From Language to Action with Object-Level Planning

David Paulius[1*], Alejandro Agostini[2], George Konidaris[1]

***Abstract*—Foundation models, such as large language models (LLMs) and vision-language models (VLMs) contain a wide breadth of domain knowledge useful to robotic tasks, specifically for planning. In terms of robot task planning, existing work uses language models to either directly output task plans or to generate planning definitions in representations like PDDL. However, we have recently shown that an LLM is best suited for *object-level planning*, where knowledge is extracted from an LLM and structured into an object-level representation (as a functional object-oriented network or FOON for short) to generate PDDL subgoals. This work briefly summarizes the current state of our work that interfaces object-level planning for task and motion planning (TAMP) while also discussing further opportunities to improve this planning approach with language models.**

I. Introduction

Motivated by the advent of *foundation models* like large language models (LLMs) and vision-language models (VLMs), contemporary research aims to exploit their capabilities for a variety of tasks, including planning for robots and embodied agents [1, 2, 3, 4]. Language models encode domain knowledge about the world, which is useful for language-conditioned or language-guided decision-making. However, our most recent work has shown that state-of-the-art approaches are not suited for handling complex goal-oriented tasks at the task level [5]. Some approaches position LLMs either as task planners [2, 3, 6], depriving such methods of the guarantees promised by classical planning (viz. optimality and completeness), or as task description generators [7, 8, 9]. These approaches fail to generate plan specifications that are guaranteed to work due to the LLM's lack of embodiment. We propose to tackle these limitations by articulating language models and TAMP using *object-level planning* (OLP) [10], which focuses on object state transitions at the object level [11]. Object-level plans, like recipes in a cookbook, are agnostic to the robot and its environment; instead, they provide object constraints, such as how object properties change when combined with other objects, rather than motion constraints.

Our previous work [12] has shown how object-level plans simplify complex long-horizon tasks by decomposing them into sub-problems that can be quickly and effectively executed using TAMP. This is due to an object-centric representation compatible with both OLP and TAMP. On the one hand, this representation encodes relevant changes in the object space for OLP. On the other hand, it allows for encoding abstraction of motion constraints for TAMP [13, 14]. With an object-centric representation, actions can be easily mapped to initial states and goals that are compatible with TAMP via PDDL [15] in a hierarchical planning approach. Our most recent work [5] generates object-level plans (via language model prompting) to realize robot execution using our framework (see Figure 1).

Object-level planning acts as an interface between human language and TAMP via an object-level representation (OLR) called the *functional object-oriented network* (FOON) [16]. Our prior work has demonstrated how object-level knowledge in FOON can automatically generate PDDL subgoals [12]. However, this method assumes that we have a collection of partial plans specified as FOON graphs. This poses the question of *how* we can acquire object-level plans that can be used to bootstrap such methods. Previous work predating foundation models has shown how to extract FOON object-level plans directly from videos [17]; more recently, we have explored how we can exploit language models to generate FOONs compatible with task and motion planning [5], providing an appealing alternative to learning FOONs from video. Ultimately, this approach overcomes the inability of LLMs to directly output feasible task plans while exploiting the higher, object-level nature of LLM output and language as a whole. This paper summarizes the relevant aspects of our recent contributions to articulate language models and TAMP using object-level planning [12, 5], and describes the ongoing and future lines of research that enable generalization and portability in robot domains.

II. Experiments

In our recent contributions, we have demonstrated the validity of our approaches through several experiments, for which we describe each setting as well as key insights and findings. We conducted several experiments in a simulated table-top environment in CoppeliaSim [18] with a robot arm affixed to the table upon which objects are initialized and randomly configured. We use Fast Downward [19], an off-the-shelf PDDL solver, for task-level planning in our method as well as competing baselines. We provide further details in the following subsections.

A. Object-level Planning for Bootstrapping TAMP

In our first set of experiments [12], we defined two complex long-horizon cooking scenarios: *Bloody Mary cocktail* and *Greek salad* preparation.[1] Our experiments have shown that our method enables a robot to successfully execute each task with 96% and 80% success for both tasks, where completing each task on average requires the successful execution of 28 and 35 actions respectively. Moreover, we have demonstrated that we can flexibly generate varying task (or micro-) plans for

[1]Brown University, Providence, RI, USA.
[2]University of Innsbruck, Innsbruck, Austria.
*Corresponding Author (*Email*: dpaulius@cs.brown.edu)

[1]Demonstration videos for Paulius and Agostini [12] can be found on the project's webpage: https://davidpaulius.github.io/foon-lhpe/

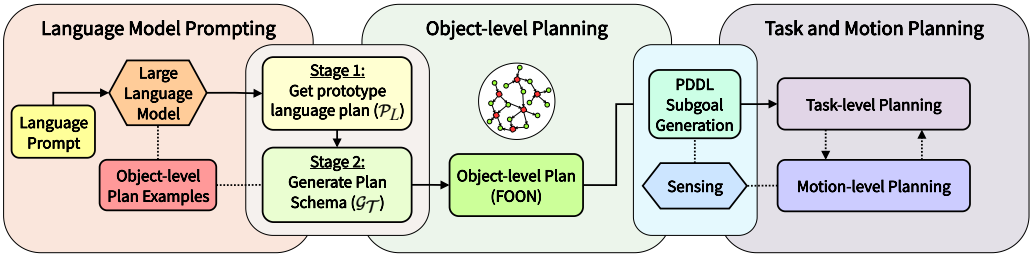

Fig. 1. Our most recent work interfaces with a language model to generate object-level plans to bootstrap task and motion planning [5]. Our approach generates task-level subgoals as PDDL definitions by grounding object-level subgoals to a robot's environment; with these task-level definitions, we perform task planning to obtain task plan segments per object-level action, which we execute using motion-level planning, improving upon previous work [12].

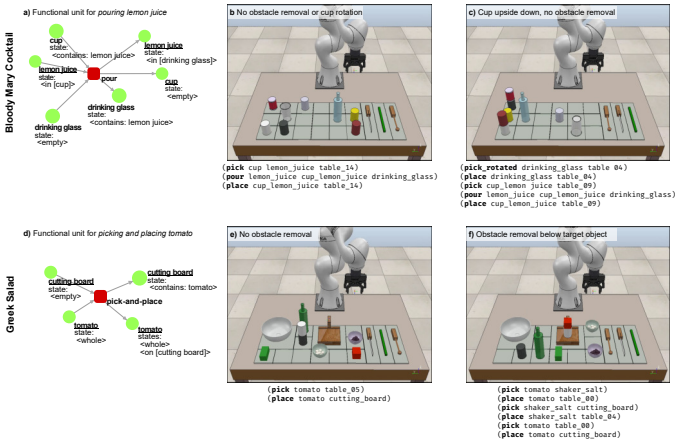

Fig. 2. Illustration from prior work [12] showing how an object-level action can be resolved by different task plans depending on the environment state.

the same object-level (or macro-) plan (see Figure 2), which mirrors the way we humans can execute recipes in varying ways depending on the state at run time. Additionally, we have compared our method against hierarchical task networks (HTNs) [20] and regular planning without object-level subgoals (provided by functional units) and show a better time complexity than said alternative methods.

B. Exploiting LLMs for Object-level Planning

In our second set of experiments [5], we evaluate our LLM-based OLP approach on three tasks of increasing difficulty: 1) *tower building*, where a robot must assemble a tower of blocks of a given height; 2) *spelling*, where a robot must construct a tower that spells a given word; and 3) *organizing table*, where a robot must place all alike blocks into piles.[2] We compare our OLP-based method to several baseline methods: LLM-Planner, LLM+P [7] and DELTA [9]. Following the previously introduced tracks of LLM-based planning work, the LLM-Planner baseline uses a LLM to directly output a task plan, given a textual description of the robot's environment state and allowable actions, while the LLM+P and DELTA baselines use a LLM to directly generate PDDL definitions, given PDDL examples and a textual description of the robot's environment state. All methods used Chat-GPT [21]. We have demonstrated how our OLP approach result in more complete executions of tasks (86%, 80%, and 81% for all three tasks, respectively) while jointly improving time complexity over baseline methods and reducing the number of tokens generated by the LLM (especially when compared to the best competing baseline DELTA [9]). Although LLM-Planner generates task plans without a solver, it does not complete a majority of tasks because the LLM has poor understanding of the configuration of the robot's environment for collision-free motion. Baselines that generate PDDL file definitions are also prone to issues due to the LLM's inability to consistently generate correct or accurate files.

[2]Project Website for Paulius et al. [5]: https://davidpaulius.github.io/olp_llm/

III. Conclusion

This work briefly reviews our recent contributions to hierarchical planning that integrate an additional planning layer situated above TAMP known as *object-level planning* [10]. This enables robots to flexibly find planning solutions from plan sketches [12] (object-level plans) that can be extracted via LLM prompting [5]. When compared to alternative LLM-based planning approaches that either use a LLM as a planner or as a generator of planning definitions like PDDL, our method flexibly enables a robot to solve a wide range of tasks that leverage the expressiveness of natural language. Finally, we have also demonstrated how object-level planning allows a robot to flexibly obtain task plans for the same object-level subgoals, and that the subgoals provided by an object-level plan aid to improve time complexity in computation.

Acknowledgements

The work discussed in this article was graciously supported by the Office of Naval Research (ONR) through grant number N00014-21-1-2584, Echo Labs, the Helmholtz Association, and the Austrian Science Fund (FWF) under Projects M2659-N38 and P36965. This work was supported by the Office of Naval Research (ONR) under REPRISM MURI N000142412603, ONR grants N00014-21-1-2584 and N00014-22-1-2592, Echo Labs, the Helmholtz Association, and the Austrian Science Fund (FWF) under Projects M2659-N38 and P36965. Partial funding was also provided by The Robotics and AI Institute (formerly "The AI Institute").

References

[1] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 162, 2022, pp. 9118–9147.

[2] B. Ichter, A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, D. Kalashnikov, S. Levine, Y. Lu, C. Parada, K. Rao, P. Sermanet, A. T. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, M. Yan, N. Brown, M. Ahn, O. Cortes, N. Sievers, C. Tan, S. Xu, D. Reyes, J. Rettinghouse, J. Quiambao, P. Pastor, L. Luu, K.-H. Lee, Y. Kuang, S. Jesmonth, N. J. Joshi, K. Jeffrey, R. J. Ruano, J. Hsu, K. Gopalakrishnan, B. David, A. Zeng, and C. K. Fu, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," in *Proceedings of the 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 205. PMLR, 14–18 Dec 2023, pp. 287–318.

[3] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "PaLM-E: An Embodied Multimodal Language Model," in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2023, pp. 8469–8488.

[4] S. S. Raman, V. Cohen, I. Idrees, E. Rosen, R. Mooney, S. Tellex, and D. Paulius, "CAPE: Corrective Actions from Precondition Errors using Large Language Models," in *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 070–14 077.

[5] D. Paulius, A. Agostini, B. Quartey, and G. Konidaris, "Bootstrapping Object-level Planning with Large Language Models," in *Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[6] 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 *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 523–11 530.

[7] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "LLM+P: Empowering Large Language Models with Optimal Planning Proficiency," *arXiv preprint arXiv:2304.11477*, 2023.

[8] Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh, "Translating Natural Language to Planning Goals with Large-Language Models," *arXiv preprint arXiv:2302.05128*, 2023.

[9] Y. Liu, L. Palmieri, S. Koch, I. Georgievski, and M. Aiello, "DELTA: Decomposed Efficient Long-Term Robot Task Planning using Large Language Models," *arXiv preprint arXiv:2404.03275*, 2024.

[10] D. Paulius, "Object-Level Planning and Abstraction," in *CoRL 2022 Workshop on Learning, Perception, and Abstraction for Long-Horizon Planning*, 2022.

[11] O. Kroemer, S. Niekum, and G. Konidaris, "A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms," *Journal of Machine Learning Research*, vol. 22, no. 30, pp. 1–82, 2021.

[12] D. Paulius, A. Agostini, and D. Lee, "Long-Horizon Planning and Execution with Functional Object-Oriented Networks," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4513–4520, 2023.

[13] A. Agostini, M. Saveriano, D. Lee, and J. Piater, "Manipulation Planning Using Object-Centered Predicates and Hierarchical Decomposition of Contextual Actions," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5629–5636, 2020.

[14] A. Agostini and J. Piater, "Unified Task and Motion Planning using Object-centric Abstractions of Motion Constraints," *arXiv preprint arXiv:2312.17605*, 2023.

[15] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "PDDL – The Planning Domain Definition Language," CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, Tech. Rep., 1998.

[16] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun, "Functional Object-Oriented Network for Manipulation Learning," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2655–2662.

[17] A. B. Jelodar, D. Paulius, and Y. Sun, "Long Activity Video Understanding Using Functional Object-Oriented Network," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1813–1824, July 2019.

[18] E. Rohmer, S. P. N. Singh, and M. Freese, "CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework," in *Proceedings of the 2013 International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1321–1326, http://www.coppeliarobotics.com.

[19] M. Helmert, "The Fast Downward Planning System," *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.

[20] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning and Acting*. Cambridge University Press, 2016.

[21] OpenAI, "GPT-4 Technical Report," 2023, accessed the model on April 11, 2025.