LLM-Planner: Few-Shot Grounded Planning for Embodied Agents with Large Language Models

Anonymous CVPR submission

Paper ID 7

Abstract

In this work, we propose a novel method, LLM-Planner, that harnesses the power of large language models to do few-shot planning for embodied agents. We further propose a simple but effective way to enhance LLMs with physical grounding to generate and update plans that are grounded in the current environment. Experiments on the ALFRED dataset show that our method can achieve very competitive few-shot performance: Despite using less than 0.5% of paired training data, LLM-Planner achieves competitive performance with recent baselines that are trained using the full training data. Existing methods can barely complete any task successfully under the same few-shot setting. Our work opens the door for developing versatile and sample-efficient embodied agents that can quickly learn many tasks.

1. Introduction

Contemporary language-driven agents still require a large number of labeled examples (pairs of language instructions and gold trajectories) to learn each task, which is highly costly and hinders the development of truly versatile agents [2, 6, 7, 10, 11, 14, 16, 17, 19, 22, 24]. Recently, an array of seminal work has shown the remarkable potential of large language models (LLMs) such as GPT-3 [4] as a few-shot planner for embodied AI agents [1, 8, 12, 20]. Agents equipped with LLM-based planners have started to show the ability to learn a new task with a few training examples.

While showing great promises as proof of concepts, existing work still presents significant limitations that may prevent larger-scale applications beyond their limited evaluation setting. As an example, SayCan [1], one of the pioneering work on using LLMs for embodied instruction following, is evaluated on two environments with only 15 object types. The agent is assumed to be able to enumerate all admissible skills (*i.e.*, [action, object] pairs) up front so it can use an LLM to rank the skills. This assumption could break easily in partially-observable environments when deploying an agent to new environments. The cost could also quickly pile up in more complex environments with more objects because the agent needs to call the LLM to evaluate every admissible skill at every step; efficiency deteriorates at the same time.

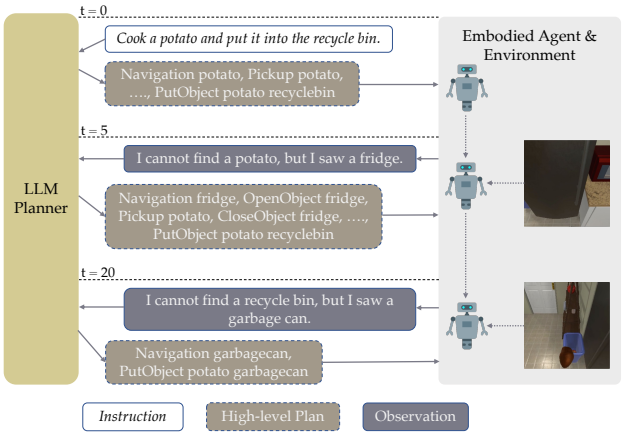

Figure 1. An illustration of LLM-Planner for high-level planning.

Finally, most existing work [1, 8, 13, 20] uses LLMs to generate a single static plan from the language instruction and then executes on the entire plan. However, the optimal plan for the same language instruction is dependent on the environment; different environments may need different plans (Figure 1).

We propose *LLM-Planner*, an LLM-based planner for embodied instruction following. An important design goal is to be able to directly generate plans in diverse, partially-observable environments, and can dynamically replan based on perceptions from the environment.

While most existing work [1, 8, 9, 13, 20] is evaluated under a limited setting (*e.g.*, limited/known environments, short-horizon tasks, or simple environments with a small number of objects), we evaluate LLM-Planner on ALFRED [19], a large-scale dataset with diverse partially-observable environments and a wide variety of tasks and objects. Using less than 0.5% of paired training data, LLM-Planner achieves competitive performance compared with HLSM and outperforms multiple other baselines, which are trained with the full training set. Under the same few-shot

setting, existing methods can barely complete any task successfully.

2. LLM-Planner

We adopt hierarchical planning models (e.g., [18, 23]), which consist of a *high-level planner* and a *low-level planner*. We use LLMs to generate high-level plans (HLPs), *i.e.*, a sequence of subgoals (*e.g.*, [Navigation potato, Pickup potato, Navigation microwave, ...]) that the agent needs to achieve, in the specified order, to accomplish the final goal specified by the language instruction. The low-level planner then maps each subgoal into a sequence of primitive actions for achieving that subgoal in the current environment and state.

To adapt LLMs such as GPT-3 as high-level planners, the first step is to design an appropriate prompt to guide them to generate high-level plans. We identify core components of the prompt and systemically compare different design choices under the true few-shot setting based on leave-one-out cross-validation (LOOCV). The prompt begins with an intuitive explanation of the task and the list of allowable high-level actions. It is then followed by the in-context examples which are the most similar samples in training dataset to the current test example, selected by the k-nearest-neighbor (kNN) retriever. With all the above designs, we have obtained the *static* version of LLM-Planner, which can already generate reasonable HLPs.

Furthermore, we equip LLM-Planner with a grounded re-planning capability to dynamically update the HLP during the course of completing a task. This is in contrast with most existing work that only predicts a fixed HLP up front and sticks to that no matter what happens during the execution. To this end, we add the subgoals that have been completed and the list of objects observed by object detector so far in the prompt. We also add logit biases to these observed objects so LLM-Planner can prioritize producing a plan with those objects if they are relevant for the task. We trigger re-planning under either of two conditions: 1) the agent fails to execute an action, or 2) after a fixed number of time steps.

3. Experiments and Results

We use the same evaluation setup and metrics provided by ALFRED [19]. For the low-level controller, we use the HLSM [3]'s low-level controller. We also implement SayCan [1] in ALFRED to compare with our method. To make it possible for SayCan to work in the complex, partially-observable environments in ALFRED, we give it an *unfair competitive advantage*—it knows all the objects and affordances in the current environment *a priori* to compile the list of skills. The main results are shown in Table 1. We find that LLM-Planner's few-shot performance is competitive to the original HLSM, and outperforms a recent baseline such as M-TRACK, despite using less than 0.5% of paired training data. On the other hand, when trained using the same 100 examples (*i.e.*, re-training HLSM's high-level planner), HLSM (and FILM as well) can barely complete any task successfully. Furthermore, the results show that SayCan still largely underperforms LLM-Planner despite the access to the full environment information. Another significant difference is *cost and efficiency*. Because of SayCan's ranking nature, it needs to call the LLM many more times than a generative model like LLM-Planner: *LLM-Planner calls GPT-3 avg. 7 times per task and SayCan calls it 22 times*, even with oracle knowledge of the current environment to shrink the skill list. Lastly, we see a considerable improvement from grounded re-planning over static planning, especially in the goal instruction only setting, where it improves 1.83% SR in the unseen test split. This confirms the effectiveness of the grounded re-planning. But we also note that there is still a large room for further improvement.

Model	Test Unseen		Valid Unseen	
	SR	GC	SR	GC
Full-data setting: 21,023 (instruction, trajectory) pairs				
Goal instruction only				
HLSM [3]	20.27	27.24	**18.28**	**31.24**
Step-by-step instructions				
M-TRACK [21]	16.29	22.60	17.29	28.98
FILM [15]	**27.80**	**38.52**	–	–
Few-shot setting: 100 (instruction, high-level plan) pairs				
Goal instruction only				
LLM-Planner (Static) + HLSM	11.58	18.47	11.10	22.44
LLM-Planner + HLSM	13.41	22.89	12.92	25.35
Step-by-step instructions				
HLSM [3]	0.61	3.72	0.00	1.86
FILM [15]	0.20	6.71	0.00	9.65
SayCan [1]	-	-	9.88	22.54
LLM-Planner (Static) + HLSM	15.83	20.99	14.26	26.12
LLM-Planner + HLSM	**16.42**	**23.37**	**15.36**	**29.88**

Table 1. Main results on the ALFRED dataset. "(Static)" means the static planning setting, otherwise it is the default dynamic setting with grounded re-planning.

4. Conclusion and Future Work

Our work can dramatically reduce the amount of human annotations needed for learning the instruction following task. Furthermore, it opens a new door for developing versatile and extremely sample-efficient embodied agents by harnessing the power of large language models and enhancing them with physical grounding. Promising future directions include exploring other LLMs such as PaLM [5], better prompt design, and more advanced methods for grounding and dynamic re-planning.

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 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. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022. 1, 2

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018. 1

[3] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. A persistent spatial semantic representation for high-level natural language instruction execution. In *Conference on Robot Learning*, pages 706–717. PMLR, 2022. 2

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. 1

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 2

[6] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Neural Information Processing Systems (NeurIPS)*, 2018. 1

[7] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1653, June 2021. 1

[8] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*, 2022. 1

[9] 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. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022. 1

[10] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2020. 1

[11] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah A. Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1494–1499, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. 1

[12] Xiaotian Liu, Hector Palacios, and Christian Muise. A planning based neural-symbolic approach for embodied instruction following. *Interactions*, 9(8):17, 2022. 1

[13] Yujie Lu, Weixi Feng, Wanrong Zhu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. Neuro-symbolic procedural planning with commonsense prompting, 2022. 1

[14] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI*, pages 259–274, 2020. 1

[15] So Yeon Min, Devendra Singh Chaplot, Pradeep Kumar Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. FILM: Following instructions in language with modular methods. In *International Conference on Learning Representations*, 2022. 2

[16] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic Transformer for Vision-and-Language Navigation. In *ICCV*, 2021. 1

[17] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8494–8502, 2018. 1

[18] Pratyusha Sharma, Antonio Torralba, and Jacob Andreas. Skill induction and planning with latent language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1713–1726, Dublin, Ireland, May 2022. Association for Computational Linguistics. 2

[19] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 1, 2

[20] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. ProgPrompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022. 1

[21] Chan Hee Song, Jihyung Kil, Tai-Yu Pan, Brian M. Sadler, Wei-Lun Chao, and Yu Su. One step at a time: Long-horizon vision-and-language navigation with milestones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15482–15491, June 2022. 2

[22] Alessandro Suglia, Qiaozi Gao, Jesse Thomason, Govind Thattai, and Gaurav Sukhatme. Embodied bert: A transformer model for embodied, language-guided visual task completion, 2021. 1

[23] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999. 2

[24] Yichi Zhang and Joyce Chai. Hierarchical task learning from language instructions with unified transformers and self-monitoring. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4202–4213, Online, Aug. 2021. Association for Computational Linguistics. 1