SELU: Self-Learning Embodied Multimodal Large Language Models in Unknown Environments

Anonymous authors
Paper under double-blind review

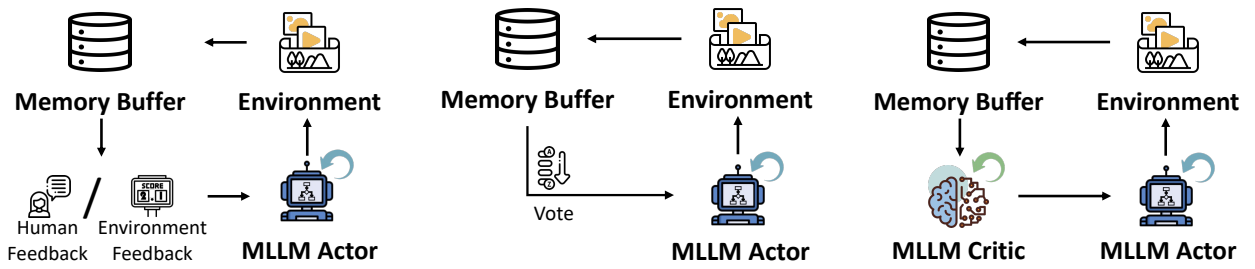
Abstract

Recently, multimodal large language models (MLLMs) have demonstrated strong visual understanding and decision-making capabilities, enabling the exploration of autonomously improving MLLMs in unknown environments. However, external feedback like human or environmental feedback is not always available. To address this challenge, existing methods primarily focus on enhancing the decision-making capabilities of MLLMs through voting and scoring mechanisms, while little effort has been paid to improving the environmental comprehension of MLLMs in unknown environments. To fully unleash the self-learning potential of MLLMs, we propose a novel actor-critic self-learning paradigm, dubbed SELU, inspired by the actor-critic paradigm in reinforcement learning. The critic employs self-asking and hindsight relabeling to extract knowledge from interaction trajectories collected by the actor, thereby augmenting its environmental comprehension. Simultaneously, the actor is improved by the self-feedback provided by the critic, enhancing its decision-making. We evaluate our method in the AI2-THOR and VirtualHome environments, and SELU achieves critic improvements of approximately 28% and 30%, and actor improvements of about 20% and 24% via self-learning.

1 Introduction

Multimodal Large Language Models (MLLMs) have demonstrated impressive perceptual and understanding capabilities across various domains, *e.g.,* , web applications (Ma et al., 2024; Tao et al., 2024; Liu et al., 2024), robotics (Xiong et al., 2024; Li et al., 2024c), gaming (Li et al., 2024d; Qi et al., 2024; Xu et al., 2024), and autonomous driving (Wen et al., 2024; Zhang et al., 2024). Thanks to their powerful capabilities, many works, *e.g.,* , Jarvis-1 (Wang et al., 2023b), STEVE-1 (Lifshitz et al., 2023), and Cradle (Tan et al., 2024b), directly utilize the pre-trained MLLMs to complete various decision-making tasks in different embodied environments.

However, the generalization ability of existing pre-trained MLLMs cannot meet the needs of all environments. For some uncommon environments, embodied MLLMs often exhibit hallucinations and poor visual understanding (Huang et al., 2024; Jiang et al., 2024a). For example, they cannot distinguish left from right or fail to recognize where objects are (Tan et al., 2024b). The reason is that MLLMs have not been further grounded with the environments (Su et al., 2022; Sun et al., 2024b). Grounding can be realized by fine-tuning on the experiences from interacting with the environments. Based on the evaluation methods, experience can be categorized into three types: human feedback (Dai et al., 2024; Kirk et al., 2024), environmental feedback (Tan et al., 2024a; Wang et al., 2024b), and self-feedback (Pang et al., 2024; Madaan et al., 2023). The first two types require additional efforts as illustrated in Figure 1a. Human feedback requires expert demonstrations and annotations, which can be costly and influenced by individual biases (McAleese et al., 2024). Environmental feedback assumes we can obtain a dynamics model of the environment or a well-designed reward model (Fan et al., 2022; Urcelay et al., 2024; Jiang et al., 2024b). Unfortunately, many environments including the real world do not meet such requirements, so these grounding methods are not general. Therefore, we are committed to finding a *general way* to fill in the remaining gaps.



(a) External Feedback (b) Static Self-Feedback (c) Our Self-Feedback

Figure 1: Comparison of our framework with other frameworks in terms of the feedback type.

When operating in an unknown embodied environment without external feedback, the MLLM must rely solely on its inherent capabilities. Some work, see Figure 1b, utilizes the evaluation (discriminative) ability of the pre-trained model itself (Pang et al., 2024) or statistics like voting (Wang et al., 2023a; Huang et al., 2023) to evaluate its own decision-making, and uses such self-feedback to enhance the model's decision-making (generative) capabilities. However, it is evident that this form of self-learning is limited by the model's **static** evaluation capacity, restricting its ability to enhance decision-making. Ideally, we hope that self-learning can work similarly to the actor-critic paradigm (Konda & Tsitsiklis, 1999) in reinforcement learning for embodied tasks, where both the actor and critic iteratively refine their performance. If so, the potential for actor enhancement can be greatly expanded. Unlike reinforcement learning, which relies on external rewards for training, we do not assume the access to any external feedback. Therefore, we aim to **develop a new actor-critic self-learning paradigm for embodied MLLMs in unknown environments.**

In this paper, we introduce a novel **SE**lf-**L**earning paradigm in **U**nknown environments, dubbed **SELU**, as illustrated in Figure 1c. Inspired by the actor-critic paradigm in reinforcement learning, our paradigm learns to simultaneously optimize the MLLM's ability to understand the environment and to make decisions. For the actor module, we fine-tune the model based on the self-feedback from the critic. As the actor gets improved, it can roll-out more successful trajectories to fine-tune the critic. However, without environmental feedback, the critic may provide inaccurate feedback at the beginning of the training phase, which might mislead the overall optimization. Therefore, we adopt self-asking to correct self-feedback and leverage hindsight relabeling to increase sample efficiency by turning the failure trajectory into a successful one for other tasks. These high-quality and diverse trajectories are deemed to enhance the critic's comprehension of the environment. Ultimately, the coupling of these two components mutually promotes the improvement of each other, unleashing the full self-learning potential of MLLMs.

Our key contributions can be summarized as follows:

- We propose SELU, a self-learning paradigm for embodied MLLMs, inspired by the actor-critic paradigm in reinforcement learning, which enables MLLMs to improve themselves in unknown environments without external feedbacks.
- We leverage self-asking and hindsight relabeling to achieve the improvement of the critic, which greatly increase the sample efficiency of our algorithm and make the self-learning possible.
- We demonstrate the effectiveness of SELU in AI2-THOR and VirtualHome, achieving critic improvements of approximately 28% and 30%, and actor improvements of about 20% and 24%, respectively.

2 Related Work

2.1 MLLMs with External Feedback

In recent years, MLLMs has achieved impressive results across various visual benchmarks (Mathew et al., 2021; Tang et al., 2024; Lu et al., 2024b), demonstrating remarkable perception and decision-making capa-

bilities. However, these models still exhibit flaws and often generate unexpected outputs, such as perceptual hallucinations and unreasonable decisions (Yu et al., 2024; Chen et al., 2024). Inspired by reinforcement learning, current approaches use external feedback to correct MLLM's erroneous outputs through a cycle of interaction, feedback, and correction (Pan et al., 2024; Gero et al., 2023). RL methods such as PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) have proven effective (Sun et al., 2024a; Zhai et al., 2024; Lu et al., 2024a). Generally, there are two sources of external feedback: human preference feedback and environmental feedback. Human preference data is used to train a reward model aligned with human preferences (Ouyang et al., 2022). In contrast, environmental feedback typically stems from a rule-based reward function (Tan et al., 2024a; Guo et al., 2025) or a pre-trained large model (Rocamonde et al., 2023; Lee et al., 2024), which often requires substantial support from expert data. For instance, Konyushkova et al. (2025) help Gemini 1.5 to understand Dialog Games through a rule-based approach, generate a positive evaluation function, and construct a high-quality dataset to improve its image understanding capabilities. Compared to existing studies, we focus on the self-learning potential of MLLMs, as external feedback may be not professional enough in unknown environments.

2.2 Self-Improvement in LLMs

Self-improvement in Large Language Models (LLMs) has gained significant attention, as researchers strive to develop models that can learn and adapt from their own outputs, interactions, and internal feedback mechanisms, without relying on external human-labeled data (Yan et al., 2023; Haluptzok et al., 2023). Early explorations in this area are based on unsupervised learning techniques, where models learn representations from vast datasets without explicit human guidance (Winter et al., 2022; Zhao et al., 2019). Expanded to LLMs, self-improvement goes further by enabling models to critique, refine, and adapt their behavior in a more autonomous manner (Tan et al., 2023; Choi et al., 2024). There are two common self-improvement methods: prompt engineering and fine-tuning. The former is an efficient and intuitive approach for large-scale LLMs, as it allows for the establishment of various chains of thought (CoTs) (Wei et al., 2022) to address the same problem (Huang et al., 2023; Feng et al., 2023). For instance, Madaan et al. (2023) demonstrate that LLMs can enhance the rationality of responses by simultaneously inputting a question and reflecting on previous answers. The latter one is proposed as the prompt engineering is unstable for small-scale LLMs, and shows limitation to solve long-horizon tasks (Huang et al., 2023; Pang et al., 2024; Wang et al., 2024a). Wang et al. (2024a) develop a fine-tuning dataset by generating synthetic negative responses to optimize the LLM evaluator, while other work explores how to improve the instruction-following abilities of LLMs (Huang et al., 2023; Pang et al., 2024). Based on these studies, we choose to use fine-tuning to optimize small-scale MLLMs. However, existing methods focus on natural language tasks and overlook the enhancement of MLLM's understanding capabilities in embodied environments. Therefore, we employ an actor-critic framework to facilitate comprehensive self-learning in MLLMs, optimizing both perception and decision-making abilities.

3 Preliminaries

3.1 Actor-Critic in Reinforcement Learning

Actor-critic (Konda & Tsitsiklis, 1999) is a widely adopted framework in reinforcement learning (Li et al., 2024a). The agent consists of two learning modules: an actor and a critic, which are optimized iteratively. The actor selects and executes actions based on current observations. The critic evaluates these observations (and actions) by estimating their values based on reward signals received from the environment, thereby guiding the actor to make improved choices in future. This framework takes advantage of both policy-based learning and value-based learning and is popular nowadays like SAC (Haarnoja et al., 2018).

3.2 Actor-Critic for MLLMs

With the development of MLLMs, the feedback provided to the agent is no longer constrained to scalar values, like rewards; it can now include diverse modalities, such as natural language (Dong et al., 2024). This enables the critic gain more specific and informative feedback on the outputs of the actor. Consequently,

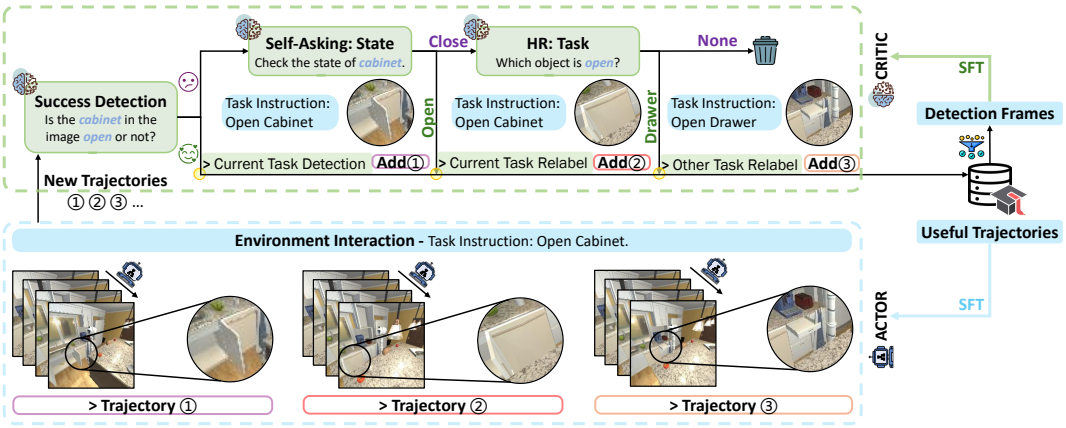


Figure 2: The framework of SELU. (*lower*) The actor MLLM, represented as a robot, collects trajectories for the given instructions. (*upper*) The critic MLLM, denoted as a brain, evaluates these trajectories and determines whether they complete the tasks, guiding the update of the actor MLLM. In addition, the critic MLLM implements self-asking and hindsight relabeling to build a dataset for optimizing itself. The whole framework does not require any external feedback, such as environmental rewards or human annotations.

it can provide more accurate guidance for actor improvement. A prevalent approach in this domain is incorporating human feedback into the critic and building a static evaluation module that can reflect human preference (Ouyang et al., 2022; Kirk et al., 2024). Specifically, human annotated data is used to train a critic model (McAleese et al., 2024) or a reward model (Sontakke et al., 2023; Wang et al., 2024b) to align the MLLM with human preferences better. More rigorous approaches leverage external evaluation mechanisms, such as tool-interactive learning (Gou et al., 2024; Chen et al., 2021), or external knowledge sources like Wikipedia and the Internet (Xu et al., 2023; Li et al., 2024b).

However, regardless of whether preference labels or external tools are used, human intervention remains inevitable. To overcome this reliance, methods like self-consistency (Wang et al., 2023a; Schick et al., 2023) employ a voting mechanism to enable the model to evaluate its own behavior without relying on external information. In addition, LLM-as-a-Judge approaches utilize an LLM as a evaluator to eliminate human intervention (Bai et al., 2022; Lee et al., 2023). However, both self-consistency and LLM-as-a-Judge lack a learnable critic module and therefore cannot improve the model's grounding knowledge of the environment it interacts with. In contrast to previous work, we propose a novel actor-critic based paradigm aimed at achieving self-learning for both the actor MLLM and critic MLLM, enabling them to iteratively improve decision-making and grounding abilities without external human feedback or environmental rewards.

4 Method

The framework of our method is shown in Figure 2. It consists of two components: the actor MLLM and the critic MLLM. The actor MLLM follows instructions and collects trajectories in the environment. The critic MLLM evaluates the collected trajectories and acquires bootstrapped data via self-asking and hindsight relabeling to optimize itself (Section 4.1). Guided by the success detection results from the critic MLLM, the actor MLLM subsequently improves its decision-making performance in the environment (Section 4.2). By combining the two processes, we can achieve coupled improvements of the critic MLLM and the actor MLLM (Section 4.3) without reliance on external feedback.

4.1 Critic: Self-Asking and Hindsight Relabeling

As introduced in Section 1, enhancing the interpretation of environmental grounding information is crucial for an MLLM to improve its performance. In our framework, we propose to achieve this objective via self-asking and hindsight relabeling to acquire bootstrapped data for optimizing the critic MLLM.

Specifically, given an instruction I, the actor MLLM collects a trajectory by following this instruction. The critic takes the last frame o_T of this trajectory as input, using it as the detection frame to determine whether the task depicted by I is completed,

$$l_d = M_c(I, p_d, o_T), \tag{1}$$

where M_c denotes the critic MLLM, $l_d \in \{\text{“}yes\text{”}, \text{“}no\text{”}\}$ is the result of the success detection, and p_d is a prompt for the detection. If the detection result is $l_d =$ “yes”, it means this trajectory is considered to be a successful sample by the critic MLLM for the given instruction and store this trajectory directly into the critic fine-tuning dataset $\mathcal{D}_{\text{critic}}$ in the format (I, p_d, o_T, l_d), as shown by trajectory 1 in Figure 2. This trajectory includes environmental grounding information that aligns with the knowledge contained in the critic MLLM.

If the detection result is $l_d =$ “no”, it means that the critic MLLM considers the trajectory unsuccessful in fulfilling instruction I. We first apply **self-asking** to examine the state of task-related objects, as the decision made by the critic MLLM might not be precise due to potential hallucinations. The critic MLLM is used to obtain the object states,

$$l'_d = M_c(l_s, I), \quad l_s = M_c(j_I, p_s, o_T) \tag{2}$$

where j_I is the object name extracted from instruction I by a text-processing function, p_s is the prompt for object state analysis, l_s is the analysis result, and M_c provides a new success detection l'_d based on l_s and I. The format is corrected to (I, p_d, o_T, l'_d) and stored into the critic fine-tuning dataset $\mathcal{D}_{\text{critic}}$ if $l'_d =$ “yes”. For example, in trajectory 2 of Figure 2, the critic MLLM initially misjudged the completion of the “open cabinet" task. However, when prompted to focus on the state of the cabinet, it successfully self-corrected its judgment.

If the critic MLLM still considers the trajectory as failure, we propose to use **hindsight relabeling** to make use of this trajectory, since it might be helpful for learning the environmental grounding of other instructions. Hindsight relabeling is a method that originated in goal-conditioned reinforcement learning (Andrychowicz et al., 2017). It is based on a simple principle: if a trajectory does not complete the target task, it can be viewed as having accomplished other tasks or subtasks. For example, as shown in trajectory 3 in Figure 2, although it does not complete the task “open cabinet", it successfully completes another task “open drawer". Therefore, we relabel this trajectory with the instruction “open drawer" to help the critic MLLM recognize the completion of the relabeled instruction. We can write this process as,

$$I' = M_c(l_h, a_I), \quad l_h = M_c(a_I, p_h, o_T) \tag{3}$$

where a_I is the verb extracted from the instruction I by another text-processing function. p_h is the prompt for hindsight relabeling, and l_h is the output, which is usually an object name or None. M_c checks whether any objects, other than the target object, in the observation o_T have completed the task associated with a_I. M_c generates a new instruction I' if l_h is not None. After that, the data $(I', p_d, o_T, \text{“}yes\text{”})$ is stored into the critic fine-tuning dataset $\mathcal{D}_{\text{critic}}$. Finally, if a failed trajectory proves meaningless after hindsight relabeling, it is considered as not helpful for the MLLM to understand the environment and discarded.

By applying self-asking and hindsight relabeling, a critic fine-tuning dataset $\mathcal{D}_{\text{critic}}$ is created by MLLM itself containing the last frames considered as successful and the last frames relabeled as successful after self-asking and hindsight relabeling.

4.2 Actor: Critic-Guided Improvement

Recent work has shown that the discriminative ability of an LLM exceeds its generative ability (Pang et al., 2024). As MLLMs are typically trained the same way as LLMs, we believe that MLLMs' evaluation abilities would also surpass their generation abilities. For instance, we can easily prompt MLLMs to extract understanding from a given image, but it is challenging to prompt them to choose an appropriate action based on perceived task-relevant information like distance or direction. Our experiment result in Section 5.2 also supports this conclusion, where the critic module always performs better than the actor. Therefore, we propose using the critic MLLM to guide the improvement of the actor MLLM in the environment without external feedback.

Specifically, the actor MLLM interacts with the environment and collects online trajectories. At each timestep t, the actor generates an action plan $l_{a,t}$ by,

$$l_{a,t} = M_a(I, p_a, o_t), \tag{4}$$

where M_a represents the actor MLLM, I is the task instruction, p_a is the prompt for action plan and o_t is the current image observation. After collecting a whole trajectory, the critic MLLM determines whether this trajectory completes the instruction I, as described in Section 4.1. If the answer is yes, the whole trajectory is put into the actor fine-tuning dataset $\mathcal{D}_{\text{actor}}$ with a format of $\{(I, p_a, o_t, l_{a,t})\}_{t=0}^{T}$. The relabeled successful trajectories after hindsight relabeling are also added into the actor fine-tuning dataset $\mathcal{D}_{\text{actor}}$. Since this dataset only contains task completion trajectories, the actor can quickly converge towards completing tasks in the current environment by fine-tuning on the dataset. Note that the actor fine-tuning dataset $\mathcal{D}_{\text{actor}}$ consists of trajectory data, while the critic fine-tuning dataset $\mathcal{D}_{\text{critic}}$ only contains the last frames of these trajectories.

4.3 Actor-Critic Coupling Improvement

We employ Supervised Fine-Tuning (SFT) (Devlin et al., 2019; Brown et al., 2020) and Low-Rank Adaptation (LoRA) (Hu et al., 2022) to update both the actor and critic MLLMs. Initially, the actor MLLM interacts with the environment to collect online trajectories containing grounding information, as depicted in the lower part of Figure 2. The critic module then evaluates and classifies these trajectories based on the last frame, as shown in the upper part of Figure 2. We select successful trajectories identified by the critic MLLM to create the actor fine-tuning dataset $\mathcal{D}_{\text{actor}}$. Subsequently, we utilize the last frame to construct the critic fine-tuning dataset $\mathcal{D}_{\text{critic}}$. The update of the actor and critic can be performed iteratively, and make them both improve step-by-step. A detailed pseudo-code for our algorithm is available in Appendix A.1.

5 Experiments

5.1 Experimental Setup

Environments. In order to simulate embodied MLLM interactions in unknown environments, we select AI2-THOR (Kolve et al., 2022) and VirtualHome (Puig et al., 2018) for our experiments. Both environments offer open-ended tasks, various interactive objects, and selectable camera perspectives, facilitating data collection for the actor and critic. More details can be seen in Appendix A.2.1.

- **AI2-THOR** is an interactive simulation environment designed for embodied AI research. The primary tasks require agents to navigate and interact with household objects. It offers highly realistic 3D environments that simulate kitchens, living rooms, and other indoor settings.
- **VirtualHome** is also an embodied simulation platform designed to imitate human activities and tasks in home environments. This environment focuses on task completion through multi-step action sequences, making it ideal for testing long-term planning.

Task Selection. The goal of SELU is to enable the MLLM to acquire atomic skills in unknown environments through self-feedback. Therefore, the instruction list is derived from the robot's executable actions. In AI2-THOR, locobot is selected as the agent, as this work does not consider low-level control of robotic arms. There are three fundamental actions: pick up, open, and break, which serve as the basis for long-horizon task composition. To ensure task diversity and feasibility, we first prompt the MLLM to explore the environment and use the discovered objects to initialize the instruction list which serves as the task set. We randomly sample 2-3 objects for each type of task. Considering the requirements of atomic skills and training costs, we restrict the maximum step for all tasks to 10. We apply a similar approach in the VirtualHome environment, selecting "female1" as the agent and primarily testing in grab, open, and sit tasks.

MLLMs. To demonstrate the generalization capability of our framework, we conduct experiments using two MLLMs: LLaVA (Liu et al., 2023) and Qwen-VL (Bai et al., 2023).

- **LLaVA** has gained prominence as one of the most popular MLLMs due to its simple architecture and lower training data requirements. These features enable LLaVA to generate responses more swiftly, and suitable for inference to investigate self-learning.
- **Qwen-VL** is the first model to use a 448x448 resolution image input. Due to its higher resolution, this model exhibits enhanced visual understanding capabilities. We opt for Qwen-VL with the aim of better success detection, thereby facilitating more efficient self-learning of MLLMs.

Baselines. We compare SELU with five methods to investigate the feasibility of self-learning of MLLMs in embodied environments:

- **DG** refers to the results obtained through direct generation from the initial MLLM without any fine-tuning.
- **SC** (Wang et al., 2023a) represents an optimization method of MLLMs through self-consistency. Specifically, we employ multiple chains of thought (CoT) to prompt an MLLM to answer the same question, followed by majority voting. In our experiments, we utilize three different CoTs to guide the MLLM, ultimately voting for the most reasonable action.
- **LMSI** (Huang et al., 2023) is a self-improvement method based on SC. It generates "high-confidence" answers for unlabeled questions to build fine-tuning datasets. This approach enables the LLM to iteratively improve its performance based on the voting mechanism, and we extend this approach to MLLMs.
- **Self-Refine** (Madaan et al., 2023) involves multiple rounds of self-reflection, followed by self-optimization based on the reflection results. This method focuses on prompt optimization and has been validated for feasibility in large-scale LLMs, such as GPT-4. In our experiments, we reflect 3 rounds to get the final result.
- **LLM-Planner** (Song et al., 2023) is a method specifically designed for multimodal embodied planning with text retrieval-augmented techniques. In experiments, we retrieve 3 trajectories as memory for the actor module.
- **RAP** (Kagaya et al., 2024) is a method employing a multimodal retrieval technique for embodied tasks. In our experiments, 3 retrieved trajectories are used for decision-making.
- **Sel-Ask** represents the method that uses grounding guidance prompt engineering techniques from SELU, to substantiate the necessity of critic optimization.
- **SELU-One** represents the method of using the same MLLM to simultaneously perform actor and critic tasks, and fine-tuning with a combination of actor and critic datasets. This approach aims to investigate the feasibility of utilizing a single MLLM to meet the requirements of our framework.

Evaluation. For the critic module, we use the MLLM's detection accuracy of whether the task is completed to reflect its ability to understand the environment. Specifically, we provide the last frame of the trajectory during testing and compare whether the MLLM's judgment is consistent with the environmental feedback. For the actor module, the MLLM's decision-making ability is measured by its task success rate. In the experiment, each task is tested 100 times to get the actor MLLM's task success rate. The final frames of these trajectories are then used to evaluate the critic MLLM's success detection accuracy. All results are evaluated at the same dataset. More details for implementation are provided in Appendix A.2.

5.2 AI2-THOR

LLaVA. We first demonstrate the effectiveness of SELU in the AI2-THOR environment. After online interactions with the environment and fine-tuning of LLaVA, the critic exhibits an average performance

improvement of approximately 27%, while the actor achieves an improvement of around 20% compared to the original model. Table 1 and Table 2 present the accuracy of task success detection and task success rate respectively.

Table 1: Accuracy of task success detection in the AI2-THOR environment.

Method	Pick up	Open	Break	Avg.
DG	80.67%	36.50%	50.50%	55.89%
Self-Ask	79.67%	41.50%	53.50%	58.22%
SELU-One	68.67%	30.50%	25.50%	41.56%
SELU	**94.33%**	**67.50%**	**87.50%**	**83.11%**

Table 2: Task success rate in the AI2-THOR environment. SC and Self-Refine use prompt engineering to realize self-learning, whereas LMSI and SELU utilize fine-tuning. LLM Planner and RAP serve as typical embodied MLLM baselines, incorporating memory retrieval mechanisms.

Method	Pick Up	Open	Break	Avg.
DG	68.33%	65.00%	15.50%	49.61%
SC	65.67%	68.50%	17.50%	50.56%
Self-Refine	69.67%	70.50%	14.50%	51.56%
LMSI	75.67%	52.50%	19.50%	49.22%
LLM-Planner	66.67%	52.00%	11.50%	43.39%
RAP	62.33%	63.50%	14.50%	46.78%
SELU-One	91.33%	**85.50%**	27.50%	68.11%
SELU	**94.67%**	83.50%	**30.50%**	**69.56%**

In Table 1, it is evidenced that the unified fine-tuning of the actor and the critic (SELU-One) leads to a decline in success detection, even worse than the original critic. Self-Ask proves that fine-tuning is more effective than prompt engineering for the critic module. Although SELU-One achieves a task success rate comparable to that of SELU as shown in Table 2, the compromised critic results in SELU-One incorrectly analyzing the trajectory in subsequent epochs.

Table 2 demonstrates that baselines are not suitable for the self-learning of embodied MLLMs. Both prompt-engineering and fine-tuning baselines struggle to improve the decision-making ability of the MLLM. The reason is that the embodied MLLM cannot give a correct task detection with a lack of environmental understanding. As we can see in Table 1, the initial judgment of the MLLM (DG) on tasks is only about 55%. In this case, merely optimizing the prompt to create multiple CoTs for repeated reflection does not help the MLLM gain task achievement details in an unknown environment. Therefore, neither SC nor Self-Refine can substantially enhance the task success rate. LLM-Planner and RAP also don't increase MLLM's ability to understand the environment. For fine-tuning baselines, relying on statistical voting to validate its own behavior even leads to worse performance. For instance, in the Open task, for LMSI the task detection accuracy of 36.5% causes the actor's performance to drop from 65% to 52.5% after fine-tuning. These results demonstrate the necessity of the critic module in SELU, and optimizing the critic is crucial for enhancing the actor's performance.

Notably, in the Open task, we can observe a low accuracy of success detection for SELU; however, it still helps the actor improve performance. This highlights the role of hindsight relabeling, which is discussed in detail in Section 5.4.

Qwen-VL. In order to prove that SELU can help different MLLMs achieve self-learning, we select Qwen-VL and test it in the AI2-THOR environment under the same setting. The result is shown in Table 3, which indicates that SELU can help Qwen-VL improve the task evaluation capability by about 24% and the decision-making performance by approximately 23%.

Table 3: Self-learning performance of SELU on Qwen-VL in the AI2-THOR environment.

Task	Qwen-VL-DG		Qwen-VL-SELU	
	Critic	Actor	Critic	Actor
Pick Up	73.33%	57.67%	**95.67%**	**95.33%**
Open	51.00%	46.50%	**81.50%**	**68.00%**
Break	63.50%	12.50%	**83.50%**	**21.50%**
Avg.	62.61%	38.89%	**86.89%**	**61.61%**

5.3 VirtualHome

We then conduct experiments in the VirtualHome environment, which incorporates a greater variety of items and human agents, thereby enriching the experimental environments to demonstrate the effectiveness of our method. The experimental results are presented in Tables 4 and 5. In this environment, SELU enhances LLaVA task evaluation capability by approximately 30% and improves decision-making performance by around 24%, and it also outperforms baselines. As the environment becomes more complex, the lack of environmental understanding causes SC and Self-Refine to negatively impact the decision-making of the original embodied MLLM. As shown in Table 5, the performance of SC and Self-Refine become even lower than DG.

Table 4: Accuracy of task success detection in the VirtualHome environment.

Method	Grab	Open	Sit	Avg.
DG	52.67%	35.33%	44.50%	44.17%
Self-Ask	54.33%	36.67%	43.50%	44.83%
SELU-One	45.33%	15.67%	**48.50**%	36.50%
SELU	**93.67%**	**83.33%**	47.50%	**74.83%**

Table 5: Task success rate in the VirtualHome environment.

Method	Grab	Open	Sit	Avg.
DG	65.00%	83.33%	56.50%	68.28%
SC	52.67%	81.67%	61.50%	65.28%
Self-Refine	59.67%	74.33%	60.50%	64.83%
LMSI	35.67%	93.67%	52.50%	60.61%
LLM-Planner	55.33%	78.67%	50.50%	61.50%
RAP	58.67%	79.33%	51.50%	63.17%
SELU-One	83.67%	**98.67%**	**94.50%**	92.28%
SELU	**93.33%**	97.67%	93.50%	**94.83%**

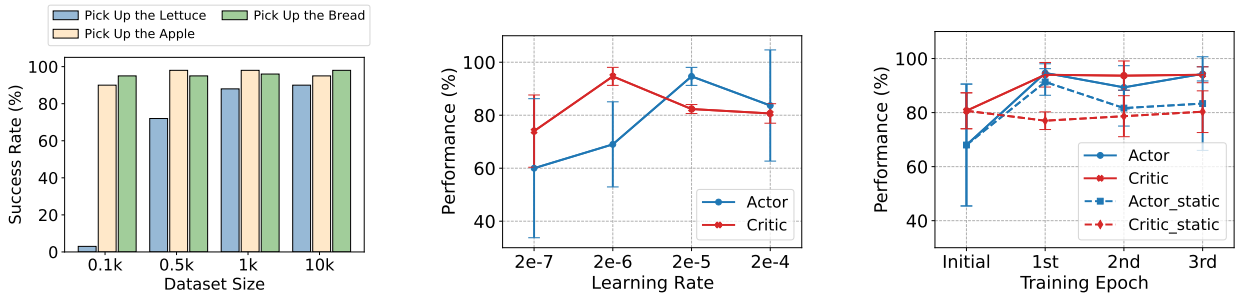
5.4 Ablation Study

We conduct ablation experiments on the critic module in the AI2-THOR environemnt, and the results are shown in Table 6. SELU w/o HR means that we do not perform hindsight relabeling to reanalyze the trajectory when evaluating the task. SELU w/o ALL means that we remove both self-asking and hindsight relabeling and use the data obtained from environment interaction to directly fine-tune the actor. In this case, the evaluation result of the critic is derived from the original MLLM.

By comparing SELU and SELU w/o ALL, we can see the importance of whole critic module clearly. Only by understanding the environment can we achieve the improvement of decision-making in all tasks. By comparing SELU w/o HR and SELU w/o ALL, we find that self-asking can correct the critic's comprehension of the environmental task, but reflection on a single task is not enough. We find that the lack of hinsight relabeling leads to a decrease in the diversity of the online fine-tuning dataset, and some tasks can not get

Table 6: Ablation study in the AI2-THOR environment.

Task	Critic (Success Detection Accuracy)			Actor (Task Success Rate)		
	SELU	w/o HR	w/o ALL	SELU	w/o HR	w/o ALL
Pick Up	**94.33%**	83.67%	80.67%	**94.67%**	67.33%	56.33%
Open	**67.50%**	31.50%	36.67%	**83.50%**	66.50%	72.50%
Break	**87.50%**	83.50%	50.50%	**30.50%**	27.50%	17.50%
Avg.	**83.11%**	66.22%	55.95%	**69.56%**	57.11%	48.78%



(a) Dataset Size (b) Learning Rate (c) Training Iterations

Figure 3: Hyperparameter study of SELU on picking up tasks in the AI2-THOR environment: (a) explores the size of the interaction dataset required for embodied MLLMs, (b) illustrates why a single MLLM is not suitable for SELU from the perspective of learning rare, and (c) demonstrates that the effect of multiple training iterations.

enough data to perform well. For example, we can observe a declined performance of success detection and decision-making in Open tasks from SELU w/o ALL to SELU w/o HR. By incorporating hindsight relabeling, from SELU w/o HR to SELU, we can perform a comprehensive multi-task evaluation for each trajectory, ensuring that the embodied MLLM achieves self-learning on each task. Consequently, self-asking and hindsight relabeling are essential components of the critic.

5.5 Hyperparameter Analysis

Online Dataset Size. Since the MLLMs fine-tuning process is sensitive to the dataset size, we explore the amount of interaction data required to achieve effective learning for embodied tasks. We conduct multiple tests on this variable based on picking up tasks in the AI2-THOR environment. The results are presented in Figure 3a.

The size of dataset is described by the number of trajectories collected for a single task. For instance, dataset-10k indicates that 10k trajectories are collected for a specific task, such as picking up an apple, during online interactions. We evaluate the performance through the actor in terms of task success rate. We can see that the performance of dataset-1k is close to that of dataset-10k. For dataset-10k, it takes about two days to collect one epoch data. To balance experimental performance with sampling efficiency, we opt to sample 1k trajectories per task in our experiments.

Based on this conclusion, we employ a data augmentation method during the actor training process. Since MLLMs tend to prioritize text over images when making decisions, they often focus excessively on the text prompt and overlook image comprehension. To address this issue, we shuffle the action lists in the prompts of training data to provide multiple prompts for the same image. This approach not only increases the dataset size, but also strengthens the connection between the MLLM policy and the observations.

Learning Rate. The learning rate is a critical factor, which prevents us from using the same MLLM for the actor and critic. We test different learning rates for the actor and critic separately on picking up tasks in the AI2-THOR, and the results are shown in Figure 3b. We find that, due to the varying sizes of the fine-tuning datasets, using a uniform learning rate inevitably leads to overfitting or underfitting in one of the components, thereby impacting the overall performance of SELU. Therefore, we ultimately use two MLLMs

to construct the SELU framework, ensuring effective self-learning. In our experiments, the learning rate for the critic is set to 2e-6 and for the actor is set to 2e-5.

Training Iterations. The goal of our framework is to achieve multi-iteration self-learning improvement; therefore, we also evaluate the results of multiple rounds of fine-tuning on picking up tasks in the AI2-THOR environment. To ensure the effectiveness of fine-tuning, we retain 30% of the last fine-tuning dataset each time and obtain the remaining 70% of the data through online interaction. The results are presented in Figure 3c. The solid line represents the performance under an optimized critic, while the dashed line indicates the results under a static critic.

The unstable improvement of the actor under a static critic highlights the necessity of fine-tuning the critic. What's more, our results indicate that multiple iterations of fine-tuning do not consistently improve SELU's performance in every iteration. Both the actor and critic exhibit significant performance improvements during the first iteration of fine-tuning, but show fluctuations and minimal growth in the subsequent iterations. We attribute this limitation to the performance of the critic. The critic performs the success detection based on the last frame, making it difficult to compare the quality of different successful trajectories. Once the actor reaches a level to roughly complete the task, we lack the nuanced supervisory signals to guide the actor for further improvement. Consequently, while there is a notable improvement after the first iteration, subsequent enhancements are limited. We also attempt to increase the number of frames for success detection. However, the performance of the small-scale MLLM does not meet our needs.

6 Conclusion and Limitation

In this paper, we introduce SELU, a method for MLLMs to achieve self-learning in unknown environments. SELU facilitates interaction with the environment, analyzes interaction trajectories, builds an online dataset, and performs coupled optimization of the actor and critic. We employ self-asking and hindsight relabeling to enhance the critic task evaluation capabilities. Ablation experiments demonstrate that relabeling significantly expands the critic task judgment range. By leveraging the principle that MLLMs possess stronger perceptual abilities than decision-making abilities, we improve the performance of the actor policy. We test SELU in the AI2-THOR and VirtualHome environments, achieving critic improvements of approximately 28% and 30%, and policy improvements of about 20% and 24%, respectively. Additionally, to validate the applicability of SELU across different MLLMs, we evaluate it on Qwen-VL, resulting in a 23% performance enhancement.

One limitation of SELU is the lack of detailed assessment of successful trajectories. SELU can help embodied MLLM to self-learn how to accomplish tasks in unknown environments. How to complete tasks more efficiently is the next problem to solve. In future work, we will explore finer-grained critic signals to perform more accurate quality assessments of trajectories, guiding embodied MLLMs to tackle more complex tasks, such as long-horizon combination tasks.

References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

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 *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Xiang Chen, Chenxi Wang, Ningyu Zhang, Yida Xue, xiaoyan yang, YUE SHEN, Jinjie GU, and Huajun Chen. Unified hallucination detection for multimodal large language models. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711. Association for Computational Linguistics, 2021.

Wonje Choi, Woo Kyung Kim, Minjong Yoo, and Honguk Woo. Embodied cot distillation from LLM to off-the-shelf agents. In *Forty-first International Conference on Machine Learning*, 2024.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Yihong Dong, Kangcheng Luo, Xue Jiang, Zhi Jin, and Ge Li. PACE: Improving prompt with actor-critic editing for large language model. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 7304–7323, 2024.

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. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Advances in Neural Information Processing Systems*, volume 36, pp. 70757–70798. Curran Associates, Inc., 2023.

Zelalem Gero, Chandan Singh, Hao Cheng, Tristan Naumann, Michel Galley, Jianfeng Gao, and Hoifung Poon. Self-verification improves few-shot clinical information extraction. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*, 2023.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Patrick Haluptzok, Matthew Bowers, and Adam Tauman Kalai. Language models can teach themselves to program better. In *The Eleventh International Conference on Learning Representations*, 2023.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1051–1068. Association for Computational Linguistics, 2023.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13418–13427, 2024.

Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. Hallucination augmented contrastive learning for multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27036–27046, 2024a.

Haobin Jiang, Junpeng Yue, Hao Luo, Ziluo Ding, and Zongqing Lu. Reinforcement learning friendly vision-language model for minecraft. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024b.

Tomoyuki Kagaya, Thong Jing Yuan, Yuxuan Lou, Jayashree Karlekar, Sugiri Pranata, Akira Kinose, Koki Oguri, Felix Wick, and Yang You. Rap: Retrieval-augmented planning with contextual memory for multimodal llm agents. In *NeurIPS 2024 Workshop on Open-World Agents*, 2024.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *The Twelfth International Conference on Learning Representations*, 2024.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai, 2022.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Ksenia Konyushkova, Christos Kaplanis, Serkan Cabi, and Misha Denil. Vision-language model dialog games for self-improvement. *arXiv preprint arXiv:2502.02740*, 2025.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 11286–11315, 2024.

Boyu Li, Haoran Li, Yuanheng Zhu, and Dongbin Zhao. Mat: Morphological adaptive transformer for universal morphology policy learning. *IEEE Transactions on Cognitive and Developmental Systems*, 16 (4):1611–1621, 2024a.

Miaoran Li, Baolin Peng, Michel Galley, Jianfeng Gao, and Zhu Zhang. Self-checker: Plug-and-play modules for fact-checking with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 163–181. Association for Computational Linguistics, 2024b.

Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18061–18070, 2024c.

Zaijing Li, Yuquan Xie, Rui Shao, Gongwei Chen, Dongmei Jiang, and Liqiang Nie. Optimus-1: Hybrid multimodal memory empowered agents excel in long-horizon tasks. *arXiv preprint arXiv:2408.03615*, 2024d.

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila A. McIlraith. STEVE-1: A generative model for text-to-behavior in minecraft. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023.

Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. Visualwebbench: How far have multimodal LLMs evolved in web page understanding and grounding? In *First Conference on Language Modeling*, 2024.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024a.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*, 2024b.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. CoCo-agent: A comprehensive cognitive MLLM agent for smartphone GUI automation. pp. 9097–9110. Association for Computational Linguistics, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506, 2024.

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. Language model self-improvement by reinforcement learning contemplation. In *The Twelfth International Conference on Learning Representations*, 2024.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Siyuan Qi, Shuo Chen, Yexin Li, Xiangyu Kong, Junqi Wang, Bangcheng Yang, Pring Wong, Yifan Zhong, Xiaoyuan Zhang, Zhaowei Zhang, Nian Liu, Yaodong Yang, and Song-Chun Zhu. Civrealm: A learning and reasoning odyssey in civilization for decision-making agents. In *The Twelfth International Conference on Learning Representations*, 2024.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

Timo Schick, Jane A. Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. PEER: A collaborative language model. In *The Eleventh International Conference on Learning Representations*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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*, pp. 2998–3009, 2023.

Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bı yık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. In *Advances in Neural Information Processing Systems*, volume 36, pp. 55681–55693. Curran Associates, Inc., 2023.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. Read before generate! faithful long form question answering with machine reading. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 744–756. Association for Computational Linguistics, 2022.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. In *Annual Meeting of the Association for Computational Linguistics*, 2024a.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented RLHF. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 13088–13110. Association for Computational Linguistics, 2024b.

Weihao Tan, Wentao Zhang, Shanqi Liu, Longtao Zheng, Xinrun Wang, and Bo An. True knowledge comes from practice: Aligning large language models with embodied environments via reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024a.

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, et al. Cradle: Empowering foundation agents towards general computer control. In *NeurIPS 2024 Workshop on Open-World Agents*, 2024b.

Xiaoyu Tan, Shaojie Shi, Xihe Qiu, Chao Qu, Zhenting Qi, Yinghui Xu, and Yuan Qi. Self-criticism: Aligning large language models with their understanding of helpfulness, honesty, and harmlessness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 650–662, Singapore, 2023. Association for Computational Linguistics.

Jingqun Tang, Qi Liu, Yongjie Ye, Jinghui Lu, Shu Wei, Chunhui Lin, Wanqing Li, Mohamad Fitri Faiz Bin Mahmood, Hao Feng, Zhen Zhao, et al. Mtvqa: Benchmarking multilingual text-centric visual question answering. *arXiv preprint arXiv:2405.11985*, 2024.

Heyi Tao, Sethuraman T V, Michal Shlapentokh-Rothman, Tanmay Gupta, Heng Ji, and Derek Hoiem. WebWISE: Unlocking web interface control for LLMs via sequential exploration. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3693–3711. Association for Computational Linguistics, 2024.

Belen Martin Urcelay, Andreas Krause, and Giorgia Ramponi. Reinforcement learning from human text feedback: Learning a reward model from human text input. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*, 2024.

Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*, 2024a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023a.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. RL-VLM-f: Reinforcement learning from vision language foundation model feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 51484–51501. PMLR, 21–27 Jul 2024b.

Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. JARVIS-1: Open-world multi-task agents with memory-augmented multimodal language models. In *Second Agent Learning in Open-Endedness Workshop*, 2023b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022.

Licheng Wen, Xuemeng Yang, Daocheng Fu, Xiaofeng Wang, Pinlong Cai, Xin Li, Tao MA, Yingxuan Li, Linran XU, Dengke Shang, Zheng Zhu, Shaoyan Sun, Yeqi BAI, Xinyu Cai, Min Dou, Shuanglu Hu, Botian Shi, and Yu Qiao. On the road with GPT-4v(ision): Explorations of utilizing visual-language model as autonomous driving agent. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024.

Robin Winter, Marco Bertolini, Tuan Le, Frank Noe, and Djork-Arné Clevert. Unsupervised learning of group invariant and equivariant representations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Chuyan Xiong, Chengyu Shen, Xiaoqi Li, Kaichen Zhou, Jiaming Liu, Ruiping Wang, and Hao Dong. Autonomous interactive correction MLLM for robust robotic manipulation. In *8th Annual Conference on Robot Learning*, 2024.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5967–5994. Association for Computational Linguistics, 2023.

Xinrun Xu, Yuxin Wang, Chaoyi Xu, Ziluo Ding, Jiechuan Jiang, Zhiming Ding, and Börje F. Karlsson. A survey on game playing agents and large models: Methods, applications, and challenges. *arXiv preprint arXiv:2403.10249*, 2024.

Yiming Yan, Tao Wang, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Mingxuan Wang. BLEURT has universal translations: An analysis of automatic metrics by minimum risk training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5428–5443. Association for Computational Linguistics, 2023.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and Tat-Seng Chua. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13807–13816, 2024.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15459–15469, 2024.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International conference on machine learning*, pp. 7523–7532. PMLR, 2019.

A Appendix/Supplementary Material

A.1 Pseudocode of SELU

algorithm 1 SELU

Input: critic MLLM M_c, actor MLLM M_a, critic fine-tuning dataset $\mathcal{D}_{\text{critic}}$, actor fine-tuning dataset $\mathcal{D}_{\text{actor}}$, maximum timestep T, initial instruction list L, success detection prompt p_d and action plan prompt p_a

Output: critic MLLM M_c, actor MLLM M_a

```
D_critic, D_actor ← {}
function "SELU"
    for instruct I in L do
        while data collecting not done do
            for timestep t = 1 to T do
                get observation o_t from env
                l_{a,t} = M_a(I, p_a, o_t)
                use l_{a,t} to interact with env
            end for
            l_d = M_c(I, p_d, o_T)
            if l_d = "yes" then
                store (I, p_d, o_T, l_d) into D_critic
                store (I, p_a, o_t, l_{a,t}), t = 1, ...T into D_actor
            else
                get l'_d through self-asking
                if l'_d = "yes" then
                    store (I, p_d, o_T, l'_d) into D_critic
                    store (I, p_a, o_t, l_{a,t}), t = 1, ...T into D_actor
                else
                    get I' through hindsight relabeling
                    if I' ≠ "None" then
                        store (I', p_d, o_T, yes) into D_critic
                        store (I', p_a, o_t, l_{a,t}), t = 1, ...T into D_actor
                    end if
                end if
            end if
        end while
    end for
    optimization M_c and M_a by D_critic and D_actor
    return critic MLLM M_c, actor MLLM M_a
end function
```

A.2 Implementation Details

A.2.1 Environments

Figure 4 shows our experiment environments. Both environments restrict agents to only interact with visible items, limiting their operational range to guarantee behavior plans realistic. Therefore, the actor MLLM makes decisions based on first-person perspective input to ensure accuracy as Figure 4a and Figure 4c show. Given the limitations of the first-person view, the critic MLLM uses a third-person perspective to evaluate the trajectory, reducing hallucinations and obtaining accurate scene information as Figure 4b and Figure 4d show.

The positioning of the third-person camera is crucial, as it should accurately capture the agent's position and the objects it interacts with. Any occlusion or interference can impair the MLLM's understanding of the image, thereby affecting the results of critic success detection and hindsight relabeling.

(a) AI2-THOR 1 (b) AI2-THOR 2

(c) VirtualHome 1 (d) VirtualHome 2

Figure 4: The diagram of experimental environments. We utilize the first-person perspective for decision-making and a third-person perspective for trajectory evaluation.

A.2.2 Hyperparameters

The specific MLLMs we use are LLaVA-V1.6-Mistral-7B and Qwen-VL. We use LoRA to fine-tune them, the hyperparameters are as follows.

Table 7: Hyperparameters of LLaVA fine-tuning by LoRA

Hyperparameters	**Value**
Train_batch_size	16
Eval_batch_size	4
Gradient_accumulation_steps	1
Learning_rate_actor	2e-5
Learning_rate_critic	2e-6
Warmup_ratio	0.03
Weight_decay	0.0
Model_max_length	2048
Lr_scheduler_type	cosine
Tf32	True

Table 8: Hyperparameters of Qwen-VL fine-tuning by LoRA

Hyperparameters	**Value**
Train_batch_size	2
Eval_batch_size	1
Gradient_accumulation_steps	8
Learning_rate_actor	1e-5
Learning_rate_critic	1e-6
Warmup_ratio	0.01
Weight_decay	0.1
Adam_beta2	0.95
Model_max_length	2048
Lr_scheduler_type	cosine
Bf16	True
Lazy_preprocess	True

Both models are configured with a temperature of 0 and a maximum token length of 2048 for response generation. The maximum number of environment steps is set to 10. The multiple chains of thought for self-consistency (Wang et al., 2023a) , and the multiple rounds of reflection in self-refine (Madaan et al., 2023) are set to 3, following the settings reported in their papers. For LLM-Planner (Song et al., 2023) and RAP (Kagaya et al., 2024), retrieval parameters follow the official implementation to ensure consistency. The number of retrieved trajectories is fixed at the minimum value of 3, as adopted in RAP. The list of hyperparameters is provided in Table 9.

Table 9: Hyperparameters List for Baselines.

Hyperparameters	**Value**
Response generation temperature	0
Maximum token length	2048
Maximum environment steps	10
Number of CoTs (Wang et al., 2023a)	3
Number of reflection rounds (Madaan et al., 2023)	3
Retrieve buffer	20
KNN retrieves (LLM Planner (Song et al., 2023))	9
Multimodal retrieves (RAP (Kagaya et al., 2024))	5
Retrieve trajectories input	3

A.3 Visualization of Actor and Critic on LLaVA in AI2-THOR

The embodied actor MLLM is used to interact with the unknown environment, and collect trajectories from the evaluation of the critic MLLM. An example for 'pick up the lettuce' is shown in figure 5.

The embodied critic MLLM is used to perform success detection on each trajectory and use self-asking and hindsight relabeling techniques to build bootstrapped dataset. An example for 'break the mug' is shown in figure 6.

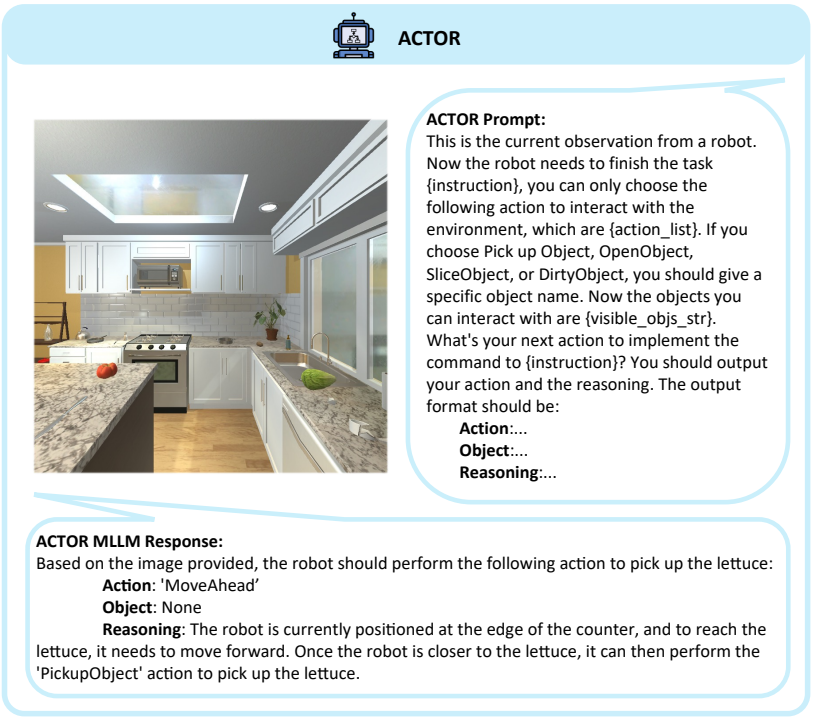


Figure 5: A visualization of the actor MLLM interacting with the AI2-THOR environment. The agent is instructed to pick up the lettuce. As the lettuce is far away, the agent needs to move closer before attempting to pick it up.

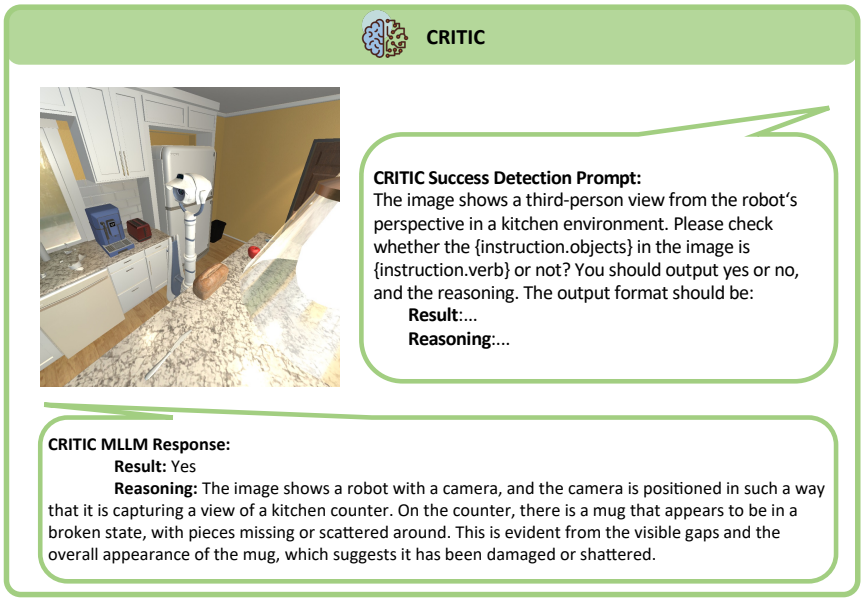


Figure 6: A visualization of the critic MLLM conducting success detection on a trajectory. Since the trajectory has completed the 'break the mug' task, it can be added to the fine-tuning dataset.

A.4 Prompts for SELU

The specific prompts we use for SELU are:

Actor-Interaction with Env

This is the current observation from a {agent} in a {AI2-THOR/VirtualHome} environment. Now the {agent} needs to finish the task {instruction}, you can only choose the following action to interact with the environment, which are {action_list}. If you choose {PickupObject/GrabObject}, OpenObject, {BreakObject/SitObject}, you should give a specific object name. Now the objects you can interact with are {visible_objs_str}. What's your next action to implement the command to {instruction}? You should output your action and the reasoning. The output format should be:
Action:...
Object:...
Reasoning:...

Critic-Success Detection

The image shows a third-person view from the {agent}'s perspective in a {AI2-THOR/VirtualHome} environment. Please check whether the {instruction.objects} in the image is {instruction.verb → adj.} or not? You should output yes or no, and the reasoning. The output format should be:
Result:...
Reasoning:...

Critic-Self Asking 1

The image shows a third-person view from the {agent}'s perspective in a {AI2-THOR/VirtualHome} environment. Please check the state of the {instruction.objects} in the image. You should output the state and the reasoning. The output format should be:
State:...
Reasoning:...

Critic-Self Asking 2

The image shows a third-person view from the {agent}'s perspective in a {AI2-THOR/VirtualHome} environment. The {instruction.objects} in the observation is in {objects.state} state, please determine whether the {instruction} has been completed or not. You should output yes or no, and the reasoning. The output format should be:
Result:...
Reasoning:...

Critic-hindsight relabeling 1

The image shows a third-person view from the {agent}'s perspective in a {AI2-THOR/VirtualHome} environment. Please see the image carefully. Determine whether there is any object that is {instruction.verb → adj.} by the {agent}? You should output the object name and the reasoning. The output format should be:
Object:...
Reasoning:...

Critic-hindsight relabeling 2

The image shows a third-person view from the {agent}'s perspective in a {AI2-THOR/VirtualHome} environment. The {relabeling.object} in the observation is {instruction.verb → adj.}, you should give a new instruction based on it. The original instruction is {instruction}, what's the new instruction?
The output format should be:
New instruction:...
Reasoning:...