NavProg: Compositional Embodied Visual Navigation Without Training

Filippo Ziliotto[1,2] Tommaso Campari[2] Luciano Serafini[2] Lamberto Ballan[1]

[1] University of Padova [2] Fondazione Bruno Kessler (FBK)

Abstract

Large Language Models (LLMs) are revolutionizing AI, demonstrating excellent reasoning capabilities in composing modules to perform complex image-based tasks. In this article, we propose an approach that extends the concept of program composition through LLMs for images, aiming to integrate them into embodied agents. Specifically, by employing a PointGoal Navigation model as a foundational primitive for guiding an agent through the world, we illustrate how a single model can address diverse tasks without additional training. We delegate primitive composition to an LLM, with only a few in-context examples given alongside the prompt. We evaluate our approach on three popular Embodied AI tasks: ObjectGoal Navigation, Instance-Image Goal Navigation, and Embodied Question Answering, demonstrating competitive results without any specific fine-tuning and establishing efficacy in a zero-shot context.

1. Introduction

Large Language Models (LLMs) have gained significant attention in the field of AI, commended for their impressive ability to generalize and produce responses akin to human reasoning [2, 9, 16, 17]. These generalization capabilities have been recently exploited in static scenarios to tackle complex visual tasks given, as an input to the model, natural language instructions, thus providing a general and modular interface for a broad range of compositional problems. Moreover, these frameworks such as, VisProg and ViperGPT [8, 15] are designed not to require any specific training.

This paper takes a significant stride in extending the key idea introduced in these seminal works, for the highly dynamic domain of Embodied AI (EAI) [6], by defining specialized modules tailored for visual navigation tasks. Recently, modular approaches have excelled in handling semantically complex tasks like ObjectGoal Navigation [1, 3, 21], and those demanding long-term memory and strategic planning, such as MultiObjectNavigation [14, 18]. However, while effective for specific tasks, these methods present a challenge: they necessitate significant adjustments for each task, despite common modules.

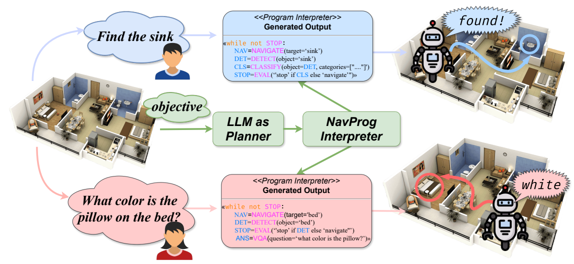

Figure 1. Given a desired user task, NavProg is able to generate a program which is then executed by the agent in the environment. This figure shows an example (top) in which NavProg synthesizes a program for the ObjectNav task, as well as an example (bottom) for Embodied QA.

To tackle this problem, our paper introduces NavProg (see Figure 1), a LLM-based compositional model able to provide key instructions for guiding agent navigation within the environment. By providing a few in-context examples/programs that show how to tackle a specific task, solely using the modules already available in NavProg, the LLM learns to combine these modules into programs to address the task at hand. NavProg integrates modules for semantic navigation, focusing on approaching objects, and image recognition during navigation. These complement the primitives needed to address diverse tasks.

To showcase the framework's flexibility, we conducted zero-shot testing on three prevalent embodied navigation tasks: namely, *i*) ObjectGoal Navigation [1], *ii*) Instance Image Goal Navigation [10], and *iii*) Embodied Question Answering [4].

2. Method and Experiments

Overview. The key component of the proposed framework is referred as to "NavProg Interpreter". It comprises visual recognition modules that can be used by the agent to extract the semantic of the scene, as well as to provide an understanding of the visual context.

To ensure the LLM delivers a reasonable output to the

Method	Trained	SR↑	SPL↑
ZSON [12]	✗	26	13
ModLearn [7]	✗	29	17
PredSem [14]	✗	30	14
L3MVN [22]	✗	50	23
OVRL [19]	✓	33	12
PIRLNav [13]	✓	62	28
OVRL2 [20]	✓	65	28
NavProg (Ours)	✗	**51**	**25**

Method	Trained	DTG↓	SR↑	SPL↑
RL Baseline [11]	✓	6.3	8	4
OVRL2-IIN [11]	✓	5.0	25	12
Mod-IIN [11]	✓	3.1	56	23
NavProg (Ours)	✗	**4.4**	**32**	**15**

Method	Trained	DTG↓	Acc.↑
PACMAN [4]	✓	8.12	40
PACMAN (BC+RF†) [4]	✓	8.13	41
NMC [5]	✓	8.43	39
NMC (BC+A3C) [5]	✓	7.94	44
NavProg (Ours)	✗	**8.7**	**38**

Table 1. **ObjNav results.** Comparison of NavProg with the SoA on the HM3D validation set (left). **InstanceImageNav results.** Comparison of NavProg with trained SoA models, on the HM3D dataset (center). **EQA results.** Comparison of NavProg against the SoA for Embodied Question Answering, evaluated on the EQA-MP3D dataset (right). *denotes abbreviation for *PACMAN*. †denotes abbrevation for *REINFORCE*.

interpreter, it is fed with 10 "in-context examples" across diverse tasks. This enables the LLM to make use of its reasoning capabilities effectively, identifying the most suitable planning for the current user task.

Each generated program is formed by a sequence of primitives (such as DETECT, CLASSIFY, VQA, etc.) that invoke the corresponding NavProg modules, implemented by pre-trained state-of-the-art vision models readily downloadable from the web. This process is made possible by a program interpreter.

All modules are equipped with methods to: *i)* **parse** lines in order to extract input argument names and values, as well as the output variable name; *ii)* **execute** the module, which may involve pre-trained vision language models as well as navigation ones, and update the program state with the output variable name and value. The outputs at each step can be used to understand the system's behavior, enhancing interpretability and enabling a complete failure analysis.

Navigation Module and Exploration Policy. In order to navigate the environment, we define a module employing a PointGoal navigation agent as our foundational module. Equipped solely with a depth image sensor and GPS+compass, the agent navigates toward its destination, given the computed target distance and angle. Once the target is identified, the focus of exploration transitions to reaching the designated goal.

Exploration is carried out using a random navigation policy, sampling distant, unreachable points and enabling the agent to navigate all the possible locations given sufficient time. Additionally, it avoids using a map in the exploration phase, due to the heavy influence of noise on map generation, particularly in depth sensors used in both simulation and real-world scenarios.

Performance analysis and comparison to SoA work. Table 1 (left) shows that our zero-shot approach achieves state of the art results in the OBJNAV setting. Specifically, in comparison to MOPA [14], our enhancements yield a $+21\%$ increase in Success rate and a $+11\%$ improvement in SPL. Moreover, NavProg shows marginally superior performance w.r.t. L3MVN [22] model across all metrics. Next, we compare SoA fully-supervised methods in OBJNAV. Both PIRLNav [13] and OVRL2 [20] outperform NavProg by a considerable margin solely in terms of Success rate, while yielding comparable results in SPL. This disparity in performance can be attributed to their utilization of advanced training strategies. In addition, we conducted an user study on OPEN-SET OBJNAV manually annotating 17 objects from HM3D Minival scenes. Our approach achieved a 42% success rate accross 90 generated episodes, showcasing its effectiveness even in this regard.

In the INSTANCEIMAGENAV task, NavProg outperforms both a Reinforcement Learning Baseline (RL Baseline) model, as well as OVRL2-IIN [11] (Table 1 center). Specifically, in the case of OVRL2-IIN, NavProg shows improvements of 7% in Success and 3% in SPL, highlighting its effectiveness. Morover, OVRL2-IIN is an end-to-end semantic navigation policy model, fine-tuned specifically for INSTANCEIMAGENAV. In contrast, NavProg is surpassed in all metrics by Mod-INN [11], which utilizes a frontier-based exploration and a keypoint-based re-identification method. To the best of our knowledge, NavProg is the only zero-shot approach addressing this task. Table 1 (right) shows that our model yields comparable results against all trained method of EQA task, both in Answer Accuracy and DTG, while requiring no training. All SoA models are trained from a predefined list of possible answers, simplyfing the overall scope to a "classification" problem. In contrast, our model can provide answers using natural language. Furthermore, the primary reasons for failure do not stem from incorrect answers by the VQA module. Instead, they are attributed to incorrect distance calculation from the target and the failure of the object detector to detect the object despite its presence (i.e. failure of DETECT module).

Acknowledgements This project was supported by PNRR MUR project PE0000013-FAIR. We acknowledge the CINECA award under the ISCRA initiative for the availability of high-performance computing resources and support.

References

[1] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 1

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[3] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[4] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[5] Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Neural modular control for embodied question answering. In *Proc. of the International Conference on Robot Learning (CoRL)*, 2018. 2

[6] Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angel X. Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez D'Arpino, Kiana Ehsani, Ali Farhadi, et al. Retrospectives on the embodied ai workshop. *arXiv preprint arXiv:2210.06849*, 2022. 1

[7] Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79), 2023. 2

[8] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[9] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. 1

[10] Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances. *arXiv preprint arXiv:2211.15876*, 2022. 1

[11] Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[12] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. ZSON: Zero-Shot Object-Goal Navigation using Multimodal Goal Embeddings. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[13] Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav, 2023. 2

[14] Sonia Raychaudhuri, Tommaso Campari, Unnat Jain, Manolis Savva, and Angel X. Chang. Mopa: Modular object navigation with pointgoal agents, 2024. 1, 2

[15] Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual Inference via Python Execution for Reasoning. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[16] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

[17] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[18] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. MultiON: Benchmarking semantic map memory using multi-object navigation. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[19] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation, 2022. 2

[20] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav, 2023. 2

[21] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary Tasks and Exploration Enable ObjectNav. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[22] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3MVN: Leveraging Large Language Models for Visual Target Navigation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023. 2