OPEx: A Large Language Model-Powered Framework for Embodied Instruction Following

Extended Abstract

Haochen Shi
Université de Montréal & Mila
Montréal, Canada
haochen.shi@umontreal.ca

Zhiyuan Sun
Université de Montréal & Mila
Montréal, Canada
zhiyuan.sun@umontreal.ca

Xingdi Yuan
Microsoft Research
Montréal, Canada
eric.yuan@microsoft.com

Marc-Alexandre Côté
Microsoft Research
Montréal, Canada
macote@microsoft.com

Bang Liu
Université de Montréal & Mila
Montréal, Canada
bang.liu@umontreal.ca

ABSTRACT

Embodied Instruction Following (EIF) is crucial for understanding natural language in a practical context, requiring agents to follow verbal instructions for complex tasks. Traditionally, EIF relies heavily on expert annotations for learning, which are costly and sometimes unattainable. Recent research shows Large Language Models (LLMs) can use their reasoning ability to help in EIF with minimal examples, but applying LLMs directly faces issues like hallucinations and partially observable environment. To bridge the gap, we introduce OPEx, a new LLM-based method for EIF that needs far less specific data. OPEx uses three LLMs for different roles: observing to gather environment data, planning by breaking down instructions, and executing tasks with learned skills. Our tests reveal OPEx significantly outperforms the FILM baseline, with 90% less training data for planning tasks and achieving up to 38% performance gain when FILM is trained on identical data.

KEYWORDS

Embodied Instruction Following; Language Grounding; Large Language Models; Grounded Planning; In Context Learning

ACM Reference Format:
Haochen Shi, Zhiyuan Sun, Xingdi Yuan, Marc-Alexandre Côté, and Bang Liu. 2024. OPEx: A Large Language Model-Powered Framework for Embodied Instruction Following: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

The creation of autonomous agents requires integrating extensive planning with precise execution, a challenge that deep learning advancements are helping to overcome [1, 7, 8, 11]. Embodied Instruction Following (EIF) has become a key area of focus, necessitating agents to follow natural language instructions through egocentric observations [19]. Traditional EIF methods rely heavily on expert annotations, which are costly and sometimes impractical. Large Language Models (LLMs) present a promising solution, trained on extensive data to exhibit common-sense reasoning [5, 16, 21, 22], but direct application to EIF faces challenges like environmental unpredictability and the need for adaptation.

To address these issues, we introduce OPEx (Observer & Planner & Executor), a novel LLM-centric framework for EIF that dynamically integrates planning and action. The Planner uses LLMs for task decomposition, the Observer updates with environmental feedback, and the Executor translates the plans into actionable steps, using a skill set to guide the agent in its tasks. OPEx demonstrates significant improvements on the ALFRED benchmark [19], achieving over 10% absolute performance gains over the baseline FILM [11], requiring 90% less training data. Besides, it achieves up to 38% absolute performance gain when FILM is trained on identical data.

2 THE OPEX FRAMEWORK

The OPEx framework introduces a novel approach for Embodied Instruction Following (EIF) with a focus on dynamic task planning and grounding, utilizing Large Language Models (LLMs) for enhanced efficiency and adaptability. Unlike previous methods that depend heavily on static plans and supervised learning, OPEx leverages the reasoning capabilities of LLMs to dynamically decompose tasks, improve grounding, and address the sparse reward problem in EIF without extensive training data or heuristic rules. As shown in Fig. 1, OPEx consists of six main components: (1) *semantic mapping module* converting egocentric visual observations into semantic maps (2) An *LLM-based planner* that decomposes language instructions into subtasks. (3) An *LLM-based observer* that updates the world state in natural language description. (4) An *LLM-based executor* selecting skills to complete subtasks. (5) A *skill library* storing predefined skills for manipulation. (6) A *deterministic action policy* for converting skills into actions.

Semantic Mapping Module. This module creates a 2D semantic map from visual inputs, utilizing UNet [18] for depth mapping and MaskRCNN [6] for instance segmentation, following FILM [11]. To address perceptual noise, a supplementary semantic map M_t' is proposed aggregating information over time and enhancing reliability.

LLM-based Planner. The *LLM-based planner* aims to break down a language instruction into subtasks, leveraging LLMs' reasoning

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

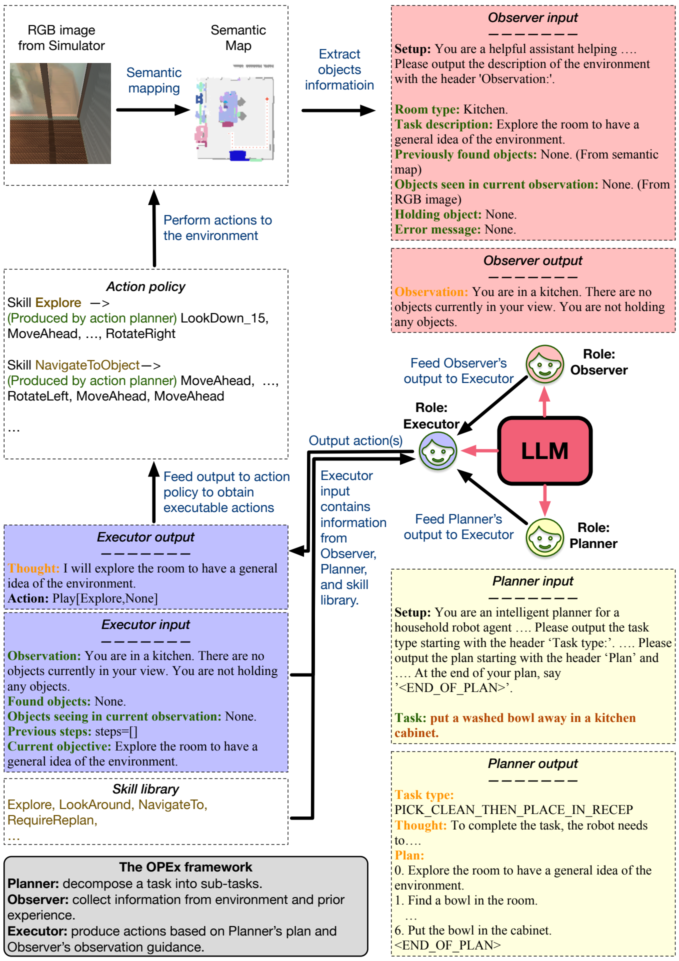

Figure 1: Overview of our OPEx framework.

capability [3]. Utilizing Chain-of-Thought (CoT) prompting and GPT-4, we enhance the planner's reasoning effectiveness through in-context learning [16, 23]. The planner's prompt incorporates a setup phase, with K in-context examples chosen by an example selector. The example selector chooses the most relevant examples for each task by ranking and selecting top-K examples based on the similarity of the input test case and the examples [4, 9].

LLM-based Observer. The LLM-based Observer plays a critical role in the OPEx framework, aiming to interpret environmental feedback and agent states into a concise natural language description using a zero-shot approach. This component utilizes *GPT-3.5* for querying, with a prompt structure designed to capture and articulate the environmental state, thus supporting the monitoring of dynamic changes over time which aids in dynamic planning and execution. Besides, the observer is also supposed to condense the gathered information into a focused description, which helps minimize distractions and hallucinations for the LLM-based executor.

LLM-based Executor. The LLM-based executor plays a pivotal role in the OPEx framework by executing subtasks using a predefined skill library. Unlike the LLM-based planner, the executor is actively involved in the environment, leveraging feedback to understand dynamics and apply the necessary skills to complete tasks.

Method	Test Seen				Test Unseen			
	PLWGC	GC	PLWSR	SR	PLWGC	GC	PLWSR	SR
High-level Goal Instruction + Low-level step-by-step instructions								
Seq2Seq [19]	6.27	9.42	2.02	3.98	4.26	7.03	0.08	3.90
MOCA [20]	22.05	28.29	15.10	22.05	9.99	14.28	2.72	5.30
E.T. [17]	**34.93**	45.44	27.78	38.42	11.46	18.56	4.10	8.57
LWIT [13]	23.10	40.53	**43.10**	30.92	**16.34**	20.91	5.60	9.42
FILM [11]	15.06	38.51	11.23	27.67	14.30	36.37	**10.55**	26.49
OPEx	14.62	**48.74**	9.52	**38.81**	14.45	**49.60**	9.35	**37.15**
High-level goal instructions only								
LAV [15]	13.18	23.21	6.31	13.35	10.47	17.27	3.12	6.38
HLSM [2]	11.53	35.79	6.69	25.11	8.45	27.24	4.34	16.29
FILM [11]	**14.17**	36.15	**10.39**	25.77	13.13	34.75	**9.67**	24.46
OPEx	14.06	**47.81**	9.18	**38.03**	**13.48**	**48.61**	9.08	**35.91**

Table 1: Main Results on the test splits of ALFRED benchmark. The top section uses low-level step-by-step instructions, while the bottom section only uses the high-level goal instruction.

Method	SR	GC	PLWSR	PLWGC
OPEx	38.12	46.13	9.03	13.45
FILM	0.00	12.18	0.00	2.78

Table 2: Performance comparison with the baseline trained on same amount of data.

Inspired by ReAct [24], the executor employs a *GPT-4* model to generate reasoning traces and action plans, enhancing decision-making and interaction with the environment. The executor's operation is guided by prompts designed to solicit both the thought process (reasoning traces) and the specific actions to be taken from the skill library, facilitating a dynamic response to the evolving task environment. This dual-output approach ensures the executor can adapt plans based on real-time feedback and handle unforeseen situations effectively.

Skill Library and Deterministic Action Policy. The skill library equips the executor with capabilities for reasoning and action, including navigation and object interaction skills. The deterministic action policy translates these skills into low-level actions, employing heuristics based on the semantic map.

3 EXPERIMENTS AND DISCUSSION

Experiment Setup. Our approach is evaluated on the ALFRED benchmark [19]. We employ four primary evaluation metrics as established in prior works [11, 19]: Success Rate (SR), Goal Condition (GC), path length weighted SR (PLWSR), and path length weighted GC (PLWGC), with SR in the test unseen split serving as the primary performance indicator.

Compared Methods. The methods compared are categorized based on their reliance on instruction level: (1) methods necessitating detailed step-by-step and high-level instructions [13, 17, 19, 20]; (2) methods operational with only high-level instructions [2, 8, 10–12, 14].

Results Analysis. The main results are shown in Table 1. Remarkably, OPEx leverages in-context learning with less than 10% of the data used for FILM's Language Processor training yet achieves more than 10% in SR on both splits under all the settings. Table 2 shows OPEx's superior performance over FILM in utilizing in-domain data. When FILM is trained on the same data, OPEx demonstrates significant improvements across all metrics.

ACKNOWLEDGMENTS

This work is supported by the Canada CIFAR AI Chair Program and the Canada NSERC Discovery Grant (RGPIN-2021-03115).

REFERENCES

[1] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. 2022. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems* 35 (2022), 24639–24654.

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

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[4] Harrison Chase. 2022. *LangChain.* https://github.com/hwchase17/langchain

[5] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378* (2023).

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[7] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*. PMLR, 9118–9147.

[8] Yuki Inoue and Hiroki Ohashi. 2022. Prompter: Utilizing Large Language Model Prompting for a Data Efficient Embodied Instruction Following. *arXiv preprint arXiv:2211.03267* (2022).

[9] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3? *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures* (2022). https://doi.org/10.18653/v1/2022.deelio-1.10

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

[11] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. 2021. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342* (2021).

[12] Michael Murray and Maya Cakmak. 2022. Following natural language instructions for household tasks with landmark guided search and reinforced pose adjustment. *IEEE Robotics and Automation Letters* 7, 3 (2022), 6870–6877.

[13] Van-Quang Nguyen, Masanori Suganuma, and Takayuki Okatani. 2021. Look wide and interpret twice: Improving performance on interactive instruction-following tasks. *arXiv preprint arXiv:2106.00596* (2021).

[14] Kolby Nottingham, Prithviraj Ammanabrolu, Alane Suhr, Yejin Choi, Hannaneh Hajishirzi, Sameer Singh, and Roy Fox. 2023. Do embodied agents dream of pixelated sheep?: Embodied decision making using language guided world modelling. *arXiv preprint arXiv:2301.12050* (2023).

[15] Kolby Nottingham, Litian Liang, Daeyun Shin, Charless C Fowlkes, Roy Fox, and Sameer Singh. 2021. Modular framework for visuomotor language grounding. *arXiv preprint arXiv:2109.02161* (2021).

[16] R OpenAI. 2023. GPT-4 technical report. *arXiv* (2023), 2303–08774.

[17] Alexander Pashevich, Cordelia Schmid, and Chen Sun. 2021. Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15942–15952.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[19] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10740–10749.

[20] Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. 2020. Factorizing Perception and Policy for Interactive Instruction Following. *arXiv preprint arXiv:2012.03208* (2020).

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[22] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).

[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[24] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* (2022).