HISR: Hindsight Information Modulated Segmental Process Rewards For Multi-turn Agentic Reinforcement Learning

Anonymous ACL submission

Abstract

While large language models excel in diverse domains, their performance on complex long-horizon agentic decision-making tasks remains limited. Most existing methods concentrate on designing effective reward models (RMs) to advance performance via multi-turn reinforcement learning. However, they suffer from delayed propagation in sparse outcome rewards and unreliable credit assignment with potentially overly fine-grained and unfocused turn-level process rewards. In this paper, we propose (**HISR**) exploiting **H**indsight **I**nformation to modulate **S**egmental process **R**ewards, which closely aligns rewards with sub-goals and underscores significant segments to enhance the reliability of credit assignment. Specifically, a segment-level process RM is presented to assign rewards for each sub-goal in the task, avoiding excessively granular allocation to turns. To emphasize significant segments in the trajectory, a hindsight model is devised to reflect the preference of performing a certain action after knowing the trajectory outcome. With this characteristic, we design the ratios of sequence likelihoods between hindsight and policy model to measure action importance. The ratios are subsequently employed to aggregate segment importance scores, which in turn modulate segmental process rewards, enhancing credit assignment reliability. Extensive experimental results on three publicly benchmarks demonstrate the validity of our method.

1 Introduction

Large language models (LLMs) have exhibited exceptional performance across diverse domains (Lu et al., 2024, 2025; Liu et al., 2025; Tian et al., 2025), even surpassing human performance in certain scenarios. However, their capabilities remain limited in complex long-horizon agentic decision-making tasks (e.g., household assistance (Shridhar et al., 2020; Puig et al., 2018)), where LLMs act as an agent to achieve multiple sub-goals in the task. To develop LLMs into comprehensive general artificial intelligence, it is crucial to explore effective solutions for agentic capabilities enhancement.

To achieve this goal, most existing methods concentrate on designing effective reward models (RMs) to advance performance via multi-turn reinforcement learning (RL) algorithms (e.g., Proximal Policy Optimization, PPO (Schulman et al., 2017)). The mainstream RMs can be broadly categorized into two paradigms: 1) **the outcome-based RM** (Ouyang et al., 2022; Shao et al., 2024) in Fig 1a. merely assigns a single reward at the end of trajectory according to the trajectory outcome. However, in the agentic tasks, the exacerbation of long-horizon and the characteristic of delayed reward make it more difficult to propagate the final reward to actions that occurred earlier, struggling to guide the optimization of decision-making in the environment (Wang et al., 2025a).

To alleviate this issue, 2) **the turn-level process RM** is presented to perform the credit assignment within the trajectory, which can be roughly divided into two categories based on the requirement of process reward labeling. One (Fig 1b.) utilizes finite Monte Carlo Tree Search (MCTS) (Wang et al., 2024; Zhang et al., 2025) or GPT-4 (Achiam et al., 2023; Lightman et al., 2023; Xie et al., 2025; Bi et al., 2025) to annotate the pseudo process rewards, yielding the noisy turn-level supervision. The others (Fig 1c.) are independent of process reward labels, instead leveraging the outcome of the trajectory to indirectly monitor the credit assignment (Wang et al., 2025a; Yu et al., 2024). Nevertheless, the former faces the difficulty of cost-effectively obtaining accurate process labels, while the latter completely neglects the process information (e.g., action importance), leading to unfocused credit assignment. Moreover, both assign rewards at turn-level granularity, which is potentially overly fine-grained for the sub-goal spanning multiple turns (Guo et al., 2025; Yin et al., 2025). This is exempli-

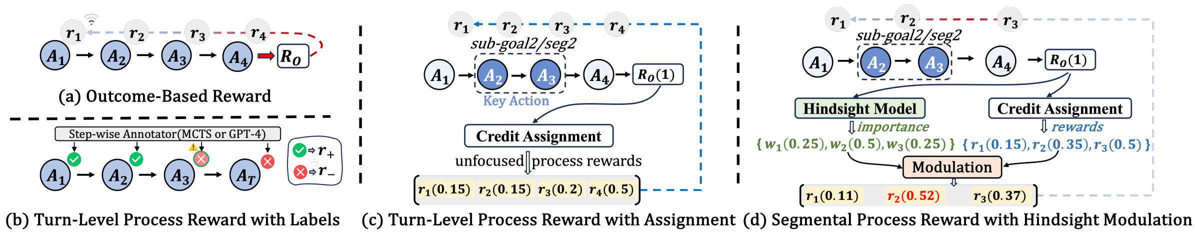

Figure 1: Comparison between mainstream reward models and our method in multi-turn reinforcement learning. A_i, R_0, r_i denote the decided action in i-th turn, the outcome of trajectory, and process rewards. (b) and (c) assign rewards at turn-level granularity. The unfocused process rewards in (c) denote that credit assignment fails to consider the action importance, while ours (d) aligns rewards with sub-goals and underscores significant segments.

fied in Fig 1(c-d), where sub-goal2 contains action A_2 and A_3. All of these lead to unreliable credit assignment.

In this paper, we propose (**HISR**) exploiting **H**indsight **I**nformation to modulate **S**egmental process **R**ewards for multi-turn agentic RL, which closely aligns rewards with sub-goals and underscores significant segments to enhance the reliability of credit assignment. Specifically, to avoid overly fine-grained allocation of turn-level rewards, we exploit GPT4o to split trajectories into several meaningful segments, where each segment corresponds to a sub-goal for the task. A segment-level process RM is then presented to assign rewards for each sub-goal in the task, facilitating the allocation of process rewards aligning with the sub-goal. Additionally, to emphasize significant segments in the trajectory, we devise a hindsight model that recovers each masked action in the turn to capture likelihoods, which reflects the preference of performing a certain action after knowing trajectory outcome. With this characteristic, the ratios of sequence likelihoods between the hindsight and policy model are designed to measure the action importance. The ratios are subsequently employed to aggregate segment importance scores, which in turn modulate segmental process rewards, underscoring significant segments to enhance credit assignment reliability.

Extensive experimental results on three publicly available agentic benchmarks and case studies demonstrate the effectiveness of our method. In summary, our main contributions include:

1) We propose (**HISR**) exploiting **H**indsight **I**nformation to modulate **S**egmental process **R**ewards for multi-turn RL, which closely aligns rewards with sub-goals and underscores significant segments to enhance credit assignment reliability.

2) We present a segment-level process RM to assign rewards for sub-goals in the task, avoiding overly fine-grained allocation of turn-level rewards. A hindsight model is also devised to capture likelihoods, which reflects the preference of performing a certain action after knowing trajectory outcome.

3) We design the ratios of sequence likelihoods between the hindsight and policy model to measure the action importance. The ratios are subsequently employed to aggregate segment importance scores, which in turn modulate the segmental process rewards, enhancing credit assignment reliability.

4) We carry out extensive experiments on three available agentic benchmarks. The newly achieved state-of-the-art performance and case study demonstrate the effectiveness of our method. The code will be released soon to foster future research.

2 Related Work

LLM-based Agent. It aims to equip LLMs with the capability of solving complex long-horizon agentic decision-making tasks (e.g., household assistance (Shridhar et al., 2020; Puig et al., 2018), webshop browsing (Yao et al., 2022a)). To achieve this goal, some preliminary explorations have been conducted in two mainstreams: (1) prompt engineering & supervised fine-tuning advocates adopting the data characterized by rigorous interactive paradigms to perform the behavior cloning for LLMs. For instance, Yao et al. (2022b) required the model to bind reasoning and acting in an interleaved manner, stimulating their synergistic effect. Shinn et al. (2023) introduced the reflection operation, alleviating the issues of hallucination and error propagation in the intermediate chain-of-thought reasoning (Wei et al., 2022). Despite improvements, these methods fail to enable LLM to interact with environment in real time and are plagued by the acquisition of diverse and high-quality trajectory data. To overcome this issue, researchers

(Wang et al., 2025a; Feng et al., 2025) instead leverage (2) multi-turn agentic RL (i.e., PPO (Schulman et al., 2017)) to optimize the decision-making of LLMs during the exploration in the environment. Compared with (1), the objective of multi-turn RL is maximizing the cumulative reward of the trajectory, which is more suitable for agentic tasks. In this paper, we adopt the latter and focus on pursuing a more reasonable reward allocation to enhance LLMs' agentic capabilities.

RM in Multi-turn RL. RM plays an important role in providing supervision signals in multi-turn RL. Early research adopts the outcome-based RM (Ouyang et al., 2022; Schulman et al., 2017), which merely assigns a single reward at the end of trajectory according to the trajectory outcome. However, in the agentic tasks, the exacerbation of long-horizon and the characteristic of delayed reward hinder propagating the final reward to actions that occurred earlier, struggling to determine whether an action contributes positively or negatively to the trajectory outcome (Wang et al., 2025a).

To alleviate this issue, researchers present turn-level process RM to perform credit assignment within trajectory, which can be roughly divided into two types based on the requirement of process reward labeling: (1) One employs finite MCTS (Guo et al., 2025; Wang et al., 2024; Xi et al., 2025) or GPT-4 (Xie et al., 2025) to annotate pseudo process rewards. However, they face the difficulty of cost-effectively obtaining accurate process labels, yielding the noisy turn-level supervision. (2) The others are independent of process reward labels, instead leveraging (Kim et al., 2023; Wang et al., 2025a) the trajectory outcome to indirectly monitor the credit assignment. But these methods completely neglect the process information (e.g., action importance), resulting in unfocused credit assignment (Verma and Metcalf, 2024). Moreover, both assign rewards at turn-level granularity, which is potentially overly fine-grained for sub-goals spanning multiple turns (Yin et al., 2025; Guo et al., 2025). All of these lead to unreliable credit assignment.

In this paper, we present segmental process rewards, which closely align the reward allocation with the sub-goal in the task, avoiding excessively granular allocation to turns. We also exploit the hindsight information to modulate rewards without extra process labels, incorporating valuable process insights to underscore significant segments and enhancing credit assignment reliability.

3 Methodology

In this section, we elaborate on exploiting hindsight information to modulate segmental process rewards, which closely aligns rewards with sub-goals and underscores significant segments to enhance the reliability of credit assignment. As illustrated in the left of Fig 2, we perform behavior cloning for LLM, equipping it with the basic task planning and reasoning capability, followed by trajectory collection. Then, as shown in the middle of Fig 2, the hindsight model and segment-level process RM characterized by progress estimation are constructed based on collected trajectories. Last, we achieve segmental importance scores based on the ratios of sequence likelihoods between the hindsight and policy model, modulating segmental process rewards for multi-turn RL algorithms (i.e., PPO), as depicted at the right of Fig 2.

3.1 Behavior Cloning and Trajectory Collection

To achieve basic capabilities of task planning and reasoning, the LLM is required to perform behavior cloning via supervised fine-tuning on the successful expert trajectories dataset $D_{bc} = \{Y_i\}_{i=1}^{|D_{bc}|}$, as shown in the left of Fig 2. Particularly, let $Y_i = \{o_{i,1}, a_{i,1}, ..., o_{i_k}, a_{i,k}, ...o_{i_m}, a_{i,m}\}$ denote the i-th trajectory with m turns in D_{bc}, where $o_{i,k}$ and $a_{i,k}$ denote the observation returned by environment and response generated by LLM in the k-th turn. It should be noted that each response $a_{i,k}$ is a thought-action pair, which adheres to ReAct (Yao et al., 2022b) style, namely executing CoT reasoning prior to determining each action for stimulating their synergistic effect. Then, we optimize LLM by minimizing the negative log-likelihood of each trajectory Y_i as below (1):

$$\mathcal{L}_{\text{bc}}(Y_i) = -\sum_{k=1}^{m}\sum_{j=1}^{|a_{i,k}|} log(a_{i,k}^{j}|o_{i,<=k}, a_{i,<k}, a_{i,k}^{<j}). \quad (1)$$

Note that we only compute the loss of thought-action pairs $a_{i,k}$, excluding the observations $o_{i,k}$ to improve the training stability (Jin et al., 2025). By doing so, the LLM acquires the fundamental capability to accomplish the agentic tasks, resulting in a reference policy model π_{ref}, which considerably accelerates the optimization of downstream RL.

Considering that effectively evaluating the impact of a certain action on trajectory outcome requires observing this action in a wide range of

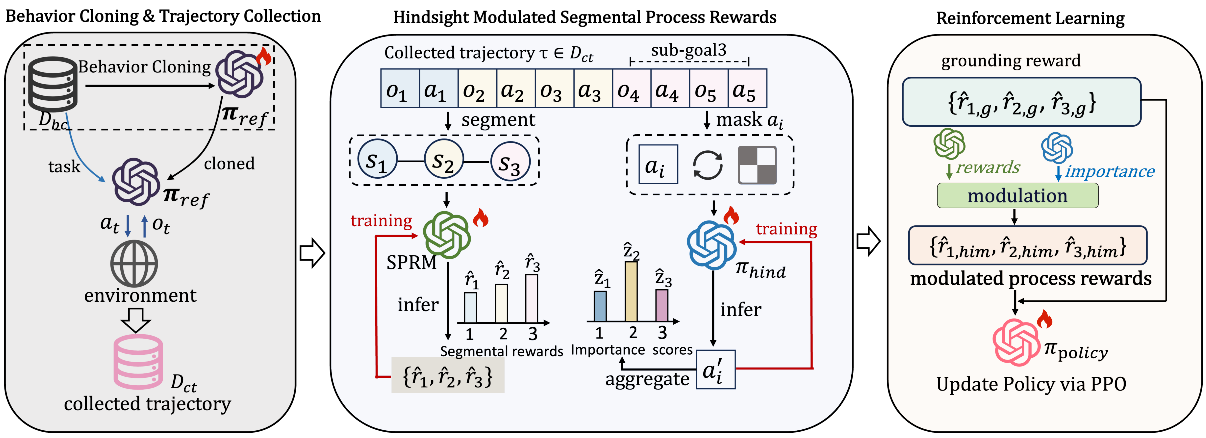

Figure 2: An illustration of our method. The left showcases the details of behavior cloning and trajectory collection. The middle illustrates the training and inference of the segment-level process reward and hindsight model. The right describes the construction of hindsight information modulated rewards and employs them for policy optimization.

interactive histories, we have the reference policy model π_{ref} to explore in the environment, thereby collecting trajectories. Concretely, we follow the line of Wang et al. (2025a), which conducts N roll-outs for each task in D_{bc} without any demonstrations and manually designed exploration schemes (Hao et al., 2023; Zhao et al., 2023) to ensure trajectory diversity. The samples with failed instruction following or repetition are further filtered to achieve the collected trajectory dataset $D_{ct} = \{\{(\tau_{i,j}, R_{i,j})\}_{j=1}^{N}\}_{i=1}^{|D_{ct}|}$, where $\tau_{i,j}$ and $R_{i,j}$ denote the j-th trajectory for the i-th task and its corresponding trajectory outcome (i.e., a scalar number to reflect the task completion).

3.2 Hindsight Information Modulated Segmental Process Rewards

To prevent potentially overly fine-grained process rewards caused by mere reward allocation at turn-level granularity and unfocused process rewards caused by the complete neglect of process information, we propose hindsight information modulated segmental process rewards centered on aligning rewards with sub-goals and underscoring significant segments to enhance credit assignment reliability, as shown in the middle and right of Fig 2.

Segmental Process Reward Model (SPRM). The SPRM is presented to perform credit assignment within the trajectory at the segment-level granularity. Here, a segment refers to a turn or a contiguous set of turns in the trajectory, typically corresponding to a sub-goal in the decomposition of the given task. To obtain segments contained in each trajectory τ in D_{ct}, we exploit GPT-4o (Hurst et al., 2024) to identify sub-goals in the task, and allocate turns to the corresponding sub-goal, achieving the segmented trajectory dataset $D_{st} = \{\{(\tau_{i,j}^{s}, R_{i,j})\}_{j=1}^{N}\}_{i=1}^{|D_{st}|}$, where $\tau^{s} = \{s_1, s_2, ..., s_n\}$ denote the trajectory with n segments. With D_{st}, we borrow the idea from progress estimation (Wang et al., 2025a) and further perform credit assignment at the segment-level granularity. For each segmented trajectory τ^{s}, SPRM relies on the reasoning capability of the reference policy model π_{ref} and intuitively decomposes the scalar trajectory outcome R into segmental contributions. It reflects how each segment incrementally propels the agent towards trajectory outcome. Concretely, a lightweight Multi-Layer Perceptron (MLP) is appended to the last hidden layer of π_{ref}, allowing it to output a scalar contribution score r_i at the end of each segment as (2). Furthermore, for a trajectory τ^{s} with n segments, we optimize the SPRM by minimizing the mean square error loss as follows (3):

$$r_i = W_2(\text{SiLU}(W_1 h_i)), \tag{2}$$

$$\mathcal{L}_{\text{sprm}}(\tau^{s}) = (R - \sum_{i=1}^{n} r_i)^2, \tag{3}$$

where h_i, SiLU, W_1, W_2, denote the hidden state of the last token in i-th segment s_i, the activation function and learnable weights in MLP.

Hindsight Information Modulation. With SPRM, we realize credit assignment at the segment-level granularity, which closely aligns rewards with sub-goals in the task. To further conquer the issue of unfocused process rewards caused by the complete neglect of process information and effectively incorporate valuable process insights, we introduce hindsight information, which refers to the importance of a certain action towards achieving the trajectory outcome in hindsight. We leverage the hindsight information to modulate segmental process rewards, making the credit assignment underscoring the significant segments within the trajectory.

To model the hindsight information, the reference policy model π_{ref} undergoes continued training with an objective akin to masked language modeling, achieving a hindsight model π_{hind}. Particularly, for a collected trajectory $\tau = \{o_1, a_1, ..., o_k, a_k, ...o_m, a_m\}$ with m turns, we mask each response a_k in the turn with an underline and demand π_{ref} to recover it as (4), capturing likelihoods which reflects the preference of performing a certain action after knowing the outcome of the trajectory.

$$\mathcal{L}_{\text{hind}}(a_k) = -\sum_{j=1}^{|a_k|} log(a_k^j | o, a_{<k}, a_{>k}, a_k^{<j}), \quad (4)$$

with this characteristic, we draw inspiration from Harutyunyan et al. (2019) and design the ratios of sequence likelihoods between the hindsight π_{hind} and policy model π_{policy} to measure the importance $z(a_k)$ of k-th action, as formalized in (5), (6).

$$r(a_k^j) = \frac{\pi_{hind}(a_k^j | o, a_{<k}, a_{>k}, a_k^{<j})}{\pi_{policy}(a_k^j | o_{<=k}, a_{<k}, a_k^{<j})}, \quad (5)$$

$$z(a_k) = exp(\frac{1}{\beta |a_k|} \sum_{j=1}^{|a_k|} log(r(a_k^j))), \quad (6)$$

where a_k^j denote j-th token in the k-th action. β is the hyper-parameter to control the distinction between different actions. Note that π_{policy} computes likelihoods of the current action a_k with only information available prior to a_k, while π_{hind} additionally incorporates the subsequent information available in hindsight. Intuitively, if $z(a_k) > 1$, it means the agent prefers to execute the current action a_k in hindsight, indicating that this action propels the agent towards the trajectory outcome, highlighting its importance and vice versa. Consequently, for a trajectory τ with m turns, we can obtain m turn-level action importance ratios $z = \{z(a_1), z(a_2), ..., z(a_m)\}$. Similarly, to align importance measurement with the sub-goal in the task, we aggregate segmental importance scores $z_s = \{z(s_1), z(s_2), ...z(s_n)\}$ by adding up turn ratios belonging to the same segment, highlighting the significant segments in the trajectory.

To coordinate segmental process rewards and importance scores, for a trajectory τ with n segments, we leverage trained SPRM to predict segmental progress rewards $\hat{R} = \{\hat{r}_1, \hat{r}_2, ..., \hat{r}_n\}$. Then π_{hind} and π_{policy} are exploited to compute segmental importance scores $\hat{z}_s = \{\hat{z}_{s_1}, \hat{z}_{s_2}, ..., \hat{z}_{s_n}\}$. We perform multiplication and normalization on $\hat{R}$ and $\hat{z}_s$ to scale rewards based on importance scores, achieving hindsight information modulated segmental process rewards $\hat{R}_{him} = \{\hat{r}_1^{him}, \hat{r}_2^{him}, ..., \hat{r}_n^{him}\}$. By doing so, the valuable process insights (i.e., action importance) are incorporated, and significant segments would be underscored, improving the credit assignment reliability.

$$\hat{R}_{him} = \frac{\hat{R} \odot \hat{z}_s}{||\hat{R} \odot \hat{z}_s||}. \quad (7)$$

3.3 Multi-turn Agentic RL

Modulated by hindsight information, $\hat{R}_{him}$ accurately reflects how each segment propels the agent towards the trajectory outcome. But it fails to indicate the executability of each action in the environment. To bridge this gap, we additionally introduce an action grounding reward $\hat{r}^g$ that takes 1 if the action is executable, else 0, achieving $\hat{r}^{fuse}$:

$$\hat{r}^{fuse} = (1-\alpha)\hat{r}^{him} + \alpha \hat{r}^g, \quad (8)$$

where α is the hyper-parameter to balance two rewards. Later, $\hat{r}_{fuse}$ is used to guide the optimization of policy model π_{policy} under PPO algorithm:

$$\mathcal{L}_{\text{clip}}(\theta) = \mathbb{E}_t \Big[\min(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} \hat{A}_t^{fuse}, \text{clip}(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon)\hat{A}_t^{fuse} \Big], \quad (9)$$

where π_θ is the policy with parameters θ. $\hat{A}_t^{fuse}$ is the advantage, which is adopted as generalized advantage estimation (GAE). It blends temporal-difference errors $\delta_t = \hat{r}_t^{fuse} + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$. V_ϕ is an alternatively trained value network.

4 Experiments

In this section, we introduce datasets, baselines, and experimental settings. Then, we present experimental results and provide a detailed analysis.

4.1 Datasets and Experimental Settings

To verify the effectiveness of our method, we carry out experiments on three publickly available agentic benchmarks: Alfworld (Shridhar et al., 2020), Virtualhome (Puig et al., 2018) for embodied household suits, and Webshop (Yao et al., 2022a) for web navigation. Particularly, for Alfworld, we follow previous works (Wang et al., 2025a,b; Feng et al., 2025) to adopt the version constructed by Song et al. (2024). For Virtualhome, we employ the filtered version provided by Wang et al. (2025b). On all benchmarks, the agent receives an environmental observation each turn and accordingly determines an action until completing the task or until reaching maximum interactive turns, ultimately achieving a scalar score as the trajectory outcome. Further details about datasets and specific experimental settings can be found in Appendix A and B.

4.2 Baselines

We compare our method with the following three types of baselines. (1) Prompt engineering evaluates the frozen LLMs (**Llama3.2** (Dubey et al., 2024), **GPT4o** (Hurst et al., 2024), and **Gemeni2.5pro** (Team et al., 2023)) under zero-shot setting. (2) Behavior cloning includes the methods of supervised fine-tuning (**SFT** (Ouyang et al., 2022)), reject sampling fine-tuning (**RFT** (Tang et al., 2024)), and direct preference optimization (**DPO** (Rafailov et al., 2023)). We serve them as baselines to evaluate the performance gain from offline samples. (3) Reinforcement Learning based Fine-tuning is our primary baseline. The methods of **PPO** (Schulman et al., 2017), **GRPO** (Shao et al., 2024), **Archer** (Zhou et al., 2024) with outcome-based rewards are selected to compare with the following methods characterized by process rewards. **StepAgent** (Deng et al., 2024) leverages the expert policy to score each turn, achieving the turn-level supervision. **RAGEN** (Wang et al., 2025c) designs a bi-level GAE, providing fine-grained rewards in different turns. **PRM4A** (Choudhury, 2025) and **SPA** (Wang et al., 2025a) are the strongest baselines with turn-level process rewards, which leverage trajectory outcome to indirectly monitor the credit assignment. More details can be found in Appendix C.

4.3 Experimental Results

The experimental results on three benchmarks are reported in Table 1. We can observe that: (1) Even the best-performing model (**Gemini2.5pro**) exhibits far from satisfactory with the method of prompt engineering, revealing the deficiency of LLM in solving complex long-horizon decision-making agentic tasks. (2) When fine-tuned on the expert trajectory dataset D_{bc}, the performance (**SFT**) substantially surges across benchmarks, demonstrating the effectiveness of equipping LLM with basic task planning and reasoning capabilities. However, further employing more high-quality trajectories in **RFT** or pairwise preference data in **DPO** can not break the performance bottleneck. We believe this is because the offline data is insufficient to represent the complex decision-making of the LLM-based agent in the environment.

(3) Compared to **SFT**, the RL methods of **PPO** and **GRPO** with outcome-based reward only exhibit a slight improvement, even suffering from degradation in some cases (e.g., Virtualhome dataset). This can be ascribed to the characteristic of long-horizon in the agentic task and sparse delayed reward in the algorithm, which makes it difficult to propagate the final reward to the actions that happened in the early turns. (4) When equipped with dense process rewards, this deficiency is partially alleviated, and performance gains are obtained (e.g., **Archer**, **StepAgent**, **RAGEN**). This is consistent with the observation in Wang et al. (2025c), where fine-grained reward plays an important role in multi-turn agentic RL. (5) Compared to the strongest baseline **SPA**, our method outperforms on all benchmarks. This can be attributed to two factors. One is closely aligning credit assignment with sub-goals in the task, avoiding excessively granular allocation to turns. Another is incorporating valuable process insights (i.e., action importance) to prevent unfocused rewards, where we introduce hindsight information to modulate segmental process rewards, underscoring significant segments within the trajectory. In the end, the considerable performance gains across benchmarks demonstrate the validity of our method.

4.4 Ablation Study

To explicitly illustrate the efficacy of our method, we conduct ablation studies to validate its core design on all benchmarks. Specifically, as shown in the last part of Table 1, we define the following ablation variants: **(1) -w/o HIM** removes hindsight information modulation, retaining only segmental process rewards to align the credit assignment with sub-goals in the task. **(2) -w/o SPR** removes segmental process rewards, retaining only the hindsight information modulation to underscore the significant turns in the trajectory. **(3) -w/o BOTH** is the combination of **(1)** and **(2)**. **(4) -w/o AGR** removes the action grounding reward, keeping other

Methods	Type	Alfworld ↑							Virtualhome ↑	Webshop ↑
		PICK ↑	CLEAN ↑	HEAT ↑	COOL ↑	LOOK ↑	PICK2 ↑	*Avg* ↑		
Llama3.2 (Dubey et al., 2024)	PE	12.5	0.0	0.0	0.0	0.0	0.0	*2.3*	1.2	19.8
GPT4o (Hurst et al., 2024)	PE	75.3	60.8	31.2	56.7	21.6	49.8	*48.0*	20.8	23.7
Gemini2.5pro (Team et al., 2023)	PE	92.8	63.3	62.1	69.0	26.6	58.7	*60.3*	31.7	35.9
SFT (Ouyang et al., 2022)	BC	79.2	77.4	73.9	61.9	83.3	58.8	*73.1*	51.8	62.0
RFT (Tang et al., 2024)	BC	83.3	83.9	78.3	66.7	77.8	47.1	*74.6*	53.1	62.9
DPO (Rafailov et al., 2023)	BC	87.5	83.9	60.9	66.7	88.9	64.7	*76.1*	52.8	62.6
PPO (Schulman et al., 2017)	RL	83.3	**87.1**	73.9	61.9	77.8	47.1	*73.9*	51.0	62.1
GRPO (Shao et al., 2024)	RL	87.5	70.9	82.6	57.1	88.9	52.9	*73.4*	51.2	61.8
Archer (Zhou et al., 2024)	RL	83.3	83.9	65.2	**85.7**	66.7	58.8	*75.4*	–	–
StepAgent (Deng et al., 2024)	RL	83.3	**87.1**	78.3	71.4	77.8	41.2	*75.4*	–	–
RAGEN (Wang et al., 2025c)	RL	91.7	77.4	78.3	57.1	88.9	52.9	*75.4*	52.1	63.0
PRM4A (Choudhury, 2025)	RL	58.3	80.6	73.9	71.4	**100.0**	58.8	*73.9*	–	–
SPA (Wang et al., 2025a)	RL	**95.8**	83.9	**87.0**	61.9	77.8	58.8	*79.1*	53.4	64.1
HISR	RL	83.3	**87.1**	65.2	**85.7**	**100.0**	**82.4**	***83.6***	**59.1**	**69.1**
-w/o HIM	RL	87.5	83.9	78.3	66.7	94.4	70.6	*80.6*	55.1	63.7
-w/o SPR	RL	87.5	83.9	78.3	76.2	94.4	70.6	*82.1*	57.9	69.1
-w/o BOTH	RL	87.5	83.9	69.6	71.4	94.4	58.8	*78.4*	53.4	63.7
-w/o AGS	RL	91.7	87.1	78.3	66.7	100.0	64.7	*82.8*	57.5	68.6

Table 1: Evaluation results and ablation study on three benchmarks. *Avg* indicates the comprehensive score in the Alfworld dataset. PE, BC, and RL indicate methods of prompt engineering, behavior cloning, and reinforcement learning. Best in bold, the runner-up with an underline. The top three parts denote baseline methods, while the last part denotes our method and its ablation variants. $*$ denotes it adopts turn-level process rewards.

core designs unchanged. Note that our method is equal to the variant -w/o SPR in the Webshop dataset. This is because GPT-4o deems that 89.3% samples do not require segmentation as shown in Appendix B. It demonstrates that most turns have already been aligned with the sub-goals in the task. Thus, we adopt the turn-level process reward.

Concretely, we can draw inferences according to the results of Table 1. (1) Removing hindsight information modulation (-w/o HIM) results in a performance drop, reflecting its crucial role in incorporating valuable insights to underscore significant segments in the trajectory. (2) Replacing segmental process rewards also leads to a performance degradation, underscoring its vital role in closely aligning rewards with sub-goals to prevent excessively granular allocation to turns. (3) Removing both hindsight information and segmental process rewards (-w/o BOTH) leads to the worst performance, indicating the necessity of each core design. (4) Removing action grounding reward (-w/o AGS) impairs performance, demonstrating the validity of making rewards represent action executability.

4.5 Analysis of SPR and HIM

To intuitively understand each core design, we conduct two statistical analyses on the Alfworld dataset. Specifically, as shown in Fig 3, we compute the number of turns in all segments and observe that GPT-4o merged most of the turns in the trajectories (segments containing two or more turns

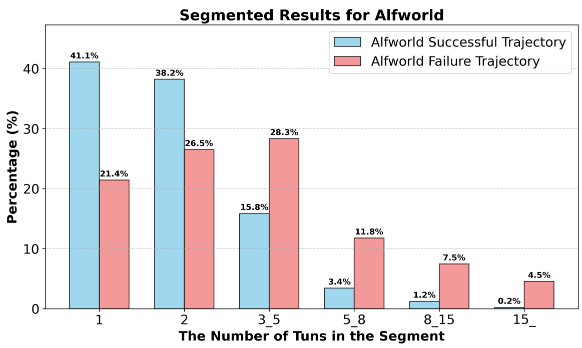

Figure 3: Statistical results of the number of turns in the segment of the Alfworld dataset.

accounted for over 50%), which demonstrates that the phenomenon of a sub-goal spanning on multiple turns is common, reflecting the feasibility of segmental process rewards. This phenomenon is more highlighted in the failure trajectory. We further scrutinize the trajectory and find that GPT-4o combines multiple trial-and-error turns aimed at achieving the same sub-goal into a single segment. Additionally, we can observe in Fig 4 that segment-level process RM inclines to allocate more rewards to the latter segment (i.e., 66.3%). This is probably because later segments are closer to the task completion without consideration of process information. However, this prior underestimates the importance of earlier segments towards propelling the trajectory outcome, resulting in unfocused credit assignment. Instead, the important scores modu-

Segment (turn)	Conversations	$(\hat{\mathbf{R}}, \hat{\mathbf{z}}, \hat{\mathbf{R}}_{\mathbf{him}})$
	Case: Clean and put bowl → Fridge	
Segment1 (turn1) Task interpretation and decomposition	**Task Define**: You are in the middle of a room... **Your task is to:** clean some bowl and put it in fridge. **Thought**: The task is to clean a bowl and put it in the fridge. I should first locate the fridge to find a bowl that needs cleaning. **Action**: go to fridge 1.	$(0.069,\ 0.127,\ 0.039)$
Segment2 (turn2-3) Opening the fridge and picking up the bowl	**Observation**: The fridge 1 is closed. **Thought**: I'm in front of the fridge, but it's closed. I need to open it to find a bowl. **Action**: open fridge 1. **Observation**: You open the fridge 1. The fridge 1 is open. In it, you see a apple 3, a apple 2, a apple 1, a bowl 1, a bread 1, and a cup 1. **Thought**: The fridge is now open, and I see a bowl inside. I should take the bowl out to clean it. **Action**: take bowl 1 from fridge 1.	$(0.118,\ \mathbf{0.392},\ \mathbf{0.205} \uparrow)$
Segment3 (turn4-5) Going to the sink and cleaning the bowl	**Observation**: You pick up the bowl 1 from the fridge 1. **Thought**: With the bowl in hand, I need to find a place to clean it. The sink is the best place for this task. **Action**: go to sinkbasin 1. **Observation**: On the sinkbasin 1, you see nothing. **Thought**: I'm at the sink and it's clear, which is perfect for cleaning the bowl without any obstructions. **Action**: clean bowl 1 with sinkbasin 1.	$(0.132,\ 0.286,\ 0.167)$
Segment4 (turn6-7) Returning to the fridge and placing the bowl	**Observation**: You clean the bowl 1 using the sinkbasin 1. **Thought**: The bowl is now clean, and I need to put it back in the fridge as per the task instructions. I should head back to the fridge. **Action**: go to fridge 1. **Observation**: The fridge 1 is open. In it, you see a apple 3, a apple 2, a apple 1, a bread 1, and a cup 1. **Thought**: I'm back at the fridge, which is still open from earlier. I can now place the clean bowl inside. **Action**: put bowl 1 in/on fridge 1.	$(0.681,\ 0.195,\ 0.589)$

Table 2: Case study on the Alfworld dataset. The left shows segmented results of the trajectory, where Segment (turn) denotes consecutive turns contained in the segment. The middle shows the explored trajectory, where blue texts highlight the highest importance scores segment. The right shows the tuple of raw segmental process rewards, hindsight information modulated rewards, and segmental importance ratio scores.

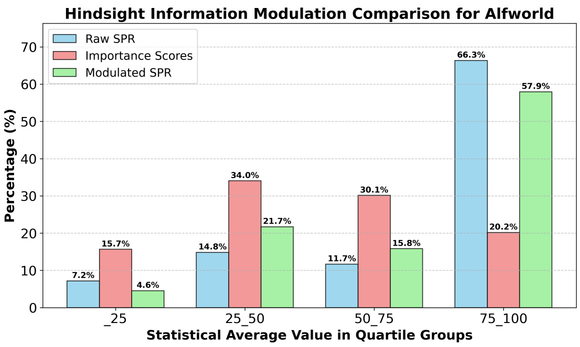

Figure 4: Statistical results of the process rewards distribution in the Alfworld datasets. SPR denote the segmental process rewards.

late SPR, facilitating underscoring earlier significant segments (i.e., 14.8% to 21.7% and 11.7% to 15.8%) in the trajectory.

4.6 Case Study

We conduct the case study on the Alfworld dataset to illustrate our core designs, as shown in Table 2. Concretely, we observe that GPT-4o tends to merge consecutive turns so that combined segments have more complete and clear sub-goals. For instance, in segment2, the goal of opening the fridge in turn2 is picking up the bowl in turn3, which is planned in segment1 ***"I should first locate the fridge to find a bowl ..."*** in advance. Compared to turn-level granularity, the semantics within each segment are more coherent, and the semantic distinctions between adjacent segments are more pronounced (corresponding to different sub-goals), which facilitates the reward model identifying the contribution of each segment towards the trajectory outcome. Moreover, turn-level rewards underestimate the crucial role of picking up the bowl in segment2 and cleaning the bowl in segment3 for completing the task, resulting in unfocused process rewards. Instead, our method leverages the importance scores $\hat{z}$ (i.e., 0.392 and 0.286) to modulate the segmental process rewards $\hat{R}$, underscoring significant segments (e.g., 0.118 to 0.205 and 0.132 to 0.167) and achieving more reasonable rewards $\hat{R}_{him}$ for RL training. More cases are shown at Table 4-5 in Appendix.

5 Conclusions

In this paper, we focus on improving LLMs' agentic capability by overcoming the issues of delayed propagation in sparse outcome rewards and unreliable credit assignment with potentially overly fine-grained and unfocused turn-level process rewards. Specifically, we propose (**HISR**) exploiting **H**indsight **I**nformation to modulate **S**egmental process **R**ewards for multi-turn RL, which closely aligns rewards with sub-goals and underscores significant segments to enhance the reliability of credit assignment. Extensive experimental results on three available agentic benchmarks and case study demonstrate the effectiveness of our method.

Limitations

Despite the impressive results of our method, we have to admit our work has the following limitations: (1) to align reward allocation with sub-goals in the task, we employ GPT-4o to split the trajectory into several segments, where each segment corresponds to a sub-goal for the task. Note that we do not utilize the internal knowledge of GPT-4o to allocate rewards and improve performance, merely segmenting the trajectory. However, it still introduces an additional step in the pipeline of RL. To overcome this issue, we plan to explore automatic segmentation of trajectory based on the change in entropy during the process of rollout. (2) to address the issue of unfocused credit assignment caused by the complete neglect of process information and effectively incorporate valuable process insights, we introduce the hindsight information (i.e., segment importance scores) to underscore significant segments and achieve more reasonable segmental process rewards. However, the hindsight model is trained in advance and keeps frozen in the subsequent stage of RL training, which may lead to limited adaptation to changes in the data distribution, impairing the performance. We believe training hindsight model online can further bootstrap performance. (3) Although the importance scores achieved by the hindsight model underscore the significant segments, the prior on raw segmental process rewards is too strong (e.g., assigning high rewards to the latter segments). This causes the modulation effect from hindsight information to not be utilized to its fullest potential. In the future, we will explore the online hindsight model, mining more hindsight information, and directly employ the hindsight information to guide the training of process reward model.

References

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

Hanbo Bi, Zhiqiang Yuan, Zexi Jia, Jiapei Zhang, Chongyang Li, Peixiang Luo, Ying Deng, Xiaoyue Duan, and Jinchao Zhang. 2025. F2rvlm: Boosting fine-grained fragment retrieval for multi-modal long-form dialogue with vision language model. *arXiv preprint arXiv:2508.17714*.

Sanjiban Choudhury. 2025. Process reward models for llm agents: Practical framework and directions. *arXiv preprint arXiv:2502.10325*.

Zhirui Deng, Zhicheng Dou, Yutao Zhu, Ji-Rong Wen, Ruibin Xiong, Mang Wang, and Weipeng Chen. 2024. From novice to expert: Llm agent policy optimization via step-wise reinforcement learning. *arXiv preprint arXiv:2411.03817*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. 2025. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *arXiv preprint arXiv:2505.23564*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore. Association for Computational Linguistics.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, and 1 others. 2019. Hindsight credit assignment. *Advances in neural information processing systems*, 32.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Changyeon Kim, Jongjin Park, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. 2023. Preference transformer: Modeling human preferences using transformers for rl. *arXiv preprint arXiv:2303.00957*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Nayu Liu, Junnan Zhu, Yiming Ma, Zhicong Lu, Wenlei Xu, Yong Yang, Jiang Zhong, and Kaiwen Wei. 2025. SARA: Salience-aware reinforced adaptive decoding for large language models in abstractive summarization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25450–25463, Vienna, Austria. Association for Computational Linguistics.

Zhicong Lu, Li Jin, Peiguang Li, Yu Tian, Linhao Zhang, Sirui Wang, Guangluan Xu, Changyuan Tian, and Xunliang Cai. 2024. Rethinking the reversal curse of LLMs: a prescription from human knowledge reversal. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7518–7530, Miami, Florida, USA. Association for Computational Linguistics.

Zhicong Lu, Changyuan Tian, Peiguang Li, Li Jin, Sirui Wang, Wei Jia, Ying Shen, and Guangluan Xu. 2025. PIPER: Benchmarking and prompting event reasoning boundary of LLMs via debiasing-distillation enhanced tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28591–28613, Vienna, Austria. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

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, Bangkok, Thailand. Association for Computational Linguistics.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: Scaling instruction tuning for mathematical reasoning. *arXiv preprint arXiv:2403.02884*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Changyuan Tian, Zhicong Lu, Shuang Qian, Nayu Liu, Peiguang Li, Li Jin, Leiyi Hu, Zhizhao Zeng, Sirui Wang, Ke Zeng, and 1 others. 2025. Rectify evaluation preference: Improving llms' critique on math reasoning via perplexity-aware reinforcement learning. *arXiv preprint arXiv:2511.10303*.

Mudit Verma and Katherine Metcalf. 2024. Hindsight priors for reward learning from human preferences. *arXiv preprint arXiv:2404.08828*.

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, Vienna, Austria. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, and 1 others. 2025c. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, Wenxiang Chen, Zhihao Zhang, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, and 1 others. 2025. Agentprm: Process reward models for llm agents via step-wise promise and progress. *arXiv preprint arXiv:2511.08325*.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. 2025. Capo: Towards enhancing llm reasoning through verifiable generative credit assignment. *arXiv e-prints*, pages arXiv–2508.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022a. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

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

Yueqin Yin, Shentao Yang, Yujia Xie, Ziyi Yang, Yuting Sun, Hany Awadalla, Weizhu Chen, and Mingyuan Zhou. 2025. Segmenting text and learning their rewards for improved rlhf in language model. *arXiv preprint arXiv:2501.02790*.

Yuanqing Yu, Zhefan Wang, Weizhi Ma, Zhicheng Guo, Jingtao Zhan, Shuai Wang, Chuhan Wu, Zhiqiang Guo, and Min Zhang. 2024. Steptool: A step-grained reinforcement learning framework for tool learning in llms.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*.

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.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*.

A Details on Datasets

Alfworld (Shridhar et al., 2020) is an embodied environment designed to assess the agentic capability of LLMs to solve complex long-horizon decision-making tasks. It provides interactive TextWorld environments, which are meticulously aligned with the ALFRED (Shridhar et al., 2020). In each episode, the agent receives the text goal and is required to accomplish it through multi-turn interaction with the environment until completing the task or until reaching the maximum interactive turns, ultimately achieving a scalar score as the outcome of the trajectory. Here, the scalar score of '1' and '0' denote success and failure, respectively. Alfworld includes six categories of common household activities: Pick & Place (Pick), Examine in Light (Look), Clean & Place (Clean), Heat & Place (Heat), Cool & Place (Cool), and Pick Two & Place (Pick2). In this work, we adopt the version constructed by Song et al. (2024) and set the maximum interactive turns as 40.

Virtualhome (Puig et al., 2018) is also an embodied environment designed to assess the agentic capability of LLMs akin to the Alfworld dataset. It encompasses 292 high-level household tasks and 1,374 unique action plans across 6,201 diverse virtual environments. For each task, the agent receives a high-level task, a descriptive explanation, and a set of executable actions. Then, the agent continuously selects an executable action and receives feedback from the environment until the task is completed or the maximum number of interaction turns is exceeded, ultimately taking '1' if the task is succeeded, else '0'. In this work, we adopt the filtered version provided by Wang et al. (2025a) and set the maximum interactive turns as 40.

Webshop (Yao et al., 2022a) is a complex, web-based interactive environment designed to test the LLM agents in realistic online shopping scenarios

(Guo et al., 2025). In the environment, the agent is required to search for, navigate to, and ultimately execute a "buy" action when a purchase decision is made. Then, the agent receives a float number between '0' and '1', which is determined by the alignment of the selected product's attributes and price within the given instructions. In this work, we set the maximum interactive turn as 10.

To the end, we ensure that our dataset setting is consistent with the strongest baseline **SPA** (Wang et al., 2025a) for a fair comparison. The specific statistical results are shown in the Table 3 as below. Note that the webshop dataset does not have fixed available actions. However, it can only choose the action types of search and click.

Benchmark	Train	Test	Available Actions	Maximum Turns
ALFWorld	2,851	134	13	40
VirtualHome	4,920	247	40	40
Webshop	1938	200	-	10

Table 3: The statistical results of three agentic benchmarks. Train and Test denote the number of expert trajectory sample and test sample, respectively.

B Experimental Settings

To make a fair comparison with other baselines, we follow them to select Llama3.2-3B-Instruct (Dubey et al., 2024) as the backbone. During behavior cloning, the backbone is trained for three epochs with the learning rate and batchsize of $(1.5e-5, 32), (4e-5, 24), (1.5e-5, 32)$ for Alfworld, Virtualhome, and Webshop datasets. Consuqentaly, we obtain the reference policy model π_{ref} (1). With π_{ref}, we conduct $N = 10$ rollouts for each task to obtain the collected trajectory dataset D_{ct} with a temperature of 0.7. Then, we filter trajectories with wrong format and repetition, obtaining 26,433, 49,199, and 14,344 samples for Alfworld, Virtualhome, and Webshop datasets.

With D_{ct}, we exploit GPT4o to split each trajectory into a set of segments, achieving D_{st}. During segmentration, GPT4o believes that 89.3% of the trajectories in Webshop do no need to segment, while Alfworld and Virtualhome are 11.5% and 16.9%. This is because the interactive turns of Webshop are shortest so that most turns have already been aligned with the sub-goals in the task. Then, we continue training the π_{policy} with D_{st} for 1 epoch with the learning rate and batchsize of $(2e-6, 16), (2e-6, 16), (2e-6, 48)$ for Alfworld, Virtualhome, and Webshop datasets (3). Later, we obtain the segmental process reward model. We also continue training the π_{policy} with D_{st} for 1 epoch with the learning rate and batchsize of $(1e-6, 192), (1e-6, 192), (1e-6, 24)$ for Alfworld, Virtualhome, and Webshop datasets (4). The hindsight model π_{hind} is then achieved. The β and α are intuitively set to 0.3 when inferring the hindsight information (6) and introducing the action grounding reward (8). During RL, we adopt the Low-Rank Adaptation (Lora (Hu et al., 2022)) with $rank = 8$. The coefficient of kl divergence, γ, and λ in PPO algorithm are set $1e-2$, 0.99, and 0.95, respectively. The learning rates for critic and policy model are set to $1e-4$ and $1e-5$, respectively. During inference, we set the temperature to 0 for deterministic generation. Consistent with previous works (Wang et al., 2025a; Feng et al., 2025), we set success rate as evaluation metric. All experiments are conducted on 8 H20 GPUs. To accelerate inference, we adopt the library of vllm (Zheng et al., 2023) and fastchat (Kwon et al., 2023). The prompt for behavior cloning, trajectory segmentation, and hindsight model training are shown in Table 9-12.

C Baselines

Prompt Engineering Baselines: To evaluate the agentic capabilities of powerful LLMs, the frozen LLMs (i.e., Llama3.2, GPT4o, Gemeni2.5pro) are adopted to execute the task in the three benchmarks. The results for GPT4o and Gemini2.5pro are from Feng et al. (2025), while the remaining results are evaluated with the prompts shown in Table xxx.

Behavior Cloning Baselines: To evaluate the performance gain from offline samples, the backbone is performing continual training with the high-quality expert trajectories or the pairwise trajectories. The results for SFT and RFT are from Wang et al. (2025a), while the DPO is reproduced with the open-sourced code Wang et al. (2025b).

Reinforcement Learning based Fine-tuning Baselines: To evaluate the performance gain from interacting with the environment in real time, the SFT backbone is performing multi-turn agentic reinforcement learning. For the methods with outcome-based reward (e.g., PPO, GRPO), we set the final scalar trajectory outcome as the final reward of the trajectory. For the methods with process reward (e.g., SPA), we ensure that the sum of process rewards is equal to the final scalar trajectory outcome. The results come from Wang et al. (2025a) and Feng et al. (2025).

Segment (turn)	Conversations	$(\hat{\mathbf{R}}, \hat{\mathbf{z}}, \hat{\mathbf{R}}_{\mathbf{him}})$
Alfworld Case: Put a clean apple → Microwave		
Segment1 (turn1) Initial task understanding and first action	**Task Define**: You are in the middle of a room. Looking quickly around you, you see a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 2, a countertop 1, a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1. **Your task is to:** put a clean apple in microwave. **Thought**: I need to put a clean apple in the microwave. First, I should locate an apple and ensure it's clean before I can use it. **Action**: go to garbagecan 1.	$(0.030, 0.220, 0.041)$
Segment2 (turn2-3) Acquiring the apple	**Observation**: On the garbagecan 1, you see a apple 2. **Thought**: I see an apple in the garbage can. I should pick it up and inspect it to see if it's clean or if it needs to be cleaned before use. **Action**: take apple 2 from garbagecan 1. **Observation**: You pick up the apple 2 from the garbagecan 1. **Thought**: The apple is not clean, so I should wash it to ensure it's suitable for eating or use in the microwave. **Action**: go to sinkbasin 1.	$(0.092, \mathbf{0.291}, \mathbf{0.168} \uparrow)$
Segment3 (turn4-5) Cleaning the apple	**Observation**: On the sinkbasin 1, you see nothing. **Thought**: I'm at the sink. Now I can clean the apple to make sure it's ready to be put in the microwave. **Action**: clean apple 2 with sinkbasin 1. **Observation**: You clean the apple 2 using the sinkbasin 1. **Thought**: The apple is now clean. I need to find the microwave and open it to place the apple inside. **Action**: go to microwave 1.	$(0.045, 0.240, 0.068)$
Segment4 (turn6) Preparing the microwave	**Observation**: The microwave 1 is closed. **Thought**: The microwave is closed. I'll need to open it before I can put the apple inside. **Action**: open microwave 1.	$(0.016, 0.110, 0.011)$
Segment5 (turn7) Placing the clean apple in the microwave	**Observation**: You open the microwave 1. The microwave 1 is open. In it, you see a apple 1, and a potato 2. **Thought**: The microwave is open and already contains an apple and a potato. I should place the clean apple inside the microwave, ensuring it's in a suitable spot where it won't get crushed by the potato. **Action**: put apple 2 in/on microwave 1.	$(0.818, 0.139, 0.712)$
Virtualhome Case: Wash clothes		
Segment1 (turn1) Task understanding and locating the basket for clothes	**Task Define**: The task is Wash clothes (I grab basket for clothes. I walk into the kitchen. I put clothes in washing machine. I turn on washing machine.). **Thought**: To successfully complete the task of washing clothes, I need to first locate the basket for clothes, as it is essential for gathering the laundry before proceeding to the washing machine. **Action**: Find basket for clothes.	$(0.415, 0.125, 0.275)$
Segment2 (turn2-3) Grabbing the basket and moving toward the washing machine	**Observation**: basket_for_clothes is CLOSED and CLEAN. **Thought**: I observed that the basket for clothes is closed, so my logical next step is to grab it in preparation to access the clothes that need washing. **Action**: Grab basket for clothes. **Observation**: You pick up the apple 2 from the garbagecan 1. **Thought**: The action "Walk to dining room" was likely chosen because the task involves moving to a different location to proceed with the washing process, possibly to find the washing machine or to transport the clothes inside it. **Action**: Walk to dining room.	$(0.209, \mathbf{0.299}, \mathbf{0.332} \uparrow)$
Segment3 (turn4-5) Reaching the washing machine and preparing it	**Observation**: basket_for_clothes is CLOSED and CLEAN. You have basket_for_clothes. **Thought**: The task requires me to wash clothes, but currently, I am holding a basket for clothes and standing in the dining room. I should locate the washing machine to proceed with the task. **Action**: Walk to washing machine. **Observation**: washing_machine is OFF and CLOSED and CLEAN and PLUGGED_IN, basket_for_clothes is CLOSED and CLEAN. You have basket_for_clothes. **Thought**: I have successfully reached the washing machine with the basket of clothes, and my next logical step is to open the washing machine so I can put the clothes in. **Action**: Open washing machine.	$(0.135, 0.227, 0.164)$
Segment4 (turn6) Placing the basket on the washing machine	**Observation**: basket_for_clothes is CLOSED and CLEAN, washing_machine is PLUGGED_IN and OPEN and OFF and CLEAN. You have basket_for_clothes. **Thought**: I will place the basket for clothes near the washing machine so that I can easily access the clothes to load them into the machine. **Action**: Put basket for clothes on washing machine.	$(0.113, 0.120, 0.073)$
Segment5 (turn7-8) Closing and turning on the washing machine	**Observation**: basket_for_clothes is CLOSED and CLEAN, washing_machine is OPEN and OFF and CLEAN and PLUGGED_IN, basket_for_clothes ON washing_machine. **Thought**: I chose to close the washing machine as the next logical step in the task of washing clothes after putting them inside, to prepare for starting the wash cycle. **Action**: Close washing machine. **Observation**: basket_for_clothes is CLOSED and CLEAN, washing_machine is CLOSED and PLUGGED_IN and OFF and CLEAN, basket_for_clothes ON washing_machine. **Thought**: Since I have already placed the basket for clothes on the washing machine and closed it, the next logical step is to turn on the washing machine to start the washing process. **Action**: Switch on washing machine.	$(0.128, 0.229, 0.156)$

Table 4: More case study on the Alfworld and Virtualhome dataset. The left shows segmented results of the trajectory, where Segment (turn) denotes consecutive turns contained in the segment. The middle shows the explored trajectory, where blue texts highlight the highest importance scores segment. The right shows the tuple of raw segmental process rewards, hindsight information modulated rewards, and segmental importance ratio scores.

Segment (turn)	Conversations	$(\hat{\mathbf{R}}, \hat{\mathbf{z}}, \hat{\mathbf{R}}_{\mathbf{him}})$
	Alfworld Case: Put a clean soapbar → Cabinet	
Segment1 (turn1) Understanding the task and initiating a search for the soapbar	**Task Define**: You are in the middle of a room. Looking quickly around you, you see a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a countertop 1, a garbagecan 1, a handtowelholder 1, a sinkbasin 2, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, and a towelholder 1. **Your task is to:** put a clean soapbar in cabinet. **Thought**: The task is to put a clean soapbar in a cabinet. I should first look for a soapbar in the immediate vicinity. **Action**: go to toilet 1.	$(0.027,\ 0.122,\ 0.022)$
Segment2 (turn2) Identifying and taking the soapbar	**Observation**: On the toilet 1, you see a soapbar 1, a spraybottle 1, and a toiletpaper 1. **Thought**: I've found a soapbar on the toilet. I need to pick it up to clean it before placing it in the cabinet. **Action**: take soapbar 1 from toilet 1.	$(0.065,\ 0.146,\ 0.064)$
Segment3 (turn3-4) Preparing and cleaning the soapbar	**Observation**: You pick up the soapbar 1 from the toilet 1. **Thought**: With the soapbar in hand, I should go to the sink to clean it since it's important to ensure cleanliness before storing or using it. **Action**: go to sinkbasin 1. **Observation**: On the sinkbasin 1, you see nothing. **Thought**: I'm at the sink. Now I should clean the soapbar to make sure it's clean before placing it in the cabinet. **Action**: clean soapbar 1 with sinkbasin 1.	$(0.063,$ **0.389**, **0.165** ↑)
Segment4 (turn5-6) Locating and accessing the cabinet	**Observation**: You clean the soapbar 1 using the sinkbasin 1. **Thought**: The soapbar is now clean. I need to find a cabinet to store the soapbar in. **Action**: go to cabinet 1. **Observation**: The cabinet 1 is closed. **Thought**: I've reached the cabinet. I need to open it to be able to put the soapbar inside. **Action**: open cabinet 1.	$(0.013,\ 0.214,\ 0.018)$
Segment5 (turn7) Storing the clean soapbar	**Observation**: You open the cabinet 1. The cabinet 1 is open. In it, you see a cloth 2. **Thought**: The cabinet is open and I see there's space inside. It's a suitable place to store the clean soapbar. **Action**: put soapbar 1 in/on cabinet 1.	$(0.832,\ 0.129,\ 0.730)$
	Alfworld Case: Cool and put some pot → Stoveburner	
Segment1 (turn1) Interpreting the task and begining exploration	**Task Define**: You are in the middle of a room. Looking quickly around you, you see a cabinet 20, a cabinet 19, a cabinet 18, a cabinet 17, a cabinet 16, a cabinet 15, a cabinet 14, a cabinet 13, a cabinet 12, a cabinet 11, a cabinet 10, a cabinet 9, a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 2, a countertop 1, a diningtable 2, a diningtable 1, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1. **Your task is to:** cool some pot and put it in stoveburner. **Thought**: I need to locate a pot to cool and then place it on a stoveburner. I'll start by checking the nearest appliances. **Action**: go to toaster 1.	$(0.039,\ 0.209,\ 0.051)$
Segment2 (turn2-3) Identifying the target object (pot)	**Observation**: On the toaster 1, you see nothing. **Thought**: The toaster is not where a pot would be, but I need to check all options systematically. **Action**: go to stoveburner 1. **Observation**: On the stoveburner 1, you see a pot 1. **Thought**: I've found a pot on stoveburner 1. I should pick it up so I can cool it as per the task. **Action**: take pot 1 from stoveburner 1.	$(0.063,\ 0.249,\ 0.097)$
Segment3 (turn4-6) Picking up and cooling the pot with the fridge	**Observation**: You pick up the pot 1 from the stoveburner 1. **Thought**: With the pot in hand, I need to find a way to cool it down. The fridge is typically used for cooling items, so I'll head there. **Action**: go to fridge 1. **Observation**: The fridge 1 is closed. **Thought**: The fridge is closed, but I can open it to cool the pot inside. **Action**: cool pot 1 with fridge 1. **Observation**: You cool the pot 1 using the fridge 1. **Thought**: Now that the pot is cooled, I need to place it back on a stoveburner. Stoveburner 1 is where I took it from, so it's the most convenient spot. **Action**: go to stoveburner 1.	$(0.058,$ **0.409**, **0.147** ↑)
Segment5 (turn7) Placing the cooled pot on the stoveburner.	**Observation**: On the stoveburner 1, you see nothing. **Thought**: I'm back at stoveburner 1, and it's clear. I can now place the cooled pot here to complete the task. **Action**: put pot 1 in/on stoveburner 1.	$(0.841,\ 0.134,\ 0.705)$

Table 5: More case study on the Alfworld dataset. The left shows segmented results of the trajectory, where Segment (turn) denotes consecutive turns contained in the segment. The middle shows the explored trajectory, where blue texts highlight the highest importance scores segment. The right shows the tuple of raw segmental process rewards, hindsight information modulated rewards, and segmental importance ratio scores.

Format	Prompt
Alfworld	
Human	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 a detailed description of the current environment and your goal to accomplish. For each of your turns, you will be given the observation of the last turn. You should first think about the current condition and plan for your future actions, and then determine your action in this turn. Your output must strictly follow this format:"Thought: your thoughts. Action: your next action". The admissible 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 returns "Nothing happened", that means the previous action is invalid, and you should try other reasonable actions. Your response should use the following format: Thought: <your thoughts> Action: <your next action> **Critical INSTRUCTIONS:** 1. Action Format: You must use the exact action formats provided above. NEVER simplify or modify them. 2. Grounding in History (previous actions and observations): When planning your future actions, you have to be strictly based on the history (previous actions and observations). Never assume anything not explicitly stated in the history. 3. Precise Task Decomposition: You should accurately understand and decompose the given task. Never overlook any details.
GPT	OK, ...

Table 6: Prompt template for behavior cloning and trajectory collection in the AlfWorld dataset.

Format	Prompt
VirtualHome	
Human	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 a detailed description of the current environment and your goal to accomplish. For each of your turns, you will be given the observation of the last turn. You should first think about the current condition and plan for your future actions, and then determine your action in this turn. Your output must strictly follow this format:"Thought: your thoughts. Action: your next action". The admissible actions are: walk to {obj} run to {obj} grab {obj} open {obj} close {obj} put {obj} on {recep} put {obj} in {recep} switch on {obj} switch off {obj} drink {obj} look at {obj} sit on {obj} stand up watch {obj} wipe {obj} type on {obj} wash {obj} cut {obj} eat {obj} sleep wake up plug in {obj} plug out {obj} pour {obj} into {recep} move {obj} release turn to {obj} 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 returns "Nothing happened", that means the previous action is invalid, and you should try other reasonable actions. Your response should use the following format: Thought: <your thoughts> Action: <your next action> **Critical INSTRUCTIONS:** 1. Action Format: You must use the exact action formats provided above. NEVER simplify or modify them. 2. Grounding in History (previous actions and observations): When planning your future actions, you have to be strictly based on the history (previous actions and observations). Never assume anything not explicitly stated in the history. 3. Precise Task Decomposition: You should accurately understand and decompose the given task. Never overlook any details.
GPT	OK, ...

Table 7: Prompt template for behavior cloning and trajectory collection in the Virtualhome dataset.

Format	**Prompt**
Webshop	
Human	You are web shopping. I will give you instructions about what to do. You have to follow the instructions. Every round I will give you an observation and a list of admissible actions, you have to respond an action based on the state and instruction. You can use search action if search is admissible. You can click one of the buttons in clickables. Your output must strictly follow this format:"Thought: your thoughts. Action: your next action". The admissible actions should be of the following structure: search[keywords] click[value] where [keywords] in search are up to you, but the [value] in click must be a value in the list of admissible actions. If the action is not valid, perform nothing. Remember that your keywords in search should be carefully designed. Your response should use the following format: Thought: <your thoughts> Action: <your next action> **Critical INSTRUCTIONS:** 1. Action Format: You must use the exact action formats provided above. NEVER simplify or modify them. 2. Grounding in History (previous actions and observations): When planning your future actions, you have to be strictly based on the history (previous actions and observations). Never assume anything not explicitly stated in the history. 3. Precise Task Decomposition: You should accurately understand and decompose the given task. Never overlook any details.
GPT	OK, ...

Table 8: Prompt template for behavior cloning and trajectory collection in the Webshop dataset.

Role	Content
Human	Imagine you are a trajectory expert of household agents. You are given a completed exploration trajectory of a household agent in the environment for a certain task, but one of the interactive turns of the household agent is lost (represented by the underline '___'). Your goal is to predict the lost interactive turn, like the task of cloze. At the begining of the given trajectory, you can see a detailed description of the environment in which the household agent is located and the task to be accomplished. Then you can see all interactive turns except for the lost one. Each interative turn contains the 'Observation' returned by the environment (Note: if the envrionment outputs "Nothing happened", it means the previous action is invalid), the 'Thought' and 'Action' output by the household agent. At the end, you can see the scalar reward for the given trajectory, which reflects how well the trajectory accomplishes the task ('0.00' and '1.00' represent complete failure and success of the task, respectively). When predicting the lost interactive turn, you should comprehensively consider the visible context before and after the lost interactive turn as well as the final scalar reward. You are only required to output the lost 'Thought' and 'Action'. Your output must strictly follow this format:"Thought: your thoughts. Action: 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. Your response should only include the lost turn in the following format: Thought: <your thoughts> Action: <your next action>
GPT	OK
Human	You are in the middle of a room. Looking quickly around you, you see a armchair 1, a bed 1, a diningtable 1, a drawer 2, a drawer 1, a garbagecan 1, a sidetable 2, and a sidetable 1. Your task is to: find two pillow and put them in armchair.
GPT	___
Human	Observation: On the sidetable 1, you see a cellphone 3, and a keychain 2.
GPT	Thought: There are no pillows on sidetable 1, only a cellphone and a keychain. I should check the other side table. Action: go to sidetable 2
...	... ...
Human	The scalar reward for the above trajectory is: 1.0 Predict the lost interactive turn, like the task of cloze
Target	Thought: I need to find two pillows to place on the armchair. I'll start by searching the nearest surfaces where pillows are likely to be found, such as side tables. Action: go to sidetable 1

Table 9: Prompt template for hindsight model training in the Alfworld & Virtualhome dataset.

Role	Content
Human	Imagine you are a trajectory expert of webshop agents. You are given a completed exploration trajectory of webshop agent in the web for a certain task, but one of the interactive turns of the webshop agent is lost (represented by the underline "___"). Your goal is to predict the lost interactive turn, like the task of cloze. At the beginning of the given trajectory, you can see a detailed description of the task that the webshop agent has to accomplish. Then you can see all interactive turns except for the lost one. Each interative turn contains the "Observation" returned by the web, the "Thought" and "Action" output by the webshop agent. At the end, you can see the scalar reward for the given trajectory, which reflects how well the trajectory accomplishes the task ("0.00" and "1.00" represent complete failure and success of the task, respectively). When predicting the lost interactive turn, you should comprehensively consider the visible context before and after the lost interactive turn, as well as the final scalar reward. Your prediction for the lost interactive turn should include only the "Thought" and "Action" fields, in the format below: Thought: <your thoughts> Action: <your next action>
GPT	OK
Human	WebShop [SEP] Instruction: [SEP] i'm looking for native american indian dream catcher feathers talisman, and price lower than 50.00 dollars [SEP] Search
GPT	___
Human	Observation: Instruction: [SEP] i'm looking for native american indian dream catcher feathers talisman, and price lower than 60.00 dollars [SEP] Back to Search [SEP] Page 1 (Total results: 50) [SEP] Next > [SEP] B07TGHZH66 [SEP] BlessLiving Wolf Dream Catcher Comforter Set Native American Wolf Comforter King Size Bed Set Indian Wolf Bedding Sets King, Brown Gold, 3 Piece, 1 Quilt 2 Shams [SEP] $69.99 [SEP] B09Q61XL96 [SEP] Native American Dream Catcher Beads Women Chiffon Short Sarongs Cover Ups Bikini Wrap Scarf Black [SEP] $7.77 [SEP] B08ZXQW35Z [SEP] Bohemian Dream Catcher Printed Shade Durm Shape Lampshade for Table Lamp and Floor Light, Home Decor, 7.87 R x 7.08 H inches [SEP] $32.99 [SEP] B095VX97GN [SEP] Native American Indian Dream Catcher Feathers Talisman 3 PCs Set, 1 Women Travel Cosmetic Bag, 1 Hanging Toiletry Wash bag, 1 Pencil case, Portable Makeup Organizer Brush Storage Case [SEP] $33.99 [SEP] ...
GPT	Thought: I think I should click on the product that matches the description of a Native American Indian dream catcher feathers talisman and is priced below $60.00. Action: click[B095VX97GN]
...	**... ...**
Human	The scalar reward for the above trajectory is: 1.0 Predict the lost interactive turn, like the task of cloze
Target	Thought: I think I should use the search bar to look for the product I need. Action: search[inner connection to spirit and talisman native american indian dream catcher]

Table 10: Prompt template for hindsight model training in the Webshop dataset.

Dataset	Prompt
Alfworld & Virtualhome	You are an expert at segmenting multi-turn interactive trajectories between intelligent agents and household environments into meaningful stages. You are given a multi-turn interactive trajectory where each round contains a 'user' (the environment: initial task or feedback/observation) and an 'assistant' (the agent: reasoning 'Thought' and chosen 'Action') interaction. Your task is to segment this trajectory into several consecutive segments, where each segment represents a distinct stage or subgoal in the agent's progress toward the specified task. **Critical INSTRUCTIONS:** 1. Segmentation criteria: • Each segment should reflect a clear change in the agent's high-level intention, intermediate sub-goal, strategy shift, or a significant event in the environment (for example, a new search target, a new object found, switching from searching for an object to manipulating it, picking up a crucial object, or a clear environmental cue causes a shift in plan). • Segments can span several turns or just one, but adjacent segments must differ in agent intent or environmental context. • Do not make segments too coarse: Prefer more segments with fewer turns each, unless the trajectory truly supports only a few distinct stages. Too few segments (fewer than {} turns) suggest your segmentation may be coarse. 2. Output requirements: • For each segment you identify, briefly explain the stage transition or boundary in 1-2 sentences; describe what defines the start/end of that segment. • At the end, return all index ranges (1-based) for each segment clearly in this format: [(1, 3), (4, 6), (7), (8, 13), (14, 20)], the index ranges should be consecutive. • Ensure the total number of segments is at least 4 whenever possible, unless the trajectory is exceptionally short. Example: For a trajectory of 8 turns, you may segment as follows: (1, 3): Agent interprets the task and starts searching. (4, 6): Agent finds possible location and investigates. (7): Agent acquires object. (8, 13): Agent manipulates the object. (14, 20): Agent arrives at the destination and puts on the obejct to compelete the task. and then return the index ranges at the end of your response in the format: "[(1, 3), (4, 6), (7), (8, 13), (14, 20)]" Now, analyze the following provided multi-turn interactive trajectory (wrapped in **********) step by step, explain transitions, return the index ranges for each segment in the Square brackets [] at the end of your response in the format: 'The segmented results are: [fill the segmented results here]'

Table 11: Prompt templates for GPT-4o to perform trajectory segmentation across Alfworld & Virtualhome datasets.

Dataset	Prompt
Webshop	`You are an expert at segmenting multi-turn interactive trajectories between intelligent agents and webshopping environments into meaningful stages.` `You are given a multi-turn interactive trajectory where each round contains a 'user' (the environment: initial task or feedback/observation) and an 'assistant' (the agent: reasoning 'Thought' and chosen 'Action') interaction.` `Your task is to segment this trajectory into several consecutive segments, where each segment represents a distinct stage or subgoal in the agent's progress toward the specified task.` **`Critical INSTRUCTIONS:`** `1. Segmentation criteria:` `• Each segment should reflect a clear change in the agent's high-level intention, intermediate sub-goal, strategy shift, or a significant event in the environment (for example, a new search target, switching from searching to clicking, deciding to buy or perform another action, or a clear environmental cue causes a shift in plan).` `• Segments can span several turns or just one, but adjacent segments must differ in agent intent or environmental context.` `• Do not make segments too coarse: Prefer more segments with fewer turns each, unless the trajectory truly supports only a few distinct stages. Too few segments (fewer than {} turns) suggest your segmentation may be coarse.` `2. Output requirements:` `• For each segment you identify, briefly explain the stage transition or boundary in 1-2 sentences; describe what defines the start/end of that segment.` `• At the end, return all index ranges (1-based) for each segment clearly in this format: [(1, 3), (4, 6), (7), (8, 10)]. the index ranges should be consecutive.` `• Ensure the total number of segments is at least 3 whenever possible, unless the trajectory is exceptionally short.`  `Example:`  `For a trajectory of 8 turns, you may segment as follows:` `(1, 3): Agent interprets the task and starts searching.` `(4, 6): Agent searchs relevant objects in the webshop.` `(7): Agent clicks a corresponding button.` `(8, 10): Agent finds the target and clicks the 'Buy Now' button to complete the task.` `and then return the index ranges at the end of your response in the format: "[(1, 3), (4, 6), (7), (8, 10)]"`  `Now, analyze the following provided multi-turn interactive trajectory (wrapped in **********) step by step, explain transitions, return the index ranges for each segment in the Square brackets [] at the end of your response in the format: 'The segmented results are: [fill the segmented results here]'`

Table 12: Prompt templates for GPT-4o to perform trajectory segmentation in the Webshop dataset.