Open-ended planning and grasping for robotic manipulation tasks using lightweight LLMs

Émiland Garrabé, Julien Gleyze, Mahdi Khoramshahi, Stéphane Doncieux*

Abstract— **Owing to their very large training datasets, foundation models such as Large Language Models (LLMs) are well-suited for open semantic interactions, making them attractive tools for robotic manipulation. However, many state-of-the-art techniques only focus on a small subset of the challenges presented by robotic manipulation, and end-to-end solutions such as Vision-Language-Action models require prohibitively large amounts of real-world data for training. In this work, we introduce the language-based components of a manipulation pipeline designed to follow free-form instructions. A modular planning element, based on lightweight LLMs, splits the task into subgoals and leverages a library of robotic primitives to follow the resulting plan. We also propose an open-vocabulary object segmentation method, designed to facilitate task-oriented object grasping. While each component is independently powerful, we provide examples of the whole pipeline being used for real-world manipulation tasks in two settings.**

I. Introduction

Large language models are promising tools for enabling robots to evolve in open environments following open instructions. Owing to their training on internet-scale data, they are able to carry out a range of tasks, but significant challenges remain in bridging the semantic gap between high-level instructions and low-level robot actions.

A. Related works

In robotics, the common sense of LLMs can manifest as an ability to follow instructions using pre-defined motion primitives [1], [2], for example to generate trajectories to train multitask policies [3]. Foundation models can also be used for reward design, either directly writing reward signals [4] or generating intermediate representations such as images [5]. Efforts have been made to leverage the semantic knowledge within foundation models in the context of grasping, but such methods rely on large, human-annotated datasets [6] or fail to account for the subsequent task [7].
Vision-language-action (VLA) models are a rapidly growing paradigm in robotics. The key idea of VLAs is to directly predict actions from the task statement and robot camera(s) output [8]. While early VLA models were trained from scratch, using language models as a backbone has been gaining traction, with the goal of leveraging such models' general knowledge [9], or even retaining multimodal understanding abilities [10]. VLAs have shown impressive results on challenging manipulation tasks, but generalization remains a challenge and they often rely on fine-tuning [11], requiring large datasets.

B. Contributions

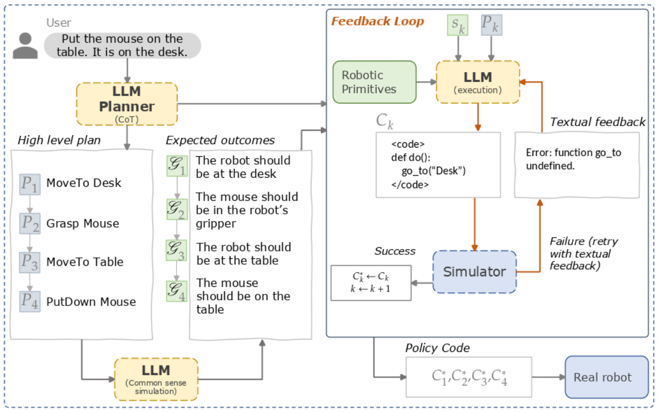
Many of the works above rely on very large models and/or high quantities of real-world data. In this work, we show the following: (i) light-weight language models can be efficient tools for robotic planning, provided they are used within an architecture designed to palliate their lower performance; (ii) world knowledge embedded in foundation models can be used for grasp segmentation and selection, exploiting the diversity afforded by some in-silico methods and (iii) combined together and with access to appropriate robotic skills, these components can be used as the semantic backbone of a manipulation pipeline, whose online compute requirements can be satisfied with a standard laptop.

II. Methods

Language-driven planning: We propose a modular architecture (see Figure 1), leveraging small language models while, by design, alleviating common LLM shortcomings in robotics. For more detail on such failures, refer to [12]. The first element of the architecture is the so-called *planning module*. This module receives the free-form task instruction and decomposes it into steps, formulated in language. This approach is, in part, inspired by the chain-of-thought method: it is easier to execute the task and react to failures when acting step by step. Then, a second LLM module explicits the *expected outcome* of each step. On top of generally improving final code quality [12], this module can be used to systematically include task-specific information such as, for manipulation tasks, which subpart of an object should be grasped. Finally, both the plan and the expected outcomes are sent to the *execution module*. This module has access to the primitives available to the robot, and is responsible for producing code solving each step plan using the robot's skills. The code is verified on a logical twin of the scene, where simple rules filter out common mistakes, and is then executed on the robot. When available, real-world perception primitives can be used as another feedback source. In our experiments, we use either llama3.1 (8bn parameters) or qwen3 (4bn parameters) for all the components of the architecture, allowing us to rely on a standard laptop.

Task-aware grasp selection: Quality-diversity algorithms are well-suited to generate diverse archives of grasps givne an object model [13]. In open-ended manipulation, selecting the optimal grasp based on the task is crucial. Our open-vocabulary segmentation pipeline for 3D objects, based on off-the-shelf components, is as follows (see Figure 2). First, the object's model is rendered as an image using principal component analysis. Then, the image is segmented using the

*All authors at ISIR, Sorbonne Université, Paris; Corresponding email: garrabe@isir.upmc.fr


User
Put the mouse on the table. It is on the desk.
LLM Planner (CoT)
High level plan
P_1 MoveTo Desk
P_2 Grasp Mouse
P_3 MoveTo Table
P_4 PutDown Mouse
Expected outcomes
$\mathcal{G}_1$ The robot should be at the desk
$\mathcal{G}_2$ The mouse should be in the robot's gripper
$\mathcal{G}_3$ The robot should be at the table
$\mathcal{G}_4$ The mouse should be on the table
LLM (Common sense simulation)
Feedback Loop
s_k
P_k
Robotic Primitives
LLM (execution)
C_k
Textual feedback
Error: function go_to undefined.
Success
$C_k^* \leftarrow C_k$
$k \leftarrow k+1$
Simulator
Failure (retry with textual feedback)
Policy Code
$C_1^*, C_2^*, C_3^*, C_4^*$
Real robot

```
<code>
def do():
   go_to("Desk")
</code>
```

Fig. 1: [12] Modular architecture for task planning and execution.

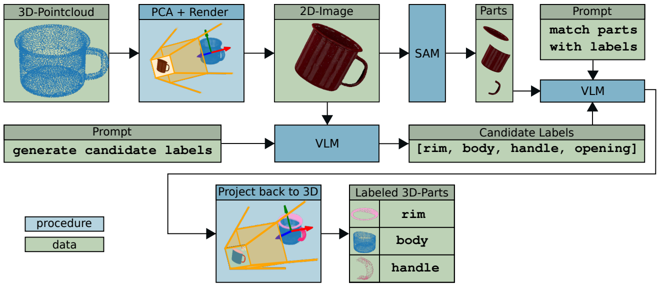


Fig. 2: [15] Open-ended object segmentation pipeline

SAM model [14] and the segmented subparts are labelled using a vision-language model. The 2-dimensional segmentation map is projected in 3D, and each grasps is labelled based on the object subpart in contact with the gripper. In standalone use, an LLM component infers a cost function based on the desired subpart to be grasped, while when in our planner the grasping skill is parametrized to allow the execution module to specify which subpart to grasp.

III. Experiments and Results

In this section, we showcase examples of our pipeline being used for 2 manipulation tasks, namely robothon task board manipulation [16] and pick-and-place, using the Franka FR3 robot. For more thorough evaluation of the modules, we refer to their respective papers [12], [15].

Robothon task board: The robothon task board is designed to simulate industrial manipulation of electronic devices. In this task, the motion primitives consist of pressing the two buttons, plugging the cable into a socket and opening the trapdoor (see Figure 3). The cable-plugging skill was obtained by using our planning architecture and a task-oriented grasp on the cable housing, showcasing the compatibility of our approach with curriculum-style methods. The task board manipulation is carried out in real time, due to the fast inference time of smaller LLMs. For a video, see https://tinyurl.com/4h2ysybd.

Pick-and-place task: We deploy our segmentation pipeline on common-object pick-and-place tasks. Here, the motion primitives are a grasp and a place skill. We deploy the pick-and-place motions on 4 YCB objects, alternating between subparts. See Figure 2 for pictures.

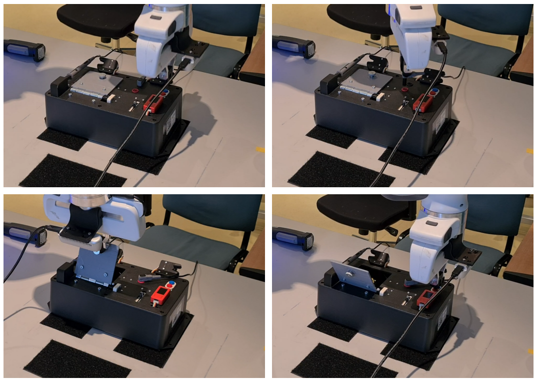


Fig. 3: Task board manipulation. The skills are executed in sequence, and some of them were designed using a task-aware grasp.

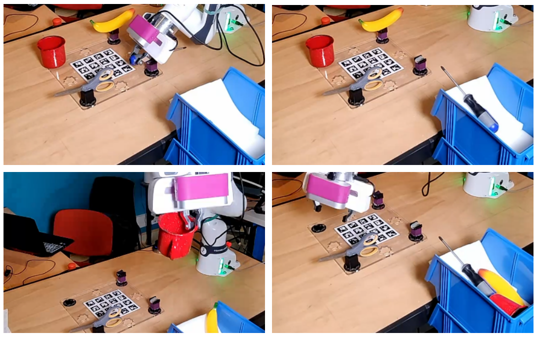


Fig. 4: Pick-and-place tasks. Top panel: pick-and-place of the screwdriver, with the handle being the optimal subpart to grasp for placing. Bottom panel: two grasps on the mug subparts 'body' and 'handle'. The objects are placed on an Aruco board for pose estimation.

IV. Concluding Remarks and Future Work

We presented the language-based components of our manipulation architecture. Namely, we introduced a first module designed to plan and execute tasks using a light-weight, local language model, leading to low compute requirements [12]. Then, we introduced a task-oriented segmentation mechanism that labels grasps from an archive to allow for task-oriented grasp selection. We showed the pipeline executing manipulation tasks given free-form inputs.

Future work directions include adding richer feedback mechanisms between the real world and the planning component to allow the pipeline to retry when primitive failure occurs, using language-based reward shaping methods to autonomously acquire missing motion primitives, and exploring how the grasp segmentation pipeline can be used to produce high-quality real-world grasping trajectory data.

Acknowledgements

This work was supported by the European Union Horizon Europe Framework Programme, through the PILLAR (grant agreement 101070381) and euROBIN (grant agreement 101070596) projects.

References

[1] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[2] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 523–11 530.

[3] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Conference on Robot Learning*. PMLR, 2023, pp. 3766–3777.

[4] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," 2024. [Online]. Available: https://arxiv.org/abs/2310.12931

[5] Z. Chen, J. Huo, Y. Chen, and Y. Gao, "Robohorizon: An llm-assisted multi-view world model for long-horizon robotic manipulation," *arXiv preprint arXiv:2501.06605*, 2025.

[6] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, "Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping," 2023. [Online]. Available: https://arxiv.org/abs/2307.13204

[7] R. Mirjalili, M. Krawez, S. Silenzi, Y. Blei, and W. Burgard, "Lan-grasp: Using large language models for semantic object grasping," 2023. [Online]. Available: https://arxiv.org/abs/2310.05239

[8] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[9] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "OpenVLA: An open-source vision-language-action model," 2024. [Online]. Available: https://arxiv.org/abs/2406.09246

[10] Z. Zhou, Y. Zhu, M. Zhu, J. Wen, N. Liu, Z. Xu, W. Meng, R. Cheng, Y. Peng, C. Shen *et al.*, "ChatVLA: Unified multimodal understanding and robot control with vision-language-action model," *arXiv preprint arXiv:2502.14420*, 2025.

[11] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[12] Émiland Garrabé, P. Teixeira, M. Khoramshahi, and S. Doncieux, "Enhancing robustness in language-driven robotics: A modular approach to failure reduction," 2025. [Online]. Available: https://arxiv.org/abs/2411.05474

[13] J. Huber, F. Hélénon, M. Kappel, E. Chelly, M. Khoramshahi, F. B. Amar, and S. Doncieux, "Speeding up 6-dof grasp sampling with quality-diversity," 2024. [Online]. Available: https://arxiv.org/abs/2403.06173

[14] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023. [Online]. Available: https://arxiv.org/abs/2304.02643

[15] A. X. Appius, E. Garrabe, F. Helenon, M. Khoramshahi, M. Chetouani, and S. Doncieux, "Task-aware robotic grasping by evaluating quality diversity solutions through foundation models," 2025. [Online]. Available: https://arxiv.org/abs/2411.14917

[16] P. So, A. Sarabakha, F. Wu, U. Culha, F. J. Abu-Dakka, and S. Haddadin, "Digital robot judge: Building a task-centric performance database of real-world manipulation with electronic task boards," *IEEE Robotics & Automation Magazine*, 2024.