the robot to `navigate` to a table that doesn't match the target object, the syntax and parameters may seem correct, but the logic is flawed. Here we utilize the VLM to cross-check the objects in view against the plan and flag the error, triggering feedback to revise the plan. Similarly, we also use this feedback loop to identify and correct missing `navigation` steps before `pick` or `place` actions. Throughout, a buffer maintaining the current execution state is maintained and included in the prompt during plan revision provided to the LLM to refine the plan on the fly, without starting from scratch.

III. SIMULATION & EXPERIMENTS

A. Simulation Setup

Retail store environments commonly have flat, smooth floors and narrow aisles so we use a robotic system consisting of two 6-DOF UR5e manipulators mounted on an in-house-built omnidirectional mobile base, allowing motion in all directions and in-place rotation, making it suitable for tight retail spaces. Arms are equipped with two different types of grippers, a standard 2-finger gripper for grasping rigid objects and a reconfigurable 3-finger soft gripper to manipulate deformable objects [12].

For planning, we use Mixtral AI's `Mixtral 8x22b` [13] LLM model (141B parameters, 64k context window) and `Pixtral 12b` [14] VLM (400M vision encoder, 12B parameters multi-modal decoder and 128k context window). Both of these models are chosen for their scale, performance, and permissive open-source license.

We consider an order-picking task in a simulated retail setting using PyBullet [15]. A mobile manipulator navigates the environment to pick and deliver items in specified quantities. The setup includes 8 object types placed on 4 tables arranged in a 2x2 grid with space for robot movement. The objects have three different shapes - cuboid, cylindrical and spherical. Cuboid objects represent solid, non-deformable items that can be grasped using a standard 2-finger rigid gripper. In contrast, cylindrical and spherical objects represent irregularly shaped or deformable objects that require a 3-finger soft gripper. The user query consists of any combination of these items with any count (up to 15).

B. Evaluation

To evaluate our approach, we designed a series of experiments that progressively test reliability, reasoning, and ability to recover from failure.

We begin with simple, feasible order-picking tasks with shuffled item types and quantities to confirm that the system generates correct and consistent plans. Next, we test the LLM's robustness by providing infeasible or irrelevant queries such as "play music" or "explore space" which fall well outside the robot's action space. This helps assess whether the model can correctly reject or ignore nonsensical instructions. We also craft intentionally misleading prompts to test fine-grained understanding. For instance, if the environment contains red cubes and white spheres, we include queries for items that don't exist such as red spheres or white cubes. These tests examine whether the LLM-VLM system can accurately ground object descriptions in the observed environment. To evaluate re-planning capability, we inject errors into valid plans. For example, incorrect actions, object mismatches, wrong action sequences, or invalid parameters. The system begins execution with these flawed plans and is expected to detect the resulting failure and self-correct through feedback-driven refinement iterations.

The results of these experiments demonstrate that our method enabled the robot to successfully understand the user query and perform generated tasks despite the uncertainties and challenges in the environment. The successful execution of these experiments validates the practicality and robustness of our approach, showcasing its potential for various real-world applications such as order picking, restocking, and organizing in the retail setup.

IV. CONCLUSIONS

We present a task-planning framework that combines a pre-trained LLM & VLM to generate and refine action plans without prior domain knowledge. The LLM proposes action sequences and parameters, while feedback from execution failures enables iterative re-planning. Simulated experiments show that, despite requiring multiple refinements, the system effectively handles order-packing tasks and demonstrates strong potential for retail restocking applications.

REFERENCES

[1] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first amazon picking challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2016.

[2] K. Wada, "New robot technology challenge for convenience store," in *2017 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2017, pp. 1086–1091.

[3] A. Guha and D. Grewal, "How robots will affect the future of retailing," *AMS Review*, vol. 12, no. 3, pp. 245–252, 2022.

[4] P. Haslum and A. B. Corrêa, "The universal pddl domain," *arXiv preprint arXiv:2411.08040*, 2024.

[5] A. Kattepur and B. Purushotaman, "Roboplanner: a pragmatic task planning framework for autonomous robots," *Cognitive Computation and Systems*, vol. 2, no. 1, pp. 12–22, 2020.

[6] S. Gupta, K. Yao, L. Niederhauser, and A. Billard, "Action contextualization: Adaptive task planning and action tuning using large language models," *IEEE Robotics and Automation Letters*, 2024.

[7] Y. Chen, J. Arkin, C. Dawson, Y. Zhang, N. Roy, and C. Fan, "Autotamp: Autoregressive task and motion planning with llms as translators and checkers," in *2024 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2024, pp. 6695–6702.

[8] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International conference on machine learning*. PMLR, 2022, pp. 9118–9147.

[9] V. Bhat, A. U. Kaypak, P. Krishnamurthy, R. Karri, and F. Khorrami, "Grounding llms for robot task planning using closed-loop state feedback," *arXiv preprint arXiv:2402.08546*, 2024.

[10] N. George, S. Saha, S. Parab, V. Vakharia, R. Lima, V. Vatsal, and K. Das, "System for autonomous management of retail shelves using an omnidirectional dual-arm robot with a novel soft gripper," in *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2024, pp. 2631–2637.

[11] R. Lima, S. Saha, N. George, V. Vakharia, S. Parab, S. Gaonkar, V. Vatsal, and K. Das, "Teleoperated omni-directional dual arm mobile manipulation robotic system with shared control for retail store," in *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2024, pp. 2935–2942.

[12] N. George and V. Vatsal, "Modeling of soft robotic grippers for reinforcement learning-based grasp planning in simulation," in *2023 Ninth Indian Control Conference (ICC)*, 2023, pp. 287–292.

[13] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024. [Online]. Available: https://arxiv.org/abs/2401.04088

[14] P. Agrawal, S. Antoniak, E. B. Hanna, B. Bout, D. Chaplot, J. Chudnovsky, D. Costa, B. De Monicault, S. Garg, T. Gervet, *et al.*, "Pixtral 12b," *arXiv preprint arXiv:2410.07073*, 2024.

[15] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.