Seeing Isn't Believing: Mitigating Belief Inertia via Active Intervention in Embodied Agents

Hanlin Wang[1], **Chak Tou Leong**[1], **Jian Wang**[1,2†], **Wenjie Li**[1]
[1] Department of Computing, The Hong Kong Polytechnic University
[2] College of Computer Science, Sichuan University
{hanlin-henry.wang, chak-tou.leong}@connect.polyu.hk
jian51.wang@polyu.edu.hk cswjli@comp.polyu.edu.hk

Abstract

Recent advancements in large language models (LLMs) have enabled agents to tackle complex embodied tasks through environmental interaction. However, these agents still make suboptimal decisions and perform ineffective actions, as they often overlook critical environmental feedback that differs from their internal beliefs. Through a formal probing analysis, we characterize this as *belief inertia*, a phenomenon where agents stubbornly adhere to prior beliefs despite explicit observations. To address this, we advocate active belief intervention, moving from passive understanding to active management. We introduce the Estimate-Verify-Update (EVU) mechanism, which empowers agents to predict expected outcomes, verify them against observations through explicit reasoning, and actively update prior beliefs based on the verification evidence. EVU is designed as a unified intervention mechanism that generates textual belief states explicitly, and can be integrated into both prompting-based and training-based agent reasoning methods. Extensive experiments across three embodied benchmarks demonstrate that EVU consistently yields substantial gains in task success rates. Further analyses validate that our approach effectively mitigates belief inertia, advancing the development of more robust embodied agents. Our code is available at https://github.com/WangHanLinHenry/EVU.

1 Introduction

Large language models (LLMs) have revolutionized embodied AI, enabling agents to solve increasingly complex, long-horizon tasks (Huang et al., 2022; Wang et al., 2023; Li et al., 2024). Effective task-solving requires not only sophisticated reasoning, but also continuous interaction with the embodied environment. To this end, prior work has explored a variety of techniques, including inference-time iteration (Yao et al., 2022; Shinn et al., 2023), imitation learning (Chen et al., 2023), and reinforcement learning (Wu et al., 2025). The key to these methods is a tight feedback loop, in which the agent perceives the environment, interprets observations to internal states, reasons, and executes actions accordingly. Since an agent's entire decision-making process critically depends on environmental feedback, integrating the observed information into its reasoning is crucial for task success (Wang et al., 2024; Fung et al., 2025).

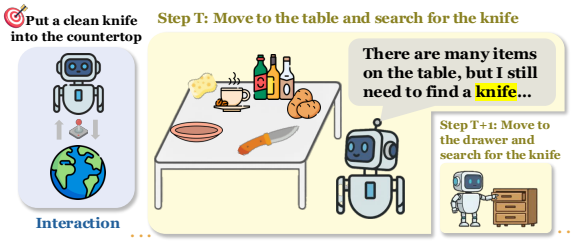

Figure 1: Illustrative example of *observational neglect* in embodied agents. While the agent observes a knife on the target table, its subsequent internal belief ("I still need to find a knife") fails to integrate the observed information, leading to an unnecessary search action.

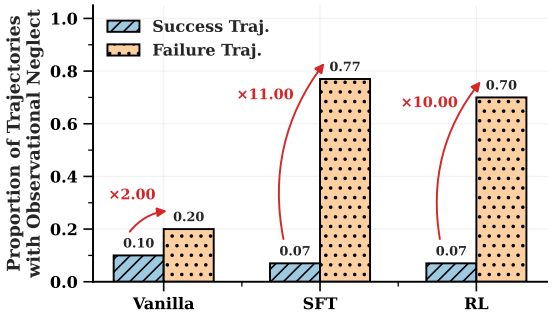

Figure 2: Statistical results of observational neglect on the ALFWorld benchmark.

However, we observe a significant gap between *receiving* observations and effectively *utilizing* them. As illustrated in Figure 1, an agent may observe a knife on a countertop, yet its subse-

[†]Corresponding author. This work was mainly conducted at PolyU, while the author is now at Sichuan University.

Findings of the Association for Computational Linguistics: ACL 2026, pages 37796–37815
July 2-7, 2026 ©2026 Association for Computational Linguistics

quent reasoning behaves as if the knife were still missing, initiating a redundant search action. We refer to this behavior as "observational neglect", where the agent observes but fails to integrate observed information into its internal reasoning process. Our statistical analysis (see Figure 2) on the ALFWorld (Shridhar et al., 2020) benchmark reveals that such neglect is not a marginal error but a predominant failure mode in unsuccessful trajectories. Moreover, this behavior is widespread across various learning paradigms, from vanilla prompting to RL-tuned models, indicating a critical bottleneck in how embodied agents transform observations into their own beliefs, i.e., the internal understandings of the environment states.

To uncover the root cause of observational neglect, we conduct probing experiments to analyze the agent's belief dynamics (see Section 3). Our analysis identifies a critical cognitive bias which we term **belief inertia**, a phenomenon that agents tend to stubbornly adhere to their prior expectations of action outcomes, even when faced with contradictory evidence. This inertia results in a belief-observation misalignment, where the agent's internal belief remains unchanged despite observing a changing environment. While recent studies have explored belief modeling (Zhang et al., 2024; Lidayan et al., 2025), they largely rely on implicit belief dynamics, where beliefs are updated passively and latently. Without an effective intervention when necessary, such strategies leave the agent's reasoning prone to being "blinded" by biased priors that overshadow its observations.

To address this, we advocate **active belief intervention**, shifting the paradigm from passive update to active cognitive management (see Section 4). We introduce the **E**stimate-**V**erify-**U**pdate (**EVU**), a simple yet effective mechanism for belief intervention. Unlike previous works, EVU decouples belief management from action generation by producing explicit belief states in a textual form. With EVU, the agent first estimates an expected outcome, verifies it against actual observations through LLM-based reasoning, and finally updates its prior belief to a grounded posterior. Crucially, we integrate EVU seamlessly into both prompting- and training-based agent learning methods. Extensive experiments across diverse embodied benchmarks demonstrate that EVU consistently yields substantial gains. Further in-depth analysis confirms that our EVU mitigates belief inertia effectively.

In summary, our contributions are as follows:

- We identify and formalize belief inertia, a critical phenomenon in embodied agents where internal beliefs overshadow actual observations, leading to widespread observational neglect.
- We propose active belief intervention, implemented via the Estimate-Verify-Update (EVU) mechanism. It provides a unified way to actively manage belief states and can be seamlessly integrated with various agent learning methods.
- We demonstrate the superiority and generalizability of EVU through extensive experiments across multiple embodied benchmarks. Further analysis validates that EVU significantly mitigates belief inertia, providing valuable insights into developing robust embodied agents.

2 Preliminary

Problem Formulation. The reasoning process of embodied agents is often formulated as a Partially Observable Markov Decision Process (POMDP), denoted as $(\mathcal{U}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{Z}, \mathcal{R})$, where $\mathcal{U}$ is the instruction space, $\mathcal{S}$ the hidden state space, $\mathcal{A}$ the action space, $\mathcal{O}$ the observation space, $\mathcal{T}$ the transition function, $\mathcal{Z}$ the observation function, and $\mathcal{R}$ the reward function. To isolate the cognitive aspects of agent–environment interaction from low-level perception, we focus on text-only settings where $\mathcal{U}$, $\mathcal{A}$, and $\mathcal{O}$ are all expressed in natural language. Accordingly, we model an embodied agent as an LLM policy π_θ that generates textual actions.

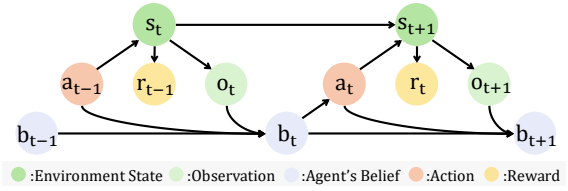

Figure 3: POMDP formulation in embodied agents.

Agent Beliefs. In a POMDP, the true environment state s_t is never observed directly, so the agent must maintain an internal belief state b_t that summarizes its estimate of s_t and serves as the basis for decision making. As shown in Figure 3, at each step the agent integrates new observation o_t with its prior belief b_{t-1} to obtain an updated belief b_t, and then reasons the next action conditioned on b_t, which in turn shapes future observations and rewards. In many LLM-based agents, such belief dynamics are handled implicitly. A common practice, i.e., ReAct-style (Yao et al., 2022) agent, appends all past actions and observations into an

interaction history $h_t = (u, o_1, a_1, o_2, \ldots, o_t)$ and relies on LLMs to latently infer b_t when reasoning the next action, without explicitly representing the belief state.

3 Belief Inertia in Embodied Agents

In this section, we investigate why ReAct-style embodied agents exhibit *observational neglect*. Since this phenomenon manifests as a misalignment between the agent's implicit beliefs and the external environment, we employ a probing-based method to explicitly elicit and track these beliefs.

3.1 Probing Agent Beliefs

To probe an agent's beliefs, we append probing questions to the interaction history and utilize the agent's responses to decode its internal understanding of the environment.

Specifically, given the interaction history h_t at time t, we define a set of task-relevant environment variables $\mathcal{V}$, such as whether an object has been acquired or whether it is currently inside a container. For each variable $v \in \mathcal{V}$, we construct a corresponding yes–no probe question q_v (e.g., "Is the key currently in the box?"). We then construct a probing prompt by concatenating h_t and q_v and feeding it into the agent policy. Finally, we read the first-token logits for the candidate answers "yes" and "no", denoted by $\ell_{\text{yes}}(h_t, q_v)$ and $\ell_{\text{no}}(h_t, q_v)$. We then define the raw belief value, the question-induced bias, and the debiased belief value as

$$\begin{aligned} s(h_t, q_v) &= \ell_{\text{yes}}(h_t, q_v) - \ell_{\text{no}}(h_t, q_v), \\ b(q_v) &= \ell_{\text{yes}}(h_\emptyset, q_v) - \ell_{\text{no}}(h_\emptyset, q_v), \\ \beta(h_t, q_v) &= s(h_t, q_v) - b(q_v). \end{aligned} \tag{1}$$

A positive $\beta(h_t, q_v)$ indicates an inclination to answer "yes" for v under h_t after filtering out question-induced bias, while a negative value indicates an inclination to answer "no".

For each variable v, let $y_v \in \{+1, -1\}$ denote the ground-truth answer in the current environment state, where $y_v = +1$ corresponds to "yes" and $y_v = -1$ to "no". We define the **True Belief Value**

$$A(h_t, v) \;=\; y_v \, \beta(h_t, q_v), \tag{2}$$

whose sign indicates whether the probed belief agrees with the true state ($A(h_t, v) > 0$) or not ($A(h_t, v) < 0$), and whose magnitude $|A(h_t, v)|$ serves as a proxy for the confidence of this belief. Correlation analyses between probing results and the agent's behavior support the reliability of our probing method (see Appendix A).

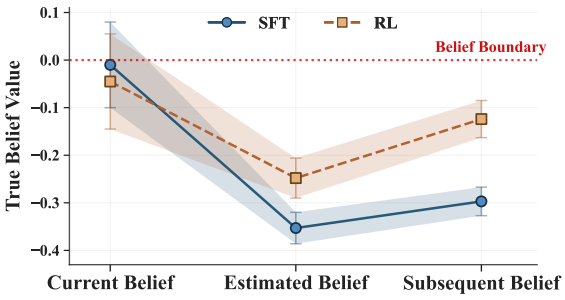

Figure 4: Probing results of belief dynamics across three stages. The belief boundary separates positive (correct) beliefs from negative (incorrect) ones.

3.2 Belief Inertia Phenomenon

To analyze the agent's decision-making process when the *observational neglect* occurs, we apply the above probing method to examine how its beliefs are evolved.

We first collect 100 observational-neglect cases from both SFT-trained and RL-trained agents in ALFWorld (Shridhar et al., 2020), where the agent's internal reasoning for the next action a_t neglects the newly received observation o_t. We then extract o_t from v that captures the critical feedback and construct the corresponding probe question q_v. To examine belief updating for this variable, we analyze how the agent's beliefs evolve between the previous action a_{t-1} that produces the new observation o_t and the subsequent action a_t. Specifically, we probe the agent's belief about v at three stages: (1) **Current belief**, representing the belief before taking a_{t-1}, probed under $(u, o_1, a_1, o_2, \ldots, \boldsymbol{o_{t-1}})$; (2) **Estimated belief**, representing the belief after taking a_{t-1} but before observing o_t, probed under $(u, o_1, a_1, o_2, \ldots, \boldsymbol{o_{t-1}}, \boldsymbol{a_{t-1}})$; and (3) **Subsequent belief**, representing the belief after receiving o_t, probed under $(u, o_1, a_1, o_2, \ldots, \boldsymbol{o_{t-1}}, \boldsymbol{a_{t-1}}, \boldsymbol{o_t})$.

Figure 4 visualizes the probed True Belief Value across the three stages for both SFT- and RL-trained agents. At the initial stage, $A(c_t, v)$ lies near the belief boundary, indicating that the agent does not hold a strong prior about the query. After taking an action, the agent forms a strong but incorrect belief about v. Although the subsequent belief increases slightly after receiving o_t, it remains negative for both agent types. This persistence suggests that the new environmental feedback fails to update the agent's internal state. Consequently, the agent reasons the next action based on a stale belief formed immediately after the previous action,

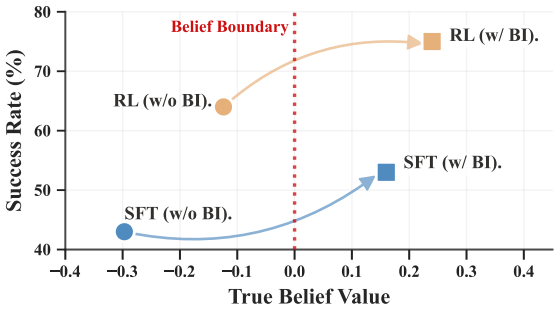

Figure 5: Impact of oracle belief intervention (BI).

which remains inconsistent with the environment state. We term this failure to update **belief inertia**, a phenomenon that manifests as observational neglect during subsequent reasoning.

3.3 Belief Intervention

To examine whether the belief inertial causes observational neglect, we conduct a **Belief Intervention (BI)** experiment: we manually correct the agent's belief to match the true environment state and then observe whether it can reason and act correctly.

Specifically, we adopt the same observational-neglect cases as mentioned above and evaluate both the SFT-trained and RL-trained agents under two distinct settings. In the typical setting (**w/o BI**), the agent generates a_t conditioned on the standard interaction history h_t. In the intervention setting (**w/ BI**), we explicitly append a description of the oracle environment state s_t^* to the interaction history, yielding $(u, o_1, a_1, \ldots, o_t, s_t^*)$. In both settings, we apply our probing method to assess the agent's belief about the environment variables relevant to o_t and report the task success rate on these observational-neglect cases.

As illustrated in Figure 5, belief intervention shifts the agents' internal states across the Belief Boundary, correlating directly with improved task performance. Without intervention, both SFT-trained and RL-trained agents linger in the negative True Belief Value region, indicating a persistence of incorrect beliefs that corresponds to lower success rates. Upon intervention, the True Belief Value becomes positive, signifying the adoption of the correct environmental state. Crucially, this belief correction translates into a marked increase in success rates for both agents. These results confirm that the primary bottleneck in observational neglect cases is the failure to update beliefs. By manually aligning the belief state with the actual environment, we mitigate the downstream consequences of this failure, demonstrating that the agents possess the necessary reasoning capabilities to succeed once the belief barrier is removed.

4 Method: Active Belief Intervention

Drawing upon the crucial findings in Section 3, we advocate active belief intervention and introduce a simple yet effective **E**stimate-**V**erify-**U**pdate (**EVU**) mechanism that estimates, verifies, and updates beliefs actively through a unified perspective. Figure 6 shows the overview of our approach.

4.1 Estimate-Verify-Update Mechanism

In contrast to typical ReAct-style agents, which passively encode the entire interaction history as an implicit belief about the environment within the latent model parameters, our EVU mechanism maintains an explicit belief state B_t, a natural language summary that sufficiently represents the understanding of the environment. Crucially, EVU recursively takes the previous belief state as input and evolves it through a structured loop of estimation, verification, and update by the agent itself.

Estimation. Initially, the agent attempts to predict the immediate consequence of its previous action before processing the actual new observation. In this step, the agent establishes a baseline expectation by estimating action outcomes E_t as:

$$E_t \sim \pi_\theta(\cdot \mid B_{t-1}, a_{t-1}, o_t), \tag{3}$$

where E_t describes what the agent expects to observe, explicitly modeling its expectation.

Verification. The agent then processes the actual observation o_t from the environment. Instead of updating the belief directly, the agent first generates a verification evidence V_t to compare its estimation against the actual observation:

$$V_t \sim \pi_\theta(\cdot \mid B_{t-1}, a_{t-1}, o_t, E_t). \tag{4}$$

Here, V_t serves as a structured "surprise signal" that explicitly captures whether the observation confirms or contradicts the expectation, preventing the agent from hallucinating success or overlooking contradictory evidence.

Belief Update. Finally, the agent synthesizes the reasoning chain to transition from the previous belief state B_{t-1} to the current belief state B_t. This

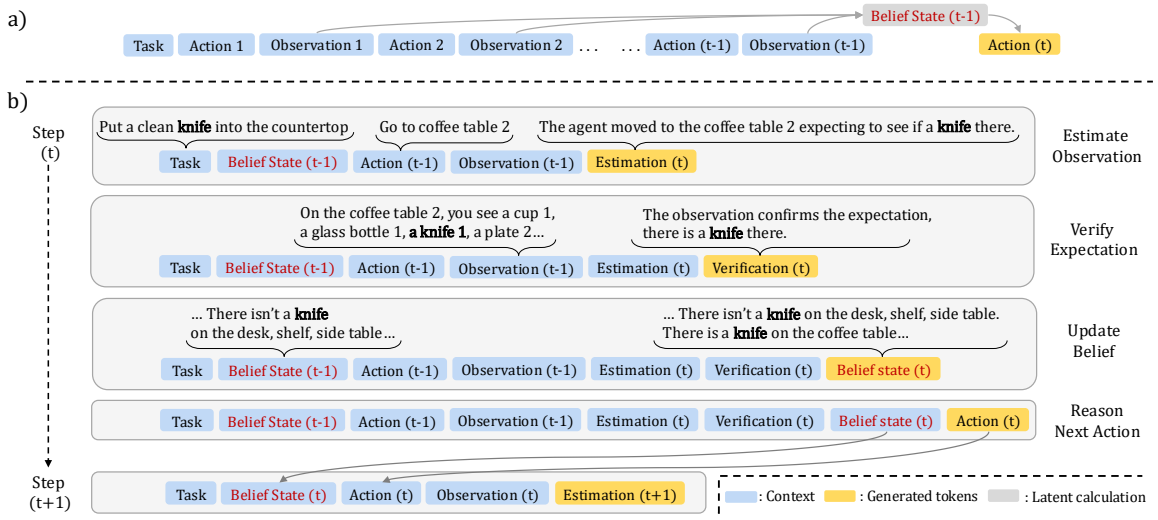

Figure 6: Overview of our proposed active belief intervention method. Compared to typical belief modeling methods (*top*), we introduce a unified Estimate-Verify-Update (EVU) mechanism (*bottom*).

process takes the prior belief, the initial estimation E_t, and the verification evidence V_t (i.e., the surprise signal) as inputs:

$$B_t \sim \pi_\theta(\cdot \mid B_{t-1}, a_{t-1}, o_t, E_t, V_t). \quad (5)$$

By leveraging this surprise-aware verification V_t, the model ensures that the new belief state B_t (e.g., "There is a knife on the coffee table…") accurately reflects the latest environmental changes while retaining valid historical information about the environment. This updated belief then serves as the foundation for subsequent reasoning.

4.2 A Unified Intervention Perspective

Our EVU is a general mechanism that decouples *state maintenance* from *action generation*. This separation allows for a unified formulation that is agnostic to both the prompting-based methods and the training-based algorithms.

4.2.1 Prompting-based Belief Intervention

Standard prompting-based methods typically employ a prompting strategy $\mathcal{S}$ to directly map interaction history h_t to an action a_t. In our approach, we realize belief intervention by augmenting the original strategy $\mathcal{S}$ with specific instructions designed to enable active belief dynamics. We denote this belief-enhanced strategy as $\mathcal{S}^*$, which explicitly guides the agent to perform active belief intervention before decision-making. This process is formulated as:

$$(E_t, V_t, B_t, a_t) \sim \pi_\theta\left(\cdot \mid \mathcal{S}^*(B_{t-1}, a_{t-1}, o_t)\right). \quad (6)$$

By requiring the agent to explicitly model its belief dynamics prior to the action a_t, this intervention ensures that the agent's decision-making is grounded in a structured and updated understanding of the environment, rather than relying solely on implicit patterns within the raw history.

4.2.2 Training-based Belief Intervention

In training-based approaches, we update the model parameters θ to internalize the belief update mechanism. Unlike typical training methods that focus solely on optimizing action generation, we unify belief update and action generation into an autoregressive process and optimize them jointly.

Formally, at each time step t, the model takes the previous belief state B_{t-1} and the recent interaction history (a_{t-1}, o_{t-1}) as inputs to generate the current reasoning chain and action:

$$(E_t, V_t, B_t, a_t) \sim \pi_\theta(\cdot \mid B_{t-1}, a_{t-1}, o_t). \quad (7)$$

To optimize this process, we define a general objective function $\mathcal{J}(\theta)$, which represents the expected utility of the generated trajectory. Depending on the training paradigm, $\mathcal{J}(\theta)$ can be flexibly instantiated as the negative log-likelihood in Supervised Fine-Tuning (SFT) or the expected reward in Reinforcement Learning (e.g., PPO, GRPO). This is formulated as:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \mathcal{J}(\theta) = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{\tau \sim \pi_\theta}[U(\tau)], \quad (8)$$

where $\tau = (B_0, a_0, o_0, E_1, V_1, B_1, a_1, \dots)$ represents the augmented trajectory containing both cognitive states and external actions. By maximizing

	Method	ALFWorld				VirtualHome				ScienceWorld			
		Seen		Unseen		Seen		Unseen		Seen		Unseen	
	DeepSeek V3.2												
Prompting	NoThinking	50.7	–	42.3	–	8.0	–	7.2	–	47.0	–	46.0	–
	↪ w/ EVU (Ours)	55.0	(↑**4.3**)	47.6	(↑**4.3**)	12.8	(↑**4.8**)	12.8	(↑**5.6**)	55.0	(↑**8.0**)	52.2	(↑**6.2**)
	Plan-and-Act	52.1	–	44.8	–	12.8	–	12.8	–	55.0	–	50.9	–
	↪ w/ EVU (Ours)	53.6	(↑1.5)	46.3	(↑1.5)	15.2	(↑2.4)	**14.4**	(↑1.6)	58.3	(↑3.3)	55.3	(↑4.4)
	ReAct	55.7	–	47.6	–	13.6	–	12.8	–	60.3	–	57.8	–
	↪ w/ EVU (Ours)	**56.4**	(↑0.7)	**49.8**	(↑2.2)	**16.0**	(↑2.4)	13.6	(↑0.8)	**62.3**	(↑2.0)	**60.9**	(↑3.1)
Training	Qwen3-1.7B-Instruct												
	SFT	37.1	–	20.1	–	7.2	–	8.0	–	7.3	–	11.2	–
	↪ w/ EVU (Ours)	41.4	(↑4.3)	33.6	(↑**13.5**)	16.0	(↑**8.8**)	25.6	(↑**17.6**)	23.2	(↑15.9)	24.8	(↑**13.6**)
	PPO	42.1	–	32.0	–	10.4	–	22.4	–	37.0	–	41.0	–
	↪ w/ EVU (Ours)	47.1	(↑5.0)	40.3	(↑8.3)	17.6	(↑7.2)	28.8	(↑6.4)	62.9	(↑**25.9**)	**54.0**	(↑13.0)
	GRPO	47.0	–	44.0	–	15.7	–	19.4	–	41.7	–	42.9	–
	↪ w/ EVU (Ours)	**52.1**	(↑**5.1**)	**49.3**	(↑5.3)	**20.0**	(↑4.3)	**36.9**	(↑17.5)	**47.7**	(↑6.0)	50.3	(↑7.4)
	Qwen2.5-3B-Instruct												
	SFT	65.7	–	50.7	–	20.0	–	20.0	–	19.9	–	13.7	–
	↪ w/ EVU (Ours)	70.0	(↑**4.3**)	56.7	(↑6.0)	27.2	(↑**7.2**)	34.4	(↑**14.4**)	49.0	(↑**29.1**)	45.3	(↑**31.6**)
	PPO	77.8	–	54.4	–	24.0	–	23.2	–	53.6	–	51.6	–
	↪ w/ EVU (Ours)	79.3	(↑1.5)	58.2	(↑3.8)	28.8	(↑4.8)	35.2	(↑12.0)	60.3	(↑6.7)	62.7	(↑11.1)
	GRPO	83.6	–	70.8	–	25.6	–	24.8	–	49.7	–	52.8	–
	↪ w/ EVU (Ours)	**85.7**	(↑2.1)	**79.1**	(↑**8.3**)	**31.2**	(↑5.6)	**36.0**	(↑11.2)	**70.9**	(↑10.6)	**70.8**	(↑18.0)

Table 1: Main results of success rates (%) on three representative embodied benchmarks. "Seen" and "Unseen" denote held-out test sets with tasks seen and unseen during training, respectively. "w/ EVU" denotes plugging our EVU mechanism into base methods. **Bold** values represent the best performance within each backbone model group.

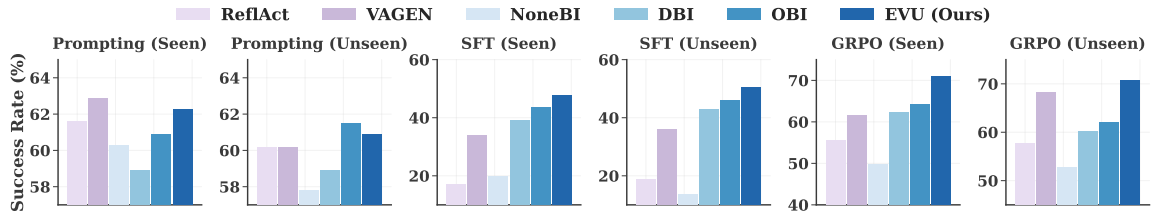

Figure 7: Success rates (%) of different methods with belief intervention variants.

$\mathcal{J}(\theta)$, the optimization algorithm adjusts the probability mass not just for the final action a_t, but for the entire reasoning chain (E_t, V_t, B_t). This ensures that the model learns to maintain high-quality beliefs that causally lead to optimal actions, allowing gradients (or reward signals) to propagate through the belief update process.

5 Experiments

5.1 Experimental Setup

Benchmarks. We evaluate our method on three representative embodied agent benchmarks: ALFWorld (Shridhar et al., 2020), VirtualHome (Puig et al., 2018), and ScienceWorld (Wang et al., 2022). Following prior studies (Song et al., 2024; Wang et al., 2025a), we adopt Success Rate (SR) as our primary evaluation metric and evaluate agents on both seen and unseen scenarios. Appendix B provides more details of these datasets.

Baseline Methods. We evaluate our method by integrating it into two categories of baselines and measuring the resulting performance gains: (1) prompting-based methods, including NoThinking (Ma et al., 2025), Plan-and-Act (Kim et al., 2025), and ReAct (Yao et al., 2022); and (2) training-based methods, including SFT (Chen et al., 2023), PPO (Schulman et al., 2017), and GRPO (Shao et al., 2024). Additional details about these baselines are provided in Appendix C.

Implementation Details. We conduct experiments on DeepSeek V3.2 (Liu et al., 2025) for prompting-based evaluations, as well as Qwen2.5-3B-Instruct (Yang et al., 2025a) and Qwen3-1.7B-Instruct (Yang et al., 2025a) for training-based evaluations. For the SFT phase, the training epochs are set to 3. For the RL phase, the training process consists of 250 steps, and we select the checkpoint with the best performance on the validation set for

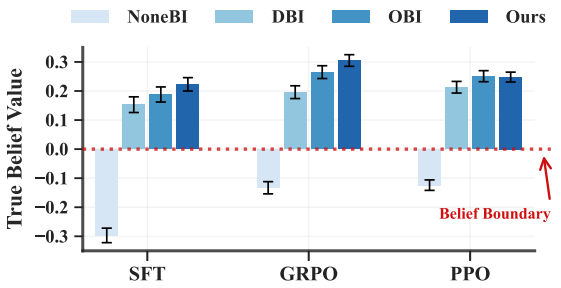

Figure 8: Quantitative probing results of different belief intervention methods in mitigating belief inertia.

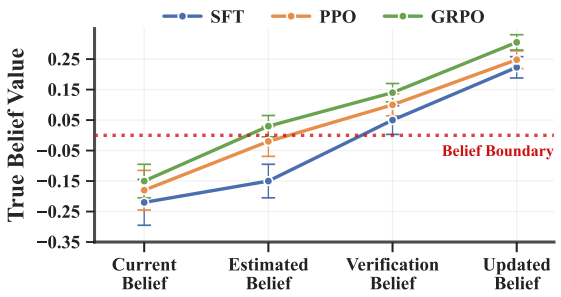

Figure 9: Quantitative probing results of different phase in mitigating belief inertia.

final testing. During inference, the decoding temperature of the LLMs is set to 0.0 for deterministic generation. Detailed hyperparameters and prompt designs are provided in the Appendix E.

5.2 Main Results

Table 1 presents a comprehensive evaluation of different methods across three benchmarks. We summarize key findings of EVU below:

Consistent improvement across benchmarks and backbone models. As shown in Table 1, EVU consistently outperforms all baseline methods across all three benchmarks, demonstrating its effectiveness and robustness. Notably, we observe that the average performance gain on Unseen splits (+9.21) is higher than that on Seen splits (+6.8). This indicates that our method effectively grounds the agent even when facing novel observations in OOD scenarios, thereby substantially enhancing generalization capabilities. Additional comparisons with advanced context-management and search-based agentic baselines are provided in Appendix D, where EVU remains consistently beneficial.

Robustness in both prompting- and training-based settings. Our method demonstrates remarkable flexibility by seamlessly integrating with both prompting-based and training-based methods. As shown in Table 1, EVU consistently enhances performance across these distinct modes. Notably, the average improvement in training-based settings (+10.37) significantly exceeds that in prompting-based settings (+3.28). This disparity suggests that while EVU serves as an effective inference-time guidance, its full potential is unleashed when the backbone model is allowed to internalize the active belief update process, leading to more substantial performance gains.

6 Analyses and Discussions

6.1 Variant Analysis

To investigate the efficacy of belief intervention (BI), we examined distinct intervention strategies injected before the thought-action generation process. These include: **NoneBI**, which serves as the baseline without any intervention; **DBI** (Direct Belief Intervention), where the belief state is generated directly; **OBI** (Observation-based Belief Intervention), which compels the agent to reiterate the recent observation prior to forming a belief; **ReflAct**, which encourages the agent to reflect on its progress relative to the goal; and **VAGEN** (Wang et al., 2025c), which tasks the agent with predicting the environmental state following a potential action. Please refer to Appendix F for more details.

The comparative results are presented in Figure 7. First, we observe that methods incorporating belief intervention consistently outperform the baseline (NoneBI) across the majority of settings. This trend underscores the fundamental efficacy of explicit belief modeling in enhancing task performance. Second, and more importantly, our proposed EVU achieves superior performance compared to all other intervention variants across diverse paradigms and evaluation splits. This consistent dominance suggests that the EVU mechanism provides a more robust and accurate strategy for belief generation than simple repetition or reflection, thereby serving as an effective intervention strategy.

6.2 Analysis on Belief Inertia Mitigation

Can EVU effectively mitigate the belief inertia phenomenon? To investigate this, we conduct a dedicated analysis to examine how our method behaves on the observational neglect cases collected in Section 3.2. Specifically, we employ our probing method to detect the agent's belief immedi-

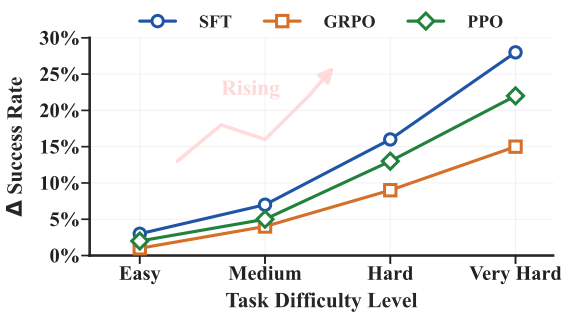

Figure 10: Relative improvement of our method compared to baselines (SFT, GRPO, PPO) across different levels of task difficulty.

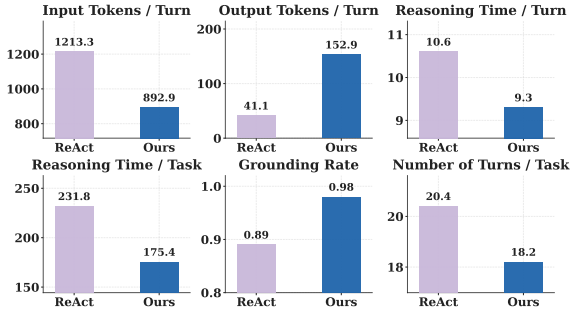

Figure 11: Comparison between ReAct and Ours in terms of computational overhead.

ately prior to the decision-making phase. (See Appendix G for detailed experimental configurations.) As illustrated in Figure 8, we observe that the true belief values for our method are consistently positive. This indicates that the agent's internal belief state aligns with the actual environmental state, successfully overcoming belief inertia. Furthermore, compared to other variants, EVU exhibits the highest true belief values. This superiority demonstrates that our approach not only corrects the belief, but also achieves the highest level of confidence in the true state of the environment.

How does EVU take effect to mitigate the belief inertia? To understand the internal mechanism, we conduct further analysis to observe the agent's belief dynamic evolution using the same set of observational neglect cases mentioned above. We probe the agent's belief at different stages within our EVU framework. Please refer to the Appendix G for more details. As illustrated in Figure 9, the agent's belief value starts in the negative region and gradually ascends. While the estimation and verification phases push the belief towards and across the boundary, respectively, it is the final update phase that significantly boosts the value to fully align with the ground truth. This indicates that the update stage is the decisive factor for synchronization, effectively building upon the foundations laid by the preceding phases.

6.3 Impact on Task Difficulty

We further investigate the necessity of active belief intervention as the task difficulty increases. Difficult tasks inherently involve longer interaction horizons, requiring the agent to process more observations. Consequently, the ability to maintain accurate and synchronized belief states becomes increasingly critical during these interactions. To verify this, we evaluate the relative success rate improvement (Δ Success Rate) of our method compared to three training baselines (SFT, GRPO, and PPO) across four difficulty levels (see Appendix H for details).

As shown in Figure 10, our approach consistently outperforms the baselines across all settings. More importantly, the performance gap exhibits a clear rising trend: as the difficulty escalates from "Easy" to "Very Hard," the relative improvement becomes significantly more pronounced. This validates that active belief intervention enables the agent to understand the environment more deeply, thereby preventing compounding errors.

7 Analysis on Computational Overhead

Do the performance gains of EVU come with substantial additional computational overhead? To investigate this, we compare the standard ReAct baseline and ReAct+EVU on the Qwen2.5-3B backbone. We report the average input/output tokens, reasoning latency, grounding rate, and average number of turns per task.

As illustrated in Figure 11, although EVU naturally increases the number of output tokens per turn due to the generation of the *Estimate*, *Verify*, and *Update* components, it substantially reduces the input context length by compressing the verbose interaction history into a concise belief state. As a result, the average total token consumption per task decreases from 1213.3 to 892.9, while the average reasoning latency per task also decreases from 231.8 to 175.4. In addition, EVU improves the grounding rate from 0.89 to 0.98 and reduces the average number of turns from 20.4 to 18.2, indicating that the agent avoids invalid exploration and plans more effectively. These results show that EVU does not incur a larger overall computational burden; instead, it improves efficiency by replacing

Method	# Reasoning Tokens	SR (%)
Multiple Reflection	153.3	46.3
EVU (Ours)	138.0	49.8

Table 2: Comparison between our EVU and the Multiple Reflection method.

low-quality exploration with more grounded and directed reasoning.

Are the gains of EVU simply due to increased inference computation (test-time scaling)? We further conducted a controlled experiment on ALFWorld with DeepSeek V3.2 in the prompting-based setting. Specifically, we compare EVU against a *Multiple Reflection* baseline, where the agent is prompted to reflect multiple times to deliberately increase the reasoning token budget. This setting allows us to examine whether the gain of EVU comes merely from consuming more reasoning tokens, or from the specific structure of the EVU process itself.

As shown in Table 2, EVU achieves a higher success rate than *Multiple Reflection* while using fewer reasoning tokens, which first rules out the explanation that the gain simply comes from spending more computation at inference time. More importantly, this comparison also highlights a key distinction between EVU and standard reflection. *Multiple Reflection* is retrospective: it asks the agent to reconsider whether it may have made a mistake after the fact. In contrast, EVU is predictive and discrepancy-driven. The *Estimate* step makes the agent explicitly predict the expected outcome before observing the new state, giving the *Verify* step a concrete reference point. This allows EVU to detect a "surprise signal," i.e., the mismatch between expectation and observation. The *Update* step then revises the belief state accordingly, rather than merely triggering another round of generic self-correction. This predictive mechanism is particularly important for overcoming belief inertia: without an explicit prior estimate, standard reflection may remain trapped in the agent's previous belief and fail to recognize the reality gap. Therefore, EVU improves performance not by encouraging longer reasoning chains, but by enforcing a simple yet effective belief-correction process that is more targeted than standard reflection.

8 Related Work

Embodied Planning. Recent advancements in Large Language Models (LLMs) have empowered embodied agents to engage in complex embodied planning (Li et al., 2025; Yang et al., 2025b; Liao et al., 2025). To facilitate effective decision-making, existing studies employ diverse strategies: Prompting methods (Yao et al., 2022; Shinn et al., 2023; Yao et al., 2023) structure reasoning at inference time, supervised finetuning (Chen et al., 2023; Wang et al., 2025b; Qiao et al., 2024) internalize expert priors, and reinforcement learning refines policy from reward signals (Song et al., 2024; Chen et al., 2025; Wang et al., 2025a; Zhang et al., 2025). However, these methods primarily prioritize action optimization, often neglecting the critical need to maintain a reliable internal world model amidst dynamic environmental changes (Huang et al., 2023; Wang et al., 2024; Kim et al., 2025).

Agent Beliefs. To achieve goals, agents are required to maintain and update their internal beliefs during interaction. Prior work has highlighted multiple complementary facets of such beliefs. First, agents often rely on task belief—estimates of progress and knowledge—to support long-horizon planning (Qiao et al., 2024; Wang et al., 2024; Zhang et al., 2024). Second, in human-agent interaction, agents capture users' latent intent to interpret ambiguous instructions (Lin et al., 2025; Ramrakhya et al., 2025). Third, in cooperative multi-agent settings, agents track others' capabilities, objectives, and likely future actions to enable coordination (Fan et al., 2025; Lică et al., 2024; Wang et al., 2026). In this work, we focus on agent beliefs regarding the evolving environment, which grounds reasoning under environment state changes during interactions.

9 Conclusion

In this work, we identify and formalize *belief inertia* as a key failure mode of LLM-based embodied agents, where they stubbornly adhere to prior beliefs despite explicit observations. To address this issue, we advocate active belief intervention and instantiate it with the Estimate-Verify-Update (EVU) mechanism. By integrating EVU into both prompting-based and training-based methods, we mitigate belief inertia effectively, thereby obtaining consistent improvements in task performance across multiple embodied agent benchmarks.

Limitations

While our approach demonstrates superior performance compared to baseline methods, it is important to acknowledge the limitations of our current work as follows:

(1) Dependency on Observation Quality: Our method relies on the quality and granularity of environmental observations to update its belief dynamics. In scenarios with extremely sparse, noisy, or ambiguous feedback, where the ground truth is difficult to discern even with active reasoning, the agent's belief updates may become unstable. Future work could explore more robust active belief dynamics that can better handle uncertainty and noise.

(2) Limited Exploration of Model Variants: Due to computational resource constraints, our experiments on prompting methods were primarily conducted using DeepSeek V3.2, and we did not extensively evaluate the approach across a broader range of LLM backbones. Furthermore, while our work addresses the fundamental challenge of belief updating and is expected to be compatible with various methods, we have not yet explored alternative designs to further facilitate accurate belief generation, such as integrating dense reward shaping or auxiliary supervision signals. Future work could incorporate these advanced designs to refine the belief intervention process.

Ethics Statement

This work aims to develop LLM-based embodied agents within simulated environments. The VirtualHome and ALFWorld environment setup and related data strictly follow the specifications of VirtualHome (Puig et al., 2018), ALFWorld (Shridhar et al., 2020), and ScienceWorld (Wang et al., 2022). We utilize VirtualHome v2.3.0[1] (MIT license[2]), ALFWorld[3] (MIT license[4]) and ScienceWorld[5] (MIT license[6]) to conduct our experiments. All the LLMs we use for fine-tuning are open-source, and we strictly follow the protocols for the academic use of these models. Additionally, while AI assistants (e.g., Cursor and ChatGPT) were partially utilized for code optimization and linguistic refinement, we affirm that all core content and findings in this paper are the original work of the authors.

[1] https://github.com/xavierpuigf/virtualhome/tree/master

[2] https://github.com/xavierpuigf/virtualhome/blob/master/LICENSE

[3] https://github.com/alfworld/alfworld

[4] https://github.com/alfworld/alfworld/blob/master/LICENSE

[5] https://github.com/allenai/ScienceWorld

[6] https://github.com/allenai/ScienceWorld/blob/main/LICENSE

Acknowledgements

This work was supported by the Research Grants Council of Hong Kong (15209724, 15205325), and also in part by the PolyU Postdoc Matching Fund Scheme (4-W40Z). The authors would like to thank the anonymous reviewers for their valuable feedback and constructive suggestions.

References

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. 2023. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*.

Hanyang Chen, Mark Zhao, Rui Yang, Qinwei Ma, Ke Yang, Jiarui Yao, Kangrui Wang, Hao Bai, Zhenhailong Wang, Rui Pan, and 1 others. 2025. Era: Transforming vlms into embodied agents via embodied prior learning and online reinforcement learning. *arXiv preprint arXiv:2510.12693*.

Xianzhe Fan, Xuhui Zhou, Chuanyang Jin, Kolby Nottingham, Hao Zhu, and Maarten Sap. 2025. Somitom: Evaluating multi-perspective theory of mind in embodied social interactions. *arXiv preprint arXiv:2506.23046*.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*.

Pascale Fung, Yoram Bachrach, Asli Celikyilmaz, Kamalika Chaudhuri, Delong Chen, Willy Chung, Emmanuel Dupoux, Hongyu Gong, Hervé Jégou, Alessandro Lazaric, and 1 others. 2025. Embodied ai agents: Modeling the world. *arXiv preprint arXiv:2506.22355*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Wenlong Huang, Fei Xia, Dhruv Shah, Danny Driess, Andy Zeng, Yao Lu, Pete Florence, Igor Mordatch, Sergey Levine, Karol Hausman, and 1 others. 2023. Grounded decoding: Guiding text generation with grounded models for embodied agents. *Advances in Neural Information Processing Systems*, 36:59636–59661.

Jeonghye Kim, Sojeong Rhee, Minbeom Kim, Dohyung Kim, Sangmook Lee, Youngchul Sung, and Kyomin

Jung. 2025. Reflact: World-grounded decision making in llm agents via goal-state reflection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33421–33453.

Manling Li, Yunzhu Li, Jiayuan Mao, and Wenlong Huang. 2025. Foundation models meet embodied agents. In *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 5: Tutorial Abstracts)*, pages 15–24, Albuquerque, New Mexico. Association for Computational Linguistics.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, and 1 others. 2024. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534.

Yi Liao, Yu Gu, Yuan Sui, Zining Zhu, Yifan Lu, Guohua Tang, Zhongqian Sun, and Wei Yang. 2025. Think in games: Learning to reason in games via reinforcement learning with large language models. *arXiv preprint arXiv:2508.21365*.

Mircea Lică, Ojas Shirekar, Baptiste Colle, and Chirag Raman. 2024. Mindforge: Empowering embodied agents with theory of mind for lifelong collaborative learning. *arXiv preprint arXiv:2411.12977*.

Aly Lidayan, Jakob Bjorner, Satvik Golechha, Kartik Goyal, and Alane Suhr. 2025. Abbel: Llm agents acting through belief bottlenecks expressed in language. *arXiv preprint arXiv:2512.20111*.

Xingyao Lin, Xinghao Zhu, Tianyi Lu, Sicheng Xie, Hui Zhang, Xipeng Qiu, Zuxuan Wu, and Yu-Gang Jiang. 2025. Ask-to-clarify: Resolving instruction ambiguity through multi-turn dialogue. *arXiv preprint arXiv:2509.15061*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. 2025. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502.

Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024. Agent planning with world knowledge model. *Advances in Neural Information Processing Systems*, 37:114843–114871.

Ram Ramrakhya, Matthew Chang, Xavier Puig, Ruta Desai, Zsolt Kira, and Roozbeh Mottaghi. 2025. Grounding multimodal llms to embodied agents that ask for help with reinforcement learning. *arXiv preprint arXiv:2504.00907*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

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

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7584–7600.

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

Hanlin Wang, Chak Tou Leong, Jian Wang, and Wenjie Li. 2024. E2cl: exploration-based error correction learning for embodied agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7626–7639.

Hanlin Wang, Chak Tou Leong, Jiashuo Wang, Jian Wang, and Wenjie Li. 2025a. Spa-rl: Reinforcing llm agents via stepwise progress attribution. *arXiv preprint arXiv:2505.20732*.

Hanlin Wang, Jian Wang, Chak Tou Leong, and Wenjie Li. 2025b. Steca: Step-level trajectory calibration for llm agent learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11597–11614.

Jiashuo Wang, Jiawen Duan, Jian Wang, Kaitao Song, Chunpu Xu, Johnny K. W. Ho, Fenggang Yu, Wenjie Li, and Johan F. Hoorn. 2026. Foresight optimization for strategic reasoning in large language models. *Preprint*, arXiv:2604.13592.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, and 1 others. 2025c. Vagen: Reinforcing world model reasoning for multi-turn vlm agents. *arXiv preprint arXiv:2510.16907*.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298.

Di Wu, Jiaxin Fan, Junzhe Zang, Guanbo Wang, Wei Yin, Wenhao Li, and Bo Jin. 2025. Reinforced reasoning for embodied planning. *arXiv preprint arXiv:2505.22050*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, and 1 others. 2025b. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *International Conference on Machine Learning*, pages 70576–70631. PMLR.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. 2024. Agent-pro: Learning to evolve via policy-level reflection and optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5348–5375.

Wenqi Zhang, Mengna Wang, Gangao Liu, Xu Huixin, Yiwei Jiang, Yongliang Shen, Guiyang Hou, Zhe Zheng, Hang Zhang, Xin Li, and 1 others. 2025. Embodied-reasoner: Synergizing visual search, reasoning, and action for embodied interactive tasks. *arXiv preprint arXiv:2503.21696*.

Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Large language models as commonsense knowledge for large-scale task planning. *Advances in neural information processing systems*, 36:31967–31987.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*.

A Reliability Evaluation

To verify that our probing method accurately reflects the agent's true internal belief state—rather than generating hallucinations or unrelated outputs—we conducted a consistency analysis. We hypothesize that if the probing method is reliable, the belief elicited by the probe at timestep t should be highly consistent with the agent's explicit reasoning (Thought) manifested in the subsequent timestep.

We validate the reliability by comparing the probing results against the agent's subsequent behavior. The specific procedure is as follows:

- **Probing at t:** At a given timestep t, we focus on a specific object variable o_t. We query the agent's belief upon receiving obervation o_t using our probing method to obtain a binary answer (Yes or No), denoted as the *Predicted Label*.
- **Action at t:** We then allow the agent to process observation o_t and conduct next step reasoning, denoted as a_t. We extract the agent's explicit understanding regarding the specific object variable o_t from the thought component of a_t, denoted as the *True Label*.
- **Comparison:** We compare the consistency between the probed answer and the subsequent thought across 100 sampled cases.

We visualized the consistency between the probed beliefs and the agent's thoughts using a confusion matrix. As illustrated in Figure 12, our probing method demonstrates a high degree of fidelity with the agent's internal reasoning at next step:

- When the agent's subsequent thought indicates a negative state (*No*), the probe correctly identifies this belief **96.0%** of the time.
- When the agent's subsequent thought indicates a positive state (*Yes*), the probe correctly aligns with this belief **92.0%** of the time.

The low off-diagonal error rates (4.0% and 8.0%) indicate minimal discrepancy. This strong alignment confirms that our method is reliable: it successfully externalizes the agent's latent beliefs without significant distortion, validating its utility for interpreting the agent's decision-making process.

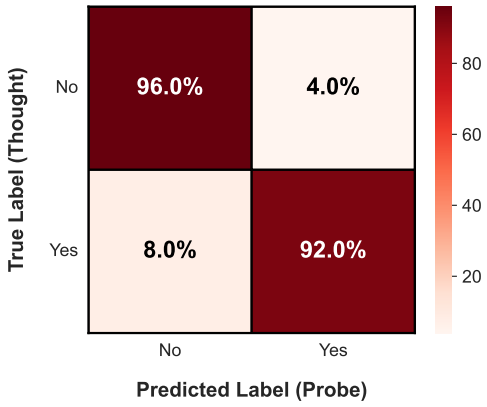

Figure 12: Confusion Matrix evaluating the consistency between the Probed Answer (Predicted Label) and the Agent's Subsequent Thought (True Label). The high values on the diagonal indicate that the probing method reliably reflects the agent's internal beliefs.

B Datasets and Preprocessing

ALFWorld is an interactive text-based environment that parallels the embodied worlds found in the ALFRED dataset. In this domain, agents are tasked with exploring a simulated household to complete high-level instructions, such as put a clean apple in the fridge." The dataset includes both seen" splits for in-distribution evaluation and "unseen" splits to test out-of-distribution generalization. For the ReAct baseline, we utilize the SFT data generated by Song et al. (2024).

ScienceWorld is a complex text-based virtual environment designed to simulate elementary science experiments. It encompasses various distinct task types, such as thermodynamics and electrical circuits, requiring agents to ground their understanding of scientific concepts through practical, embodied interaction. Similar to ALFWorld, we adopt the training trajectories for the ReAct baseline provided by the Song et al. (2024). To ensure computational efficiency, we Task-8, Task-9, Task-1 (boil/freeze), Task-4 (grow fruit/plant), Task-5 (clean), Task-7 (paint), and Task-10 (decorate) due to their excessively long task-solving trajectories.

VirtualHome is a platform that simulates complex daily household activities, where agents execute programs to interact with objects and the environment. Unlike the previous datasets, we align the SFT data for the ReAct baseline and the experimental setup with the methodology established in STeCa (Wang et al., 2025b). Consistent with the STeCA setting, we filter the dataset significantly; specifically, we remove approximately half of the

training and testing data instances. This reduction is performed because many tasks in the original dataset are highly similar, ensuring a more efficient evaluation.

Across all three domains, the environments provide binary final rewards, where a reward of 1 indicates successful task completion and 0 indicates failure. Consequently, we report the average reward across the tested tasks as the success rate.

Dataset Statistics We summarize the detailed statistics of the three datasets in Table 3. The table reports the number of instances in the training set, as well as the Test-Seen" (in-distribution) and Test-Unseen" (out-of-distribution) evaluation sets. It also lists the average number of interaction turns required for expert trajectories, which serves as an indicator of task complexity across the different environments.

Dataset	#Train	#Test-Seen	#Test-Unseen	#Turns
ALFWorld	2851	140	134	7.97
VirtualHome	2460	125	125	8.79
ScienceWorld	1253	151	161	9.64

Table 3: Statistics of datasets. "Test-Seen" and "Test-Unseen" are test set with seen and unseen scenarios respectively. "#Turns" denotes the average number of interaction turns for the expert trajectories.

C Additional Details about Baselines

In this section, we provide additional implementation details for the baseline methods, categorized by their prompting strategies and training methodologies.

Prompting Settings. We consider three frameworks: (1) No-Thinking(Ma et al., 2025): The agent generates an action directly at each time step without any reasoning step. (2) Plan-and-Act(Kim et al., 2025): The agent conducts reasoning only at the first step and outputs actions without further thoughts in subsequent steps. and (3) ReAct (Yao et al., 2022): The agent first reasons about the next action at each time step and then generates an action.

Training Settings. We employ three distinct approaches: (1) SFT(Chen et al., 2023): The model is fine-tuned using standard supervised learning on a dataset of expert trajectories. (2) PPO(Schulman et al., 2017): A proximal policy optimization algorithm that utilizes a separate value network (critic) to reduce variance and stabilize training. and (3) GRPO (Shao et al., 2024): A group relative policy optimization method that eliminates the need for a critic model by estimating the baseline from the average reward of a group of sampled outputs for the same input.

D Additional Comparisons with Advanced Baselines

we provide additional comparisons with more advanced baselines to further clarify the role of EVU. Specifically, we aim to answer two questions: (1) whether the gain of EVU mainly comes from better context management or from its active belief intervention mechanism, and (2) whether EVU can also benefit stronger agentic frameworks beyond a standard ReAct-style policy. To this end, we conduct three additional experiments: a short-context control experiment, a comparison against a history-summarization baseline, and an integration with a search-based agentic planner.

Is EVU more than context management? To disentangle belief inertia from long-context crowding, we first revisit the failure cases identified in our analysis and truncate the interaction history to retain only the most recent two turns, so that the critical observation remains explicitly visible and no long-context retrieval is required. Even under this short-context setting, 95% of the cases still exhibit belief inertia, where the agent ignores the immediate contradictory observation and continues to follow its prior belief. This result suggests that belief inertia is not merely a symptom of crowded context, but a distinct failure to integrate contradictory evidence. We further compare EVU against MEM1 (Zhou et al., 2025), a representative history-summarization baseline that manages context through memory compression but does not include an explicit estimate-verify-update loop. As shown in Table 4, EVU consistently outperforms MEM1 on ScienceWorld, improving performance from 67.1 to 70.9 on seen tasks and from 64.9 to 70.8 on unseen tasks. These results indicate that passive context management alone is insufficient to overcome belief inertia; the key benefit comes from actively forcing the agent to estimate, verify, and update its belief state.

Can EVU benefit stronger agentic planners? To evaluate whether EVU is compatible with more advanced agentic systems, we integrate EVU with LLM-MCTS (Zhao et al., 2023) on VirtualHome. This experiment examines whether EVU is com-

Method	Seen	Unseen
MEM1	67.1	64.9
EVU (Ours)	70.9	70.8

Table 4: Comparison with the history-summarization baseline MEM1 on ScienceWorld. EVU consistently outperforms passive context management on both seen and unseen tasks.

plementary to search-based planning rather than being tied to a simple action-generation policy. As shown in Table 5, incorporating EVU consistently improves over the strong LLM-MCTS baseline, raising performance from 28.8 to 31.2 on seen tasks and from 25.6 to 28.8 on unseen tasks. This result shows that EVU acts as a generalizable cognitive module that complements advanced planning algorithms by helping the agent maintain a more accurate belief state during search and execution, rather than being useful only in a standalone intervention setting.

Method	Seen	Unseen
LLM-MCTS	28.8	25.6
LLM-MCTS + EVU (Ours)	31.2	28.8

Table 5: Comparison with the search-based agentic planner LLM-MCTS on VirtualHome. EVU consistently improves over a stronger planning baseline.

E Additional Implementation Details

Our training infrastructure for both Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) is built upon the verl-agent framework (Feng et al., 2025). All key hyperparameters for both phases are summarized in Table 6. The templates corresponding to the ALFWorld, ScienceWorld, and VirtualHome environments are illustrated in Figure 13, Figure 14, and Figure 15, respectively.

F Experimental Setup for Different Belief Intervention Strategies

In this section, we provide a detailed formulation of the belief intervention strategies employed in our experiments. We first define a general framework for belief intervention and then describe how each specific strategy instantiates this framework.

F.1 General Formulation

We consider an agent interacting with an environment to achieve a goal G. At any time step t, given the interaction history and current observation (collectively denoted as Context C_t), the standard agent directly generates a thought chain and an action.

Hyperparameter	**Value**
SFT Phase	
Learning Rate	1e−5
Scheduler	Cosine
Epochs	3
RL Phase	
Group Size (G)	6
Learning Rate	1×10^{-6}
Total Steps	250
KL Coefficient (β)	0.01
Temperature (rollout)	0.8
Temperature (validation)	0.0
Common / Other	
Max Prompt Length	5120
Max Response Length	512
Micro Batch Size	4
Mini Batch Size	32
Invalid Action Penalty	0.1
Gradient Checkpointing	True
Tensor Parallel Size	2
Temperature (eval)	0.0

Table 6: Key hyperparameters for SFT and RL training.

To investigate the role of environmental reasoning, we introduce an intermediate **Belief Intervention (BI)** module. The decision-making process is decomposed into two phases:

1. **Belief Generation:** The agent first generates a belief state content $\mathcal{B}$ based on a specific intervention strategy $\mathcal{S}$.
2. **Action Generation:** The agent then generates the thought and action A_t conditioned on both the context and the generated belief.

Formally, this process can be represented as:

$$C_t \xrightarrow{\text{Strategy } \mathcal{S}} \mathcal{B}_t \rightarrow A_t \tag{9}$$

where $\mathcal{B}_t$ varies depending on the definition of the strategy $\mathcal{S}$.

F.2 Strategy Instantiations

Below, we detail the five strategies compared in the main text. To illustrate the differences, we provide a running example where the agent's goal is to *"Put the apple in the fridge"*, and the agent is currently standing in front of a **closed fridge** holding an apple.

NoneBI (Baseline). This represents the standard setting without any explicit belief intervention. The agent proceeds directly from observation to action.

- **Formulation:** $\mathcal{B}_t = \emptyset$ (Empty set).
- **Mechanism:** The thought-action generation is solely dependent on the context C_t.

  Example Output:
  Thought: The fridge is closed. I need to open it to put the apple inside.
  Action: `Open(Fridge)`

DBI (Direct Belief Intervention). DBI requires the agent to explicitly generate the current belief state regarding the environment before planning.

- **Formulation:** $\mathcal{B}_t = S_t$, where S_t is the description of the current environmental state relevant to the goal.

  Example Output:
  [Belief]: The fridge is currently closed. I am holding the apple.
  Thought: Since the fridge is closed, I must open it first.
  Action: `Open(Fridge)`

OBI (Observation-based Belief Intervention). OBI compels the agent to ground its belief in raw perception. It forces the agent to reiterate the recent observation text explicitly before summarizing the belief state.

- **Formulation:** $\mathcal{B}_t = \{O_t \oplus S_t\}$, where O_t is the reiteration of the observation and S_t is the inferred state.

  Example Output:
  [Belief]: *Observation:* I see a white fridge door in front of me. *State:* The fridge is closed and I cannot see inside.
  Thought: I need to access the interior of the fridge.
  Action: `Open(Fridge)`

ReflAct. ReflAct (Kim et al., 2025) encourages the agent to reflect on its dynamic progress rather than just the static state. It prompts the agent to evaluate what has been accomplished and what remains to be done relative to the goal G.

- **Formulation:** $\mathcal{B}_t = \text{Reflect}(S_t, G)$, focusing on the gap between current state and goal.

  Example Output:
  [Belief]: I have successfully picked up the apple. However, the goal is to put it in the fridge, and the fridge is currently closed.
  Thought: The next logical step is to open the fridge.
  Action: `Open(Fridge)`

VAGEN. VAGEN (Wang et al., 2025c) tasks the agent with a predictive "look-ahead" process. The agent must identify the current belief state, propose a hypothetical action, and then predict the resulting environmental state.

- **Formulation:** $\mathcal{B}_t = \{S_t \xrightarrow{\text{Hypothetical Action } \tilde{a}} S_{t+1}\}$, representing a mental simulation of the immediate future.

  Example Output:
  [Belief]: *Current State:* The fridge is closed. *Hypothetical Action:* Open the fridge. *Predicted Future State:* The fridge door will be open, revealing the contents inside.
  Thought: This action will allow me to place the apple inside.
  Action: `Open(Fridge)`

G Experimental Setup for Belief Inertia Mitigation Analysis

To investigate the internal mechanism of EVU in mitigating belief inertia, we conduct a fine-grained probing analysis using the identical set of *observational neglect* cases described in Section 3.2. These cases represent critical moments where the agent's action a_t fails to reflect the current observation o_t, indicating a strong inertia from prior beliefs.

We track the belief evolution by probing the agent at four distinct stages of our active belief dynamic process. For each stage, we construct the input prompt by progressively accumulating the intermediate reasoning outputs. Specifically, the probing stages are defined as follows: (1) **Current Belief**: This represents the baseline state prior to any active intervention, probed using the historical context $(B_{t-1}, a_{t-1}, o_{t-1})$; (2) **Estimated Belief**: This captures the belief state immediately after the estimation phase, where the context is augmented with the generated estimation to form $(B_{t-1}, a_{t-1}, o_{t-1}, E_t)$; (3) **Verification Belief**: This reflects the state after the agent validates the estimation against the observation, probed under the context $(B_{t-1}, a_{t-1}, o_{t-1}, E_t, V_t)$; and (4) **Updated Belief**: This represents the final consolidated

state after the update phase, probed using the complete context $(B_{t-1}, a_{t-1}, o_{t-1}, E_t, V_t, B_t)$. By comparing the belief values across these stages, we quantify the contribution of each component in correcting the belief inertia.

H Experimental Setup for Task Difficulty Analysis

To systematically evaluate the robustness of active belief dynamics as tasks become more challenging, we conduct a difficulty analysis within the ALFWorld environment. We classify task difficulty based on the minimum number of subgoals required to achieve the final objective, as tasks with more subgoals necessitate longer interaction horizons. We obtain four distinct difficulty levels: *Easy* (0–4 subgoals), *Medium* (5–8 subgoals), *Hard* (9–12 subgoals), and *Very Hard* (13–16 subgoals). We randomly sampled a total of 200 tasks to ensure diverse coverage across these difficulty levels. For each level, we compare the success rate of our proposed method against three training baselines: SFT, GRPO, and PPO. To quantify the advantage, we calculate the relative success rate improvement (Δ Success Rate) of our method over each baseline.

ALFWorld Prompt Template

Interact with a household to solve a task. Imagine you are an intelligent agent in a household environment and your target is to perform actions to complete the task goal.
At each step, you will be given task goal, action history and the last turn's information (Reason, Belief State, Thought, and Action).
You need to process the information in a specific order:

1. **Reason**: Analyze the last action and the observation in one or two concise sentences. What did you expect to see? What did you actually see? Does this confirm or contradict your previous belief?
2. **Belief State**: State where the agent is, what it is holding, and the known status of goal-related objects. Do NOT list irrelevant objects.
3. **Thought**: Plan your future actions based on the updated belief.
4. **Action**: Output your next action.

The available actions are:

1. go to (recep)
2. task (obj) from (recep)
3. put (obj) in/on (recep)
4. open (recep)
5. close (recep)
6. toggle (obj) (recep)
7. clean (obj) with (recep)
8. heat (obj) with (recep)
9. cool (obj) with (recep)

where (obj) and (recep) correspond to objects and receptacles.
After your each turn, the environment will give you immediate feedback based on which you plan your next few steps. If the environment output "Nothing happened", that means the previous action is invalid and you should try more options.

Your response should use the following format:

```
Reason: <Analyze expectation vs. actual observation to update your understanding>
Belief State: <your belief state>
Thought: <your thoughts>
Action: <your next action>
```

Your task is to complete the task goal: {task_goal}
Below is the action history and the last turn's information:
Action History: {action_history}
Last Turn's Information: {last_turn_information}

Figure 13: Prompt template of our method on the ALFWorld benchmark.

VirtualHome Prompt Template

Interact with a household to solve a task. Imagine you are an intelligent agent in a household environment and your target is to perform actions to complete the task goal. At the beginning of your interactions, you will be given the detailed description of the current environment and your goal to accomplish.
At each step, you will be given task goal, action history and the last turn's information (Reason, Belief State, Thought, and Action).
You need to process the information in a specific order:

1. **Reason**: Analyze the last action and the observation in one or two concise sentences. What did you expect to see? What did you actually see? Does this confirm or contradict your previous belief?
2. **Belief State**: State where the agent is, what it is holding, and the known status of goal-related objects. Do NOT list irrelevant objects.
3. **Thought**: Plan your future actions based on the updated belief.
4. **Action**: Output your next action.

The available actions are:

1. walk to (obj)
2. run to (obj)
3. grab (obj)
4. open (obj)
5. close (obj)
6. put (obj) on (recep)
7. put (obj) in (recep)
8. switch on (obj)
9. switch off (obj)
10. drink (obj)
11. look at (obj)
12. sit on (obj)
13. stand up
14. watch (obj)
15. wipe (obj)
16. type on (obj)
17. wash (obj)
18. cut (obj)
19. eat (obj)
20. sleep
21. wake up
22. plug in (obj)
23. plug out (obj)
24. pour (obj) into (recep)
25. move (obj)
26. release
27. turn to (obj)

After your each turn, the environment will give you immediate feedback based on which you plan your next few steps. If the environment output "Nothing happened", that means the previous action is invalid and you should try more options.

Your response should use the following format:

```
Reason: <Analyze expectation vs. actual observation to update your understanding>
Belief State: <your belief state>
Thought: <your thoughts>
Action: <your next action>
```

Your task is to complete the task goal: {task_goal}
Below is the action history and the last turn's information:
Action History: {action_history}
Last Turn's Information: {last_turn_information}

Figure 14: Prompt template of our method on the VirtualHome benchmark.

ScienceWorld Prompt Template

You are a helpful assistant to do some scientific experiment in an environment. In the environment, there are several rooms: kitchen, foundry, workshop, bathroom, outside, living room, bedroom, greenhouse, art studio, hallway. You should explore the environment and find the items you need to complete the experiment. You can teleport to any room in one step. All containers in the environment have already been opened, you can directly get items from the containers.
At each step, you will be given task goal, action history and the last turn's information (Reason, Belief State, Thought, and Action).
You need to process the information in a specific order:

1. **Reason**: Analyze the last action and the observation in one or two concise sentences. What did you expect to see? What did you actually see? Does this confirm or contradict your previous belief?
2. **Belief State**: State where the agent is, what it is holding, and the known status of goal-related objects. Do NOT list irrelevant objects.
3. **Thought**: Plan your future actions based on the updated belief.
4. **Action**: Output your next action.

The available actions are:

- `open OBJ`: open a container
- `close OBJ`: close a container
- `activate OBJ`: activate a device
- `deactivate OBJ`: deactivate a device
- `connect OBJ to OBJ`: connect electrical components
- `disconnect OBJ`: disconnect electrical components
- `use OBJ [on OBJ]`: use a device/item
- `look around`: describe the current room
- `examine OBJ`: describe an object in detail
- `look at OBJ`: describe a container's contents
- `read OBJ`: read a note or book
- `move OBJ to OBJ`: move an object to a container
- `pick up OBJ`: move an object to the inventory
- `pour OBJ into OBJ`: pour a liquid into a container
- `mix OBJ`: chemically mix a container
- `teleport to LOC`: teleport to a specific room
- `focus on OBJ`: signal intent on a task object
- `wait`: task no action for 10 steps
- `wait1`: task no action for a step

Your response should use the following format:

```
Reason: <Analyze expectation vs. actual observation to update your understanding>
Belief State: <your belief state>
Thought: <your thoughts>
Action: <your next action>
```

Your task is to complete the task goal: {task_goal}
Below is the action history and the last turn's information:
Action History: {action_history}
Last Turn's Information: {last_turn_information}

Figure 15: Prompt template of our method on the ScienceWorld benchmark.