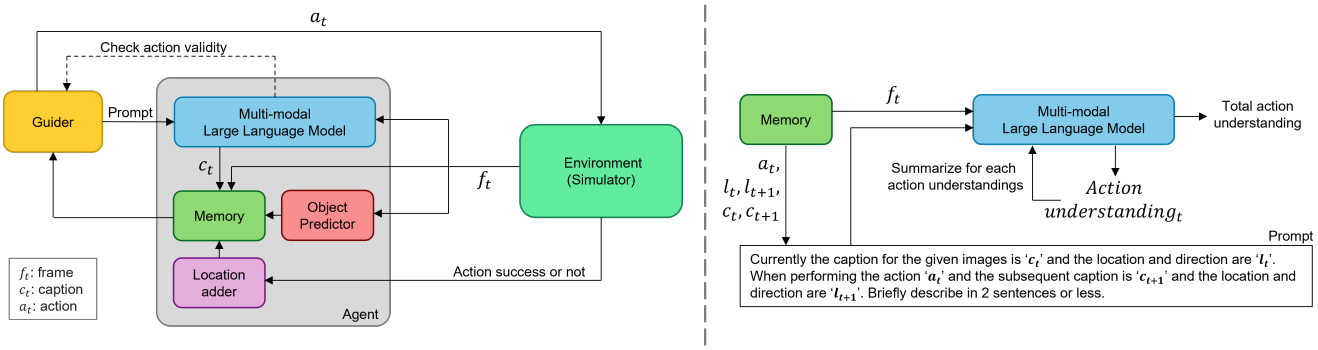
Environmental Understanding Generation with M-LLM for Embodied AI

Jinsik Bang Taehwan Kim
Artificial Intelligence Graduate School, UNIST
{bang, taehwankim}@unist.ac.kr



(a) M-LLM exploration process (b) Action understanding generation process

Figure 1. **Overview of a proposed approach for action understanding generation.** Our approach consists of two stages. In the first stage (a), M-LLM explores the environment while storing experiences and generating action outputs with modules and memory, then the guider checks the validity of M-LLM outputs. In the second stage (b), M-LLM utilizes its structured memory to generate action understandings and summarizes them into one.

1. Introduction

Recent advancements in large language models (LLMs) have demonstrated their capability to be applied to various Embodied AI environments [3, 4, 6, 8], operating more flexibly through planning without the need for training [9, 11, 12]. However, it is difficult for LLMs to immediately understand environments that are new and have not been observed without fine-tuning [2, 13], and even using the supervised data samples as in-context input does not considerably improve the performance [9]. Existing LLM agent studies [12–14] have aimed to enhance performance by augmenting memory [12], leveraging environmental textual information [13], or utilizing predefined action knowledge [14]. However, these approaches have limitations; memory contents are not entirely understandable texts [12], they require textual environmental data [13], and they incur high costs [14]. To address these issues, we propose environmental understanding generation using a multi-modal large language model (M-LLM). It interacts directly with environments, stores experiences in memory, and generates action understanding based on these experiences. These generated understandings aid LLMs in task-based action planning. In summary, our contributions are as follows:

- We propose a novel approach to generating action understanding from M-LLM to augment LLM in embodied AI tasks.
- We introduce a method that M-LLM directly interacts with the environment through information from multiple modules stored in memory. Furthermore, we summarize the constructed memory to build action understanding.
- In experimental results, our approach outperforms baselines by utilizing generative action understanding.

2. Method

In this section, we propose a novel two-stage approach to generate action understanding. The first stage involves M-LLM interacting with the environment and storing information in memory. In the second stage, M-LLM generates action understanding based on the constructed memory. The generated action understanding will be bundled into a single paragraph and input into the LLMs prompt.

2.1. M-LLM Exploration

Overview. To enable M-LLM to directly interact with the environment, it generates one executable action based on predictive information about the environment from various modules. At each time step t, the information is stored in memory, and M-LLM performs exploration and interaction actions randomly for the specified N steps, as shown in Figure 1 (a).

Action space. The action space $\mathcal{A}$ that M-LLM generates is specified depending on the environment, where $\mathcal{A} =$

PickupObject	feature allows the AI agent to interact with the environment by physically lifting and removing an object from its current location, which can change the scene's appearance.
OpenObject	feature allows the AI agent to interact with objects in a scene by opening them, revealing their contents or changing their state, as indicated by the direction of the action.
ToggleObjectOn	feature is a versatile tool that can be used to activate, deactivate, or modify the state, appearance, or functionality of various objects within a scene, depending on the context in which it is applied.
SliceObject	is a feature that allows the AI agent to perform the action of cutting or slicing an object using a tool, such as a knife or a saw, resulting in a change of state or appearance of the object in the scene.

Figure 2. Generated action understandings by M-LLM.

$[a_1, a_2, ..., a_n]$, e.g., MoveAhead and PickupObject in ALFRED [7].

Location Adder. The location adder is utilized to generate actions that enable the exploration of various locations from previous positions during navigation. At each time step t, the 2D coordinates $l_t = (x_t, y_t)$ are updated based on the outcome of the action, allowing exploration of different locations.

Memory. For each scene, the memory module stores previous experiences during exploration to generate action understanding. Additionally, M-LLM generates a caption for the current frame and stores the location. Memory $\mathcal{M}$ represents as follows: $m_t = (f_t, c_t, o_t, a_t, l_t)$, where t is the step, m_t is an element of $\mathcal{M}$, f_t is the frame provided by the environment, c_t is the caption generated by M-LLM for f_t, o_t is the list of predicted objects by the object predictor, a_t is the action performed by M-LLM, and l_t is the location provided by the location adder.

Guider. We propose a guider module that updates prompts based on information from memory $\mathcal{M}$. This module ensures that when the M-LLM generates invalid responses within the action space, it uses regular expressions and additional prompts to request regeneration. Consequently, the M-LLM generates valid actions. This process ensures proper interaction with the environment, enabling the M-LLM to generate valid actions.

2.2. Action Understanding Generation

We introduce a process for generating action understanding that can serve as additional information for planning to perform using LLM or M-LLM, as shown in Figure 1 (b). To generate what M-LLM has understood while exploring the environment, we construct prompts using memory such as frame f_t, action a_t, current and next step's locations l_t, l_{t+1}, and current and next step's captions c_t, c_{t+1} to generate action understanding. At this point, action understanding is only considered for successful action between M-LLM and the environment. Furthermore, M-LLM generates textual forms to describe how the environment changes when a_t is performed. These understandings are summarized into one understanding per action when generating all action understandings. Figure 2 shows the generated action understandings by M-LLM in ALFRED [7].

3. Experiments

We utilize the ALFRED [7] to run our approach, and we evaluate in ALFRED [7] and Watch-And-Help (WAH) [6].

Models	ALFRED [7]		WAH [6]	
	SR (%)		SG-SR (%)	
	Baseline	Ours	Baseline	Ours*
LLaMA2-7B [10]	11.76	11.76	22.50	**25.53**
LLaMA2-70B [10]	20.58	**23.52**	36.08	**37.50**
GPT-4 [1]	38.23	**44.11**	-	-

Table 1. The quantitative result on ALFRED and WAH environment. SR and SG-SR refer to success rate and sub-goal success rate, respectively. * refers to the result of transferring ALFRED understanding.

Figure 3. We evaluate the baseline and our approach on ALFRED. Our approach successfully plans and executes than the baseline.

Experiments setup. We choose the LLaVA-1.6v-13B model [5] as our primary M-LLM. To generate action understanding, we build a memory dataset comprising 20 scenes by randomly selecting from the AI2THOR environment [4]. We evaluate using the LoTA-Bench benchmark [2], which utilizes various LLMs to automatically quantify task planning performance in ALFRED [7] and WAH [6].

Results. Table 1 presents the results of baselines and results of adding action understanding prompts to LLMs. In ALFRED, models with large parameters like LLaMA2-70B and GPT-4 showed increases in success rates, 2.94%, and 5.88%, respectively. Figure 3 shows a successful case in ours. The LLM plans better using the in-context examples provided by the action understanding. The WAH results indicate sub-goal success rates for the transferred understanding of similar actions generated in ALFRED. LLaMA2-7B and LLaMA2-70B showed increases of 3.03% and 1.42% in sub-goal success rate, respectively. This suggests that understanding from different environment can enhance success rates by utilizing shared concepts across environments. Note that GPT-4 could not be evaluated on the WAH [6] due to an error in the provided LoTA benchmark [2].

4. Conclusion and Future Work

Our approach demonstrates increased planning task performance when understanding of the action is added to LLMs. This approach is simple and can be applied to LLMs performing similar tasks. Furthermore, we expect our approach to be utilized without environmental constraints.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Jae-Woo Choi, Youngwoo Yoon, Hyobin Ong, Jaehong Kim, and Minsu Jang. Lota-bench: Benchmarking language-oriented task planners for embodied agents. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 2

[3] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 1

[4] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017. 1, 2

[5] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 2

[6] Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B. Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-{ai} collaboration. In *International Conference on Learning Representations*, 2021. 1, 2

[7] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020. 2

[8] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020. 1

[9] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[10] 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. 2

[11] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv: Arxiv-2305.16291*, 2023. 1

[12] Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models. *arXiv preprint arXiv: 2311.05997*, 2023. 1

[13] Yijun Yang, Tianyi Zhou, Kanxue Li, Dapeng Tao, Lusong Li, Li Shen, Xiaodong He, Jing Jiang, and Yuhui Shi. Embodied multi-modal agent trained by an llm from a parallel textworld. *arXiv preprint arXiv:2311.16714*, 2023. 1

[14] Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Ningyu Zhang, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. Knowagent: Knowledge-augmented planning for llm-based agents. *arXiv preprint arXiv:2403.03101*, 2024. 1