Turn-level Multiscale Density Ratio Estimation for LLM Agents

Anonymous ACL submission

Abstract

With the rapid development of Large language model (LLM), agent systems enhanced by LLMs show huge potential in being able to deal with complex tasks, especially involving multi-step thinking or interaction with tools. For applying LLM techniques with a well-designed agent paradigm, post-training of LLM on multiple agent scenarios is necessary to achieve better performance. Among the variable post-training techniques, alignment methods such as PPO, DPO, DIL and GRPO, become popular since many papers show the significant positive impact on the model's performance by introducing the negative samples to be punished while keeping acceptable training complexity. However, most alignment methods address simple single-turn tasks, and there remains room for improvement for complex multi-turn tasks. We propose turn-level multiscale Density Ratio Estimation (**tlm-DRE**), which assigns different weights on corresponding turns and proposes asymmetric token-level training based on the positive-negative space gaps across multiple turns of tasks. The results of the experiment on a wide range of agent benchmarks show that the proposed method performs competitively compared to traditional alignment methods. The proposed training method enables LLMs to perform robust in multi-turn reasoning tasks with both in-domain and out-of-domain conditions.

1 Introduction

Enhancing agents' capabilities to tackle diverse complex tasks, which often involve interacting with a sophisticated environment equipped with a bunch of tools, has attracted considerable attention. For example, such tasks include complex social dialogue (Wang et al., 2023; Park et al., 2023), scientific experiment (Wang et al., 2022), embodied housework (Shridhar et al., 2020; Li et al., 2024), multi-hop question answering (Yang et al., 2018; Ho et al., 2020), etc.

*Corresponding author.

To accomplish these tasks, LLM-based agents must interact with the environment step by step, decomposing the final goal into sub-goals, and then plan the trajectories based on feedback from the environment. Research on LLM-based agents initially focused on directly generating trajectories using large language models. Most studies employ prompt engineering to enhance the trajectory generation capabilities of large language models, such as CoT (Wei et al., 2022), ReAct (Yao et al., 2022), and Reflexion (Shinn et al., 2023). Subsequent research focused on trajectory tuning to further enhance the agent's trajectory planning capabilities (Chen et al., 2023; Yin et al., 2023).

Meanwhile, reinforcement learning of Large Language Models (Achiam et al., 2023; Touvron et al., 2023; Bai et al., 2023) becomes one of the most important post-training approaches (Kumar et al., 2022) to tune LLMs more applicable to overcome shortcomings, such as hallucinations and logical consistency. Several efficient alignment training methods have been proposed in recent research discourse (Ouyang et al., 2022; Rafailov et al., 2023; Shao et al., 2024). Such post-training paradigms have also been applied to LLM-based agent tasks recently, trying to overcome the drawbacks of simply utilizing the zero-shot LLMs, which neglect agent training.

More specifically, LLM-based agent tasks typically employ the heuristic model(e.g., GPT-4) to generate a group of expert trajectories with a certain CoT form via a set of sampling strategies as a filter. Further supervised fine-tuning (SFT) is then launched to enhance the model's reasoning and planning adaptation to certain domains. Driven by the SFT-trained reference policy, more trajectories are sampled Song et al.; Shi et al.; Xiong et al.; Kong et al. and rewarded step by step to evaluate the capability gaps of the reference model through feedback from the simulated environment. Subsequently, an alignment approach such as RHLF

(Ouyang et al., 2022), DPO (Rafailov et al., 2023), DRE (Xiao et al., 2025), GRPO (Shao et al., 2024) will be applied as a key role of post-training in those tasks to calibrate the sampling distribution by penalizing low-quality trajectories while preserving the original probability mass over high-quality samples.

However, existing methods of alignment on agent tasks still exhibit discrepancies for future development. For example, the aforementioned RL approaches perform alignment directly at the trajectory level while being lack of attention to turn-level details, resulting in suboptimal overall alignment performance. Furthermore, even though a new LLM alignment paradigm has been put forward that views the process as a typical imitation learning under the framework of density ratio estimation, studies on the performance of the "imitation type" of alignment in agent-based tasks remain relatively scarce. Aimed to address these challenges, we propose Turn-level Multiscale Density Ratio Estimation (**tlm-DRE**) for LLM Agents, which applies imitation learning on agent-based tasks with turn-level alignment efficiency.

In particular, we assign lower weights to the turns of the samples with higher policy confidence, since there is no need to further align between chosen and rejected sample pairs. On the other hand, we assign higher weights to the turns with lower policy confidence since those turns are more crucial to be calibrated in the language space. Then, we design a novel density ratio estimation definition under turn-level varying scales based on the turn weights derived from the reference policy. Afterwards, we launch the alignment over extensive agent-based tasks in the framework of imitation learning.

Our contributions are as follows: (1) We propose a novel **tlm-DRE** method that systematically applies imitation learning to agent-based tasks in a turn-level multiscale probability space. (2) We conduct extensive experiments on several agent-based tasks (Shridhar et al., 2020; Wang et al., 2022; Yang et al., 2018) to show that our method surpasses current methods with stable and robust performance. (3) We present a comprehensive analysis to support the efficacy of our method from various points of view.

2 Related Work

LLM application in agent-based tasks The development of LLM has inspired the intelligent agents designing complex tasks involved with a dynamic environment and a complicated real-world toolbox. The main motivation to utilize LLM in those tasks is that the agents need strong abilities of reasoning(both externally-oriented and reflectional) and action planning. CoT (Wei et al., 2022) enables LLM to articulate its own thought processes, enhancing its reasoning capabilities and laying the groundwork for subsequent agent reasoning frameworks. ReAct (Yao et al., 2022) integrates feedback from the environment into its reasoning process, allowing the model to think while taking actions and interacting with the environment, and adjust subsequent actions based on that feedback. Reflexion (Shinn et al., 2023) builds upon ReAct by incorporating LLM self-reflection, allowing the model to autonomously correct its previous erroneous actions. There are even more complicated multi-agent designs (Park et al., 2023) that simulate believable human behavior in the simulated sandbox.

Alignment process: Reinforcement Learning Many researchers focus on using reinforcement learning to further enhance agent performance. ETO (Song et al., 2024) introduced Direct Preference Optimization for post alignment. They used the SFT model as a reference policy to generate bad cases, which were then paired with expert trajectories to form sample pairs. DMPO (Shi et al., 2024) has noted that training directly with DPO leads to mismatching trajectory step lengths between positive and negative samples, and length regularization has been proposed. In contrast, SDPO (Kong et al., 2025) directly utilizes GPT to capture positive and negative samples of equal step lengths, avoiding the issue of mismatched step length.

Some research approaches agent training using online RL, such as GRPO (Shao et al., 2024), GSPO (Zheng et al., 2025). Such kind of methods are group-based RL approaches that are critic-free, making training simple and stable. GiGPO (Feng et al., 2025) groups objects that share identical environmental interaction states, employing this hierarchical Group-in-Group structure to train GRPO. These methods have also been shown to be highly effective in post-alignment.

Alignment process: Imitation Learning Other research focuses on applying imitation learning(IR)

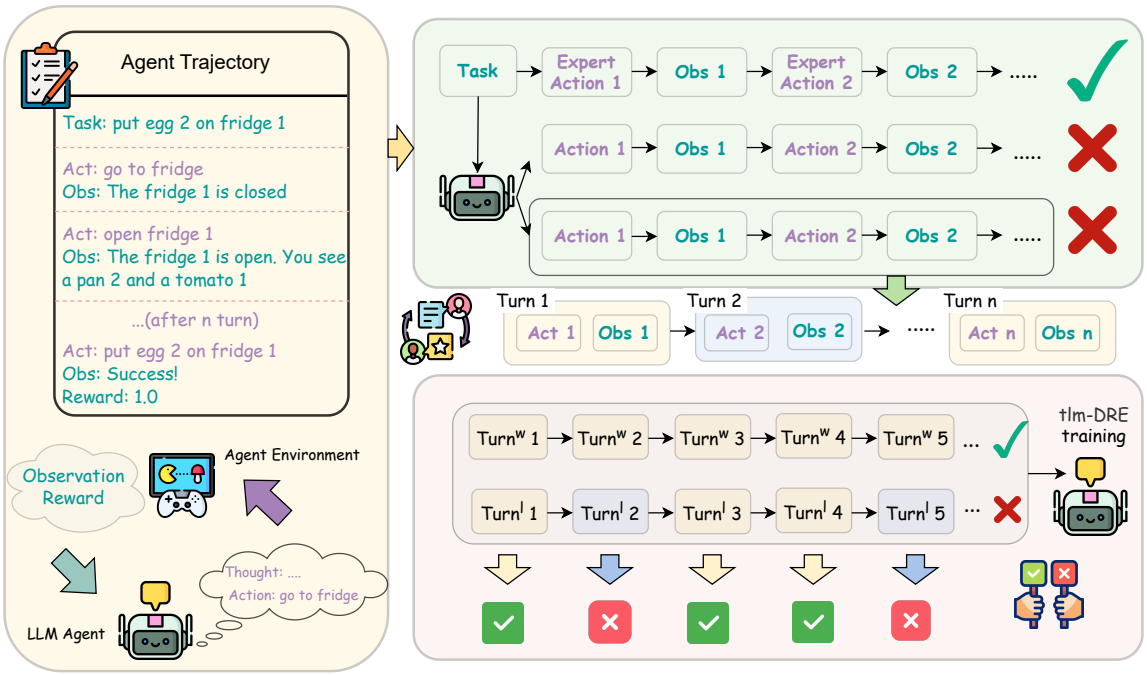

Figure 1: The overall architecture of **tlm-DRE** in a single iteration. First, the agent initialized with the SFT policy samples trajectory paths and collects failure trajectories. Then, within these failure trajectories, we identify low-confidence turns—i.e., turns where the model exhibits low self-confidence. Finally, we train the agent using Multiscale Density Ratio Estimation, which explicitly upweights these identified failure turns during optimization.

in the artificial intelligence domain. The key idea of IR is to directly extract knowledge from demonstrations by human experts or artificially created agents by replicating their behavior. Inverse Reinforcement Learning has been proposed to recover the reward function of empirical resources from the uncertain environment (Russell, 1998; Ng et al., 2000; Sun and van der Schaar, 2024). Behavioral cloning (Pomerleau, 1991; Ross et al., 2011) efficiently bridges the domain gap by supervised fine-tuning on expert trajectories. The adversarial Imitation Learning method, such as GAIL (Ho and Ermon, 2016), AIRL (Fu et al., 2018), leverages adversarial training with online interaction with the environment.

Density Ratio Estimation (Sugiyama et al., 2012; Amari and Cichocki, 2010) can be viewed as a typical way to mimic the behavior of experts under a certain distribution space. DRE can be measured by optimizing the discrepancy between the learnable policy and the ideal distribution equipped with the Bregman divergence. Some methods have already tried to transplant this idea into the post training of LLM based tasks, such as DIL (Xiao et al., 2025), GSIL (Xiao et al., 2024). Furthermore, DIL (Xiao et al., 2025) also shows that traditional reinforcement learning methods such as RLHF and DPO are actually a form of imitation learning.

3 Notations and Preliminaries

3.1 Task Formulation

Consider one agent which is designed to deal with a certain bunch of complex tasks interacting with the real world or simulation environment. Let $o_i \in \mathcal{O}$ represent the observation of the agent for the i turn of interaction with the environment. When $i = 1$, o_i represents the initial description of a given task and the initial status of the agent. Based on the agent's observation from the environment, as well as the previous context or trajectory $c_t = (o_1, a_1, ..., o_{t-1}, a_{t-1}, o_t)$, a type of action from the action space $\mathcal{A}$ will be chosen after a reasoning process. To avoid redundancy, we might as well view the thinking process and the action that the agent takes as a whole, named $a_t \in \mathcal{A}$ for t turns, which follows a parameterized policy $\pi_\theta(a_t|c_t)$.

For simplicity, the potential stochastic effect of how the action is being executed is neglected. Furthermore, if the policy π is driven by a certain large language model, then the action space $\mathcal{A}$ is equivalent to a language space $\mathcal{L}$. In detail, the whole trajectory c_T can be viewed as $(\boldsymbol{x}, \boldsymbol{y_T})$, where

$\boldsymbol{x} = [x_1, x_2, ...]$ belongs to L while x_i is the language token. Similarly, $\boldsymbol{y_T} = [\boldsymbol{y_{a_1}}, \boldsymbol{y_{o_2}}, \boldsymbol{y_{a_2}}, ..]$, $\boldsymbol{y_{o_i}}, \boldsymbol{y_{a_i}}$ also belong to the language space $\mathcal{L}$.

Denote π_{ref} as the reference policy. Based on the reference policy and the given input language sequences $\boldsymbol{x}$, several trajectories $\boldsymbol{y}$ will be sampled following a sample strategy. Denote $\boldsymbol{y_w}$ as the preferred trajectory which is marginally superior to another trajectory $\boldsymbol{y_l}$, based on the final behavior and the evaluation of the whole process. Suppose that the ideal optimized policy of the entire policy space is π_c. The object is to find the optimized projected policy π_θ for a parameterized policy space

3.2 Directed Preference Optimization

For standard reinforcement learning from human feedback(RLHF), a reward model is trained to evaluate whether the policy behaves well enough, such as PPO. (Rafailov et al., 2023) proposes a way to directly use π_θ / π_{ref} as the reward function $r(\boldsymbol{y}|\boldsymbol{x})$ and interpret the alignment process as a task to optimize the margin between the pair-wise samples

Bradley-Terry Model The PPO method applies a trained reward model to Within the framework of probability theory, the Bradley-Terry model is applied to measure the partial order relation of a sample pair $(\boldsymbol{y_w}, \boldsymbol{y_l})$ stipulating the human preference distribution p^* as

$$p^*(\boldsymbol{y_w} \succ \boldsymbol{y_l}) = \sigma(r(\boldsymbol{x}, \boldsymbol{y_w}) - r(\boldsymbol{x}, \boldsymbol{y_l})), \quad (1)$$

where σ is the sigmoid function.

Assuming a static dataset of pair-wise samples $\mathcal{D} = \{\boldsymbol{x}^i, \boldsymbol{y}_w^i, \boldsymbol{y}_l^i\}_{i=1}^N$ exists, then the DPO method will try to find a reward model r_θ in a parametrized model space that minimizes the following negative log-likelihood loss

$$\mathcal{L}_{DPO} = -\mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l \sim \mathcal{D})}[log\sigma(r_\theta(\boldsymbol{y}_w) - r_\theta(\boldsymbol{y}_l))] + \beta \mathbb{D}_{kl}[\pi_\theta \| \pi_{ref}]. \quad (2)$$

3.3 Density Ratio Estimation

Bregman Divergence Denote the density ratio as r_θ as $\frac{\pi_\theta}{\pi_{ref}}$, the true density ratio is r^* as $\frac{\pi_c}{\pi_{ref}}$. To employ the Bregman divergence(BR) for estimating the density ratio according to (Sugiyama et al., 2012), let f be a differentiable and strictly convex function on the density-ratio model space. the discrepancy from the true density ratio r^* to the density-ratio model r_θ can be measured as

$$BR'_f(r^* \| r_\theta) = (f(r^*) - f(r_\theta) - \partial f(r_\theta)(r^* - r_\theta)). \quad (3)$$

Fig. 4 in Appendix A illustrates how the current point r_θ can slide towards the target point r^* iteratively by optimizing the Bregman divergence. Several kernel functions f for Bregman divergence can be found in Appendix A.

Directly Imitation Learning Under the sample assumption that we have datasets $\mathcal{D}$ defined above. The DIL method (Xiao et al., 2025) tries to imitate the optimal density ratio by minimizing the discrepancy under the Bregman divergence in a parameterized constriction

$$\min_\theta D_h(r^* \| r_\theta) = \sum_{\boldsymbol{y}} \pi_{ref}(f(r^*) - f(r_\theta) - \partial f(r_\theta)(r^* - r_\theta)). \quad (4)$$

Removing the θ irrelevant constant and under the assumption that $\boldsymbol{y}_l \sim \pi_{ref}(\boldsymbol{y}_l \| \boldsymbol{x})$, then we get the loss function of the DIL method

$$\mathcal{L}_{DIL} = -\mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l \sim \mathcal{D})}\{\partial f(r_\theta(\boldsymbol{y}_w)) - [\partial f(r_\theta(\boldsymbol{y}_l)) r_\theta(\boldsymbol{y}_l) - f(r_\theta(\boldsymbol{y}_l))]\}. \quad (5)$$

4 Methodology

Due to varying degrees of training exposure of tokens in the embedding space within the reference model, one daunting challenge is how to train the policy over the candidate tokens more efficiently. This problem warrants more attention for the tasks that need multi-step thinking and reasoning. (Liu et al., 2025) introduced a way to allocate token weights to each token according to a certain importance measurement for the DPO. However, this idea cannot be trivially generalized to turn-level optimization for the DRE loss due to the fact that most of the Bregman divergence functions show a high degree of asymmetry and nonlinearity. We will introduce a new density ratio function that will be used to solve this problem. Our method framework is illustrated in Figure 1.

4.1 Multiscale Density Ratio Representation

Since the density ratio is designed to estimate how close an approximation to the optimal solution can be achieved over the output language space, we can only focus on the linguistic action results $\boldsymbol{y_{a_i}}$ under the assumption that the observation results are determined by omitting the stochastic effect from the simulation environment. More specifically, the

density ratio can be expressed as

$$r(\boldsymbol{y_t}\|\boldsymbol{x}) = \frac{\pi(\boldsymbol{y_t}\|\boldsymbol{x})}{\pi_{ref}(\boldsymbol{y_t}\|\boldsymbol{x})} = \frac{\pi(\boldsymbol{y_{a_t}}\|\boldsymbol{x}, \boldsymbol{c_{t-1}})...\pi(\boldsymbol{y_{a_1}}\|\boldsymbol{x})}{\pi_{ref}(\boldsymbol{y_{a_t}}\|\boldsymbol{x}, \boldsymbol{c_{t-1}})...\pi_{ref}(\boldsymbol{y_{a_1}}\|\boldsymbol{x})}, \tag{6}$$

where $\boldsymbol{c}_{t-1} = [\boldsymbol{y_{a_1}}, ..., \boldsymbol{y_{o_{t-1}}}]$

The main drawback of the traditional expression of the DRE is the inefficiency of training the underfitting, while well-trained tokens will possibly suffer the risk of overfitting. Meanwhile, as demonstrated in (Liu et al., 2025), the importance of each token in latent embedding spaces is not uniformly distributed, implying heterogeneous training demands. In the agent-related tasks with multi-step reasoning, the importance weighting with non-uniformity becomes more pronounced from the turn-level perspective.

Suppose that a turn-level weighted function

$$\omega_t = \omega(\boldsymbol{y_{a_t}}\|\boldsymbol{x}, \boldsymbol{c_{t-1}}), \tag{7}$$

that measures how important and how well-trained each turn in the agentic reasoning process is. Then the density ratio is to be measured as

$$r(\boldsymbol{y_t}\|\boldsymbol{x}) = \frac{\hat{\pi}(\boldsymbol{y_t}\|\boldsymbol{x})}{\hat{\pi}_{ref}(\boldsymbol{y_t}\|\boldsymbol{x})} = \frac{\pi^{\omega_t}(\boldsymbol{y_{a_t}}\|\boldsymbol{x}, \boldsymbol{c}_{t-1})...\pi^{\omega_1}(\boldsymbol{y_{a_1}}\|\boldsymbol{x})}{\pi^{\omega_t}_{ref}(\boldsymbol{y_{a_t}}\|\boldsymbol{x}, \boldsymbol{c}_{t-1})...\pi^{\omega_1}_{ref}(\boldsymbol{y_{a_1}}\|\boldsymbol{x})}. \tag{8}$$

4.2 Turn-level Weighted Measurement

We define the adequacy estimate of the i-th turn output $\boldsymbol{y_i}$ as p_i:

$$\begin{aligned} \log p_i &= \frac{1}{|\boldsymbol{y_i}|}\log \pi_{ref}(\boldsymbol{y}_{a_i}\|\boldsymbol{x}, \boldsymbol{c}_{i-1}) \\ &= \frac{1}{|\boldsymbol{y_i}|}\sum_{j=1}^{|\boldsymbol{y}_i|}\log \pi_{ref}(y_{i,j}|\boldsymbol{x}, \boldsymbol{c}_{i-1}, y_{i,<j}), \end{aligned} \tag{9}$$

where $|\boldsymbol{y}_i|$ is the total number of tokens of the $\boldsymbol{y}_i$.

In some previous searches (Nagumo and Fujisawa, 2024), they view the statistical possibility of the output as an outlier measurement. However, we argue that, under the assumption that previous training in the agent-based task is inefficient and imbalanced, the adequacy estimation actually shows more weight on how well trained the ith turn of the trajectory based on the reference policy is, rather than the outliers measurement. Hence, a natural point is to make sure that there is a negative correlation between the adequacy estimation and the turn-weights applied on the following alignment.

We define the turn-level weights ω_i for the i-th turn of the trajectory as

$$\omega_i = \begin{cases} \omega_L & p_i > p_{pivot}, \\ \omega_U & p_i \le p_{pivot}. \end{cases} \tag{10}$$

4.3 Turn-level Weighted DRE Optimization

Applying the turn-level weights version of density ratio as defined in Equation (8) on the DRE optimization, Equation (4) becomes

$$\begin{aligned} \min_\theta D_{\omega,f}(\hat{r}^*\|\hat{r}_\theta) = \sum_{\boldsymbol{y}} \pi_{ref}(f(\hat{r}^*) \\ - f(\hat{r}_\theta) - \partial f(\hat{r}_\theta)(\hat{r}^* - \hat{r}_\theta)). \end{aligned} \tag{11}$$

Subtracting the constant $\sum_{\boldsymbol{y}} \pi_{ref} f(\hat{r}^*)$, we obtain

$$\begin{aligned} \min_\theta D_{\omega,f}(\hat{r}^*\|\hat{r}_\theta) = -\sum_{\boldsymbol{y}} \pi_{ref}\partial f(\hat{r}_\theta)\hat{r}^* \\ + \sum_{\boldsymbol{y}} \pi_{ref}(\partial f(\hat{r}_\theta)\hat{r}_\theta - f(\hat{r}_\theta)). \end{aligned} \tag{12}$$

The last term above $\pi_{ref}\partial f(\hat{r}_\theta)\hat{r}^*$ can be expanded as

$$\begin{aligned} &\pi_{ref}\partial f(\hat{r}_\theta)\hat{r}^* \\ &= \pi_{ref}\partial f(\hat{r}_\theta)\prod_{i=1}^{t}\frac{\pi_c^{\omega_i}(\boldsymbol{y_{a_i}}\|\boldsymbol{x}, \boldsymbol{c}_{i-1})}{\pi_{ref}^{\omega_i}(\boldsymbol{y_{a_i}}\|\boldsymbol{x}, \boldsymbol{c}_{i-1})} \\ &= \partial f(\hat{r}_\theta)\prod_{i=1}^{t}\frac{\pi_c^{\omega_i}(\boldsymbol{y_{a_i}}\|\boldsymbol{x}, \boldsymbol{c}_{i-1})}{\pi_{ref}^{\omega_i-1}(\boldsymbol{y_{a_i}}\|\boldsymbol{x}, \boldsymbol{c}_{i-1})}. \end{aligned} \tag{13}$$

Due to the fact that the possibility of the un-chosen trajectory by policy π_c is vanishingly small and $\omega \in [\omega_L, \omega_U]$, Substituting (13)into (12) the equation yields

$$\begin{aligned} &\min_\theta D_{\omega,f}(\hat{r}^*\|\hat{r}_\theta) \\ &= \sum_{\boldsymbol{y}} \pi_{ref}(\partial f(\hat{r}_\theta)\hat{r}_\theta - f(\hat{r}_\theta)) \\ &- \sum_{\boldsymbol{y}_w} \pi_c \partial f(\hat{r}_\theta)\prod_{i=1}^{t}\frac{\pi_c^{\omega_i-1}(\boldsymbol{y_{a_i}}\|\boldsymbol{x}, \boldsymbol{c}_{i-1})}{\pi_{ref}^{\omega_i-1}(\boldsymbol{y_{a_i}}\|\boldsymbol{x}, \boldsymbol{c}_{i-1})}, \end{aligned} \tag{14}$$

under the same assumption that $\boldsymbol{y}_l \sim \pi_{ref}(\boldsymbol{y}_l\|\boldsymbol{x})$ the loss function of (**tlm-DRE**) is

$$\begin{aligned} &\mathcal{L}_{tlm-DRE} \\ &= \int_{\boldsymbol{y}_l}[\partial f(\hat{r}_\theta(\boldsymbol{y}_l))\hat{r}_\theta(\boldsymbol{y}_l) - f(\hat{r}_\theta(\boldsymbol{y}_l))] \\ &- \int_{\boldsymbol{y}_w}\partial f(\hat{r}_\theta)\prod_{i=1}^{t}\frac{\pi_c^{\omega_i-1}(\boldsymbol{y_{a_i}}\|\boldsymbol{x}, \boldsymbol{c}_{i-1})}{\pi_{ref}^{\omega_i-1}(\boldsymbol{y_{a_i}}\|\boldsymbol{x}, \boldsymbol{c}_{i-1})}. \end{aligned} \tag{15}$$

Since the distribution shows significant concentration on the candidate tokens from the chosen trajectory over the language space, we can assume that $\pi_c(\boldsymbol{y_{a_i}} \| \boldsymbol{x}, \boldsymbol{c}_{i-1}) \sim 1$ uniformly, leading to the final **tlm-DRE** loss for training

$$\begin{aligned}
&\mathcal{L}_{tlm-DRE} \\
&= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l\sim\mathcal{D})}\{[\partial f(\hat{r}_\theta(\boldsymbol{y}_l))\hat{r}_\theta(\boldsymbol{y}_l) - f(\hat{r}_\theta(\boldsymbol{y}_l))] \\
&- \partial f(\hat{r}_\theta(\boldsymbol{y}_w)) \prod_{i=1}^{t} \frac{\pi_c^{\omega_i-1}(\boldsymbol{y_{w,a_i}} \| \boldsymbol{x}, \boldsymbol{c}_{w,i-1})}{\pi_{ref}^{\omega_i-1}(\boldsymbol{y_{w,a_i}} \| \boldsymbol{x}, \boldsymbol{c}_{w,i-1})}\} \\
&\approx \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l\sim\mathcal{D})}\{[\partial f(\hat{r}_\theta(\boldsymbol{y}_l))\hat{r}_\theta(\boldsymbol{y}_l) - f(\hat{r}_\theta(\boldsymbol{y}_l))] \\
&- \partial f(\hat{r}_\theta(\boldsymbol{y}_w)) \prod_{i=1}^{t} \frac{1}{\pi_{ref}^{\omega_i-1}(\boldsymbol{y_{w,a_i}} \| \boldsymbol{x}, \boldsymbol{c}_{w,i-1})}\}.
\end{aligned} \tag{16}$$

In this paper, we use UKL (Nguyen et al., 2010) defined in Appendix A as the kernel function for Bregman divergence. Then equation (16) will be as

$$\begin{aligned}
&\mathcal{L}_{tlm-DRE} \\
&= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l\sim\mathcal{D})}\{\prod_i \frac{\pi_\theta^{\omega_{l,i}}(\boldsymbol{y}_{l,i})}{\pi_{ref}^{\omega_{l,i}}(\boldsymbol{y}_{l,i})} - \\
&\sum_i \frac{\omega_{w,i}}{\prod_{i=1}^{t} \pi_{ref}^{\omega_i-1}(\boldsymbol{y_{w,i}})} \log \frac{\pi_\theta(\boldsymbol{y}_{w,i})}{\pi_{ref}(\boldsymbol{y}_{w,i})}\}.
\end{aligned} \tag{17}$$

5 Experiments

In this section, we demonstrate the performance of **tlm-DRE** across various agent based tasks, provide detailed experimental procedures, and introduce other related baselines.

5.1 Experimental Settings

Datasets We perform experiments across a diverse set of environments to evaluate the capabilities of our agent. Specifically, we use SciWorld (Wang et al., 2022) under the Apache-2.0 license for grounded scientific experimentation in simulated labs, ALFWorld under the MIT License (Shridhar et al., 2020) for embodied household tasks requiring object manipulation and planning in 3D environments. In addition, we also include HotpotQA (Yang et al., 2018) under the CC BY-SA 4.0 license as the benchmark of multi-hop reasoning in open-domain settings, a task that requires the agent to retrieve and integrate information from multiple sources to answer complex questions. SciWorld provides dense final rewards on a continuous scale from 0 to 1, whereas ALFWorld offers only sparse binary rewards that indicate whether the task was completed or not. For multi-hop QA tasks, we measure reasoning performance using EM and F1 scores against ground-truth answers.

Baselines We compare our approach with a series of benchmarks: (1) Zero-shot using LLMs such as GPT-4o, Qwen2.5-7B, applying the ReAct prompting paradigm, which represents the state-of-the-art zero-shot capabilities of LLMs.(2) SFT (Supervised Fine-Tuning) conducts behavioral cloning on expert trajectories. (3) PPO (Proximal Policy Optimization) as an actor-critic reinforcement learning algorithm to directly optimize the SFT-initialized policy. (4) ETO (Song et al., 2024) uses successful and failure trajectories as sample pairs for DPO training. (5) DMPO (Shi et al., 2024) adds length regularization to ETO to eliminate noise caused by failure trajectories with excessive steps. (6) GiGPO (Feng et al., 2025) groups objects with identical environmental interaction states, employing the hierarchical group structure to train GRPO.

All experiments were conducted on 8x Nvidia A100 GPUs, each with 80GB of memory, implemented using PyTorch in Python. Details of the experiment setup can be found in Appendix B.

5.2 Main Results

Results on Interactive Tasks Table 1 demonstrates the strong performance of **tlm-DRE** on both ALFWorld and SciWorld. As shown, prompt-only ReAct baselines achieve only moderate results: GPT-4 attains 38.1 average reward on ALFWorld (unseen) and 64.4 on SciWorld (unseen), while GPT-3.5 lags significantly behind. Qwen2.5-7B performs modestly in all settings, with an average reward of around 25, indicating limited effectiveness without further alignment or training.

For SFT and RL training, most prior work adopts LLaMA2-7B as the backbone. To ensure a maximally fair comparison, we also report results with LLaMA2-7B. We further evaluate a stronger and more recent model, Qwen2.5-7B. ETO directly applies DPO to agent tasks, avoiding the need for a critic network as in PPO and thus offering a lighter and simpler training pipeline. With LLaMA2-7B, ETO achieves 72.4 score on ALFWorld (*unseen*) and 61.1 on SciWorld (*unseen*), though it is weaker in the seen setting. DMPO performs particularly well on SciWorld with LLaMA2-7B, reaching 72.4 on SciWorld (*seen*), but degrades on ALFWorld. Based on the same base model, our method attains the best results in the seen tasks on both datasets;

Paradig	Models	ALFWorld		ScienceWorld	
		Seen	Unseen	Seen	Unseen
Prompt-based	GPT-3.5 (Ouyang et al., 2022)	7.9	10.5	16.5	13.0
	GPT-4 (Achiam et al., 2023)	42.9	38.1	64.8	64.4
	Qwen2.5-7B (Team et al., 2024)	25.1	28.4	26.8	25.2
Llama-2-7B-Chat					
SFT & RL	SFT (Chen et al., 2023)	60.0	67.2	56.8	56.0
	PPO (Trung et al., 2024)	22.1	29.1	59.4	51.7
	RFT (Zhang et al., 2023)	62.9	66.4	71.6	54.3
	ETO (Song et al., 2024)	68.6	72.4	68.5	61.1
	DMPO (Shi et al., 2024)	43.3	55.0	72.4	61.7
	tlm-DRE (ours)	**70.7**	**72.4**	**73.0**	61.2
Qwen2.5-7B-instruct					
RL-training	SFT (Chen et al., 2023)	70.7	83.6	71.8	61.8
	ETO (Song et al., 2024)	75.0	86.6	69.1	62.8
	tlm-DRE w/o DRE	75.0	86.6	72.1	63.3
	tlm-DRE w/o tlm	75.7	88.8	71.9	63.7
	tlm-DRE (ours)	**75.7**	**90.3**	**73.1**	**63.8**

Table 1: Performance of different methods on ALFWorld and SciWorld, reported as average reward. "Seen" denotes the held-out test set containing task types observed during training, while "Unseen" refers to test tasks with critical unseen variations (e.g., novel object or goal). "tlm-DRE w/o DRE" denotes the linearly turn-weighted DPO training. For fair comparison, all methods use the same base models: LLaMA2-7B and Qwen2.5-7B.

Method	ALFWorld (all)
RLOO (Ahmadian et al., 2024)	75.5
GRPO (Shao et al., 2024)	77.6
GiGPO w/ std (Feng et al., 2025)	90.8
GiGPO w/o std (Feng et al., 2025)	90.2
tlm-DRE (ours)	**92.4**

Table 2: For fair comparison, results are evaluated in the GiGPO (Feng et al., 2025) test environment (with longer exploration steps and different test categories). All methods use the same base models: Qwen2.5-7B-Instruct

Method	HotpotQA	
	EM	F1
SFT (Chen et al., 2023)	27.80	36.45
CoH (Liu et al., 2023)	28.60	39.53
PPO (Trung et al., 2024)	28.20	36.47
DPO (Song et al., 2024)	26.40	34.83
NAT (Wang et al., 2025)	29.60	42.50
tlm-DRE (ours)	**33.8**	**43.74**

Table 3: Overall results on open-domain question answering tasks. We measure the performance using Exact Match and F1 score. For fair comparison, all methods use the same base models: LLaMA2-7B

moreover, when switching to Qwen2.5-7B, it further boosts ALFWorld (*unseen*) to 90.3, far surpassing the LLaMA2-7B counterpart.

GiGPO uses a different evaluation protocol with longer exploration steps. For a fair comparison, we also evaluated our method in the GiGPO environment. As shown in Table 2, our approach outperforms GiGPO by approximately three points in overall success reward, demonstrating strong effectiveness in this more challenging setting.

Results on Multi-Hop QA Tasks As shown in Table 3, **tlm-DRE** consistently improves performance on multi-turn search-augmented QA tasks, achieving an Exact Match (EM) of 33.8 and an F1 score of 43.74 on HotpotQA, substantially outperforming strong baselines such as NAT. Although search-augmented QA uses a different set of tools and typically requires fewer exploration steps than traditional agent tasks (e.g., ALFWorld), it implies generalization of our method.

Ablation Study We conduct ablation studies to evaluate the effectiveness of each component: (1) Density Ratio Estimation (DRE) with UKL-kernel for Bregman divergence. (2) A variant version of DRE proposed with turn-level multiscale. Table 1 shows that both components yield clear improve-

ments over standard SFT and DPO. Specifically, turn-level weights and DRE each contribute independently, but exhibit different strengths across evaluation splits. We observe that the turn-level weights tend to provide larger gains in the *seen* setting on both datasets, suggesting that emphasizing critical turns helps better fit in-distribution interaction patterns. Moreover, **tlm-DRE** brings more consistent improvements in *unseen* tasks and achieves the best overall performance.

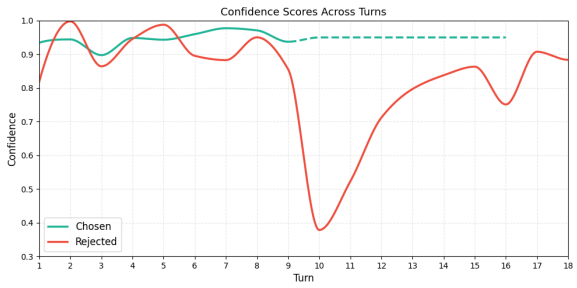

Figure 2: Illustration of confidence scores for different turns computed using the SFT policy, shown for both the chosen and rejected trajectories.

6 Analysis

6.1 Illustration of turn weights

Previous methods, such as ETO and DMPO, often view the entire trajectory as a unified object, thereby neglecting the contribution gaps of different turns within the trajectory. For example, in a failed trajectory, not every turn is necessarily incorrect. Therefore, as shown in Eq. (9), we use the SFT policy to recompute the confidence scores of the model in different turns for both positive and negative examples. As illustrated in Fig. 2, we present the confidence scores of the SFT policy on different turns for both chosen expert trajectories and rejected trajectories. We observe that, in the rejected trajectories, the model exhibits notably low confidence at Turns 10 and 11. By examining the chosen expert trajectories and specific case studies of Fig. 5, we find that these two turns are indeed critical steps that lead the entire trajectory astray. Consequently, in the subsequent alignment stage, we place particular emphasis on such uncertain turns, and our experimental results confirm that this idea is effective.

6.2 Margin analysis

Fig. 3 shows the reward margin between the chosen and rejected trajectories during training for tlm-

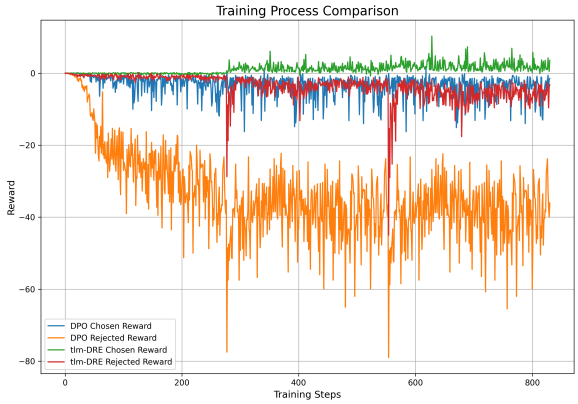

Figure 3: Illustration of the reward margin between the chosen and rejected trajectories during training for tlm-DRE and DPO

DRE and DPO. We argue that DPO separates chosen and rejected trajectories in a coarse-grained manner, treating a failed trajectory as entirely negative and neglecting that most turns within it can still be correct. As a result, many correct turns from rejected trajectories may be pushed toward the negative region, which can degrade overall performance. In practice, the failure trajectory is often caused by only a few critical erroneous steps.

To address this issue, our method up-weights critical turns and implicitly down-weights correct turns to be punished. Concretely, we only decrease the reward for the key erroneous turns while keeping the weights of other turns almost unchanged. As illustrated in Fig. 3, our method produces a chosen-rejected margin smaller than the standard DPO, reflecting a finer-grained distinction between positive and negative samples. The empirical results further support the effectiveness of this design.

7 Conlusions

In this paper, Turn-level Multiscale Density Ratio Estimation (**tlm-DRE**) is proposed to enhance LLM performance in multi-turn agent tasks. We theoretically design a multiscale density ratio representation that fully leverages the contrastive information between positive and negative samples. Our approach assigns turn-specific weights under the framework of DRE to make the alignment of multi-turn tasks more flexible and robust. Our approach has demonstrated strong performance across multiple multi-turn agent tasks, including ALFWorld, SciWorld, and HotpotQA. In the future, we will explore the generalizability of this method across more multi-turn scenarios.

8 Limitations

Our Turn-level Multiscale Density Ratio Estimation (**tlm-DRE**) employs pair-level asymmetric training with turn weights assigned to key steps, which adapts well to multi-turn agent tasks with long trajectories. However, this study is still limited from several perspectives, pointing to promising directions for future research. First, **tlm-DRE** is conducted mainly based on the offline sampling strategy, traditionally adopted by PPO or DPO. Whether the gain is still maintained when we combine the proposed alignment method with the dynamic sampling strategy as used in GRPO or GSPO deserves careful investigation. Second, while the UKL kernel of Bregman divergence shows advances in several multi-turn agent tasks, systematic analysis of the impact of varying the choice of different function kernels needs to be further conducted. Moreover, with the rapid development of the AI/LLM agent area, there are more agent-based tasks emerging in recent years, which need more complex task decomposition and tool-using strategy, as well as more sophisticated environmental interaction. Extending the experiment of our method to more relevant tasks also warrants an in-depth analysis. We hope our work will inspire researchers to explore multi-step agent training in this field.

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*.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*.

Shun-ichi Amari and Andrzej Cichocki. 2010. Information geometry of divergence functions. *Bulletin of the polish academy of sciences. Technical sciences*, 58(1):183–195.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Ayanendranath Basu, Ian R Harris, Nils L Hjort, and MC Jones. 1998. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559.

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*.

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

Justin Fu, Katie Luo, and Sergey Levine. 2018. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*.

Trevor Hastie. 2009. The elements of statistical learning: data mining, inference, and prediction.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. 2009. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Aobo Kong, Wentao Ma, Shiwan Zhao, Yongbin Li, Yuchuan Wu, Ke Wang, Xiaoqian Liu, Qicheng Li, Yong Qin, and Fei Huang. 2025. Sdpo: Segment-level direct preference optimization for social agents. *arXiv preprint arXiv:2501.01821*.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. 2022. Llm post-training: a deep dive into reasoning large language models (2025). *URL https://arxiv.org/abs/2502.21321*, 3(7).

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, and 1 others. 2024. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*.

Aiwei Liu, Haoping Bai, Zhiyun Lu, Yanchao Sun, Xiang Kong, Xiaoming Wang, Jiulong Shan, Albin Madappally Jose, Xiaojiang Liu, Lijie Wen, Philip Yu, and Meng Cao. 2025. Tis-dpo: Token-level importance sampling for direct preference optimization with estimated weights. In *International Conference*

on Representation Learning, volume 2025, pages 51339–51368.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023. Chain of hindsight aligns language models with feedback. *arXiv preprint arXiv:2302.02676*.

Ryosuke Nagumo and Hironori Fujisawa. 2024. Density ratio estimation with doubly strong robustness. In *Forty-first International Conference on Machine Learning*.

Andrew Y Ng, Stuart Russell, and 1 others. 2000. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. 2010. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861.

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.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Dean A Pomerleau. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97.

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.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Stuart Russell. 1998. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103.

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*.

Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. 2024. Direct multi-turn preference optimization for language agents. *arXiv preprint arXiv:2406.14868*.

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 for llm agents. *arXiv preprint arXiv:2403.02502*.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 2012. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044.

Hao Sun and Mihaela van der Schaar. 2024. Inverse-rlignment: Inverse reinforcement learning from demonstrations for llm alignment. *arXiv preprint arXiv:2405.15624*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

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

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7601–7614.

Renxi Wang, Xudong Han, Yixuan Zhang, Timothy Baldwin, and Haonan Li. 2025. Nat: Enhancing agent tuning with negative samples. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7385–7398.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang,

and 1 others. 2023. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv preprint arXiv:2310.00746*.

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.

Teng Xiao, Mingxiao Li, Yige Yuan, Huaisheng Zhu, Chao Cui, and Vasant G Honavar. 2024. How to leverage demonstration data in alignment for large language model? a self-imitation learning perspective. *arXiv preprint arXiv:2410.10093*.

Teng Xiao, Yige Yuan, Mingxiao Li, Zhengyu Chen, and Vasant G Honavar. 2025. On a connection between imitation learning and RLHF. In *The Thirteenth International Conference on Learning Representations*.

Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. 2024. Watch every step! llm agent learning via iterative step-level process refinement. *arXiv preprint arXiv:2406.11176*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

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*.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. 2023. Agent lumos: Unified and modular training for open-source language agents. *arXiv preprint arXiv:2311.05657*.

Yifan Zhang, Jingqin Yang, Yang Yuan, and Andrew Chi-Chih Yao. 2023. Cumulative reasoning with large language models. *arXiv preprint arXiv:2308.04371*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

A Details of Density Ratio Estimation

Fig. 4 shows how optimizing the Bregman divergence gradually drives the point toward the target point.

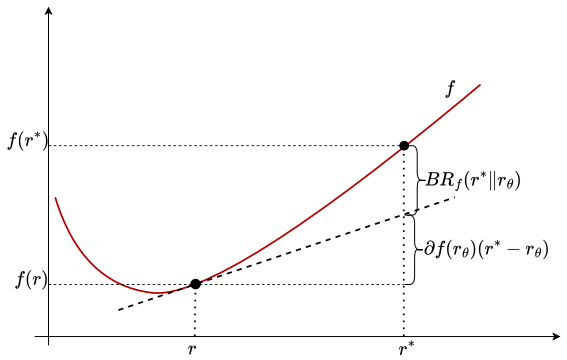

Figure 4: Illustration of DRE with Bregman divergence.

Several kernel functions for Bregman divergence have been proposed in the past discourse, such as LSIF (Kanamori et al., 2009), UKL (Nguyen et al., 2010), BCE (Hastie, 2009), and Basu's power divergence (Basu et al., 1998). We list the details of those functions below.

LSIF

$$f(r) = \frac{1}{2}(r-1)^2. \tag{18}$$

Bregman divergence (BR) defined in Equation 3 is reduced to the squared (SQ) distance:

$$SQ'(r^*\|r_\theta) = \frac{1}{2}(r^* - r)^2. \tag{19}$$

UKL

$$f(r) = r\log r - r. \tag{20}$$

BR is reduced to the unnormalized Kullback–Leibler (UKL) divergence:

$$UKL'(r^*\|r_\theta) = r^* \log \frac{r^*}{r} - r^* + r. \tag{21}$$

BCE/BKL

$$f(r) = r\log r - (r+1)\log(r+1). \tag{22}$$

BR is reduced to the binary Kullback–Leibler (BKL) divergence:

$$BKL'(r^*\|r_\theta) = (1+r^*)\log\frac{1+r}{1+r^*} + r^*\log\frac{r}{r^*}. \tag{23}$$

Basu's power For $\alpha > 0$

$$f(r) = \frac{r^{1+\alpha} - r}{\alpha}. \tag{24}$$

Then BR is reduced to the BA divergence:

$$BA'_\alpha(r^*\|r_\theta) = r^\alpha(r - r^*) - \frac{r^* r^\alpha - (r^*)^{1+\alpha}}{\alpha}. \tag{25}$$

B Experiment Setup

We mainly select `Qwen2.5-7B-Instruct` (Team et al., 2024) for experiment. For a comprehensive comparison, we also selected `Llama2-7B-Chat` (Touvron et al., 2023). Our model is fully fine-tuned (not PEFT) in two stages: 3 epochs of SFT followed by 1 epoch of alignment training, optimizing with AdamW (Kingma and Ba, 2014). For SFT, initial learning rate is 1×10^{-6} for Alfworld, Sciworld and 2×10^{-5} for HotpotQA. For alignment, initial learning rate is 7×10^{-7} for Alfworld and 1×10^{-6} for Sciworld, HotpotQA. Training uses a batch size of 4. In the alignment stage, policy sampling uses temperature 1; Inference testing uses temperature 0. Contrastive pairs are filtered by score reward with threshold $\tau = 0.4$. The pivot value in Equation 10 is 0.9. The lower bound and upper bound of the turn weight in Equation 10 are 0.2 and 1.0.

C Trajectory Sample

Fig. 5 shows how different turns in a trajectory vary in their importance and completeness of training.

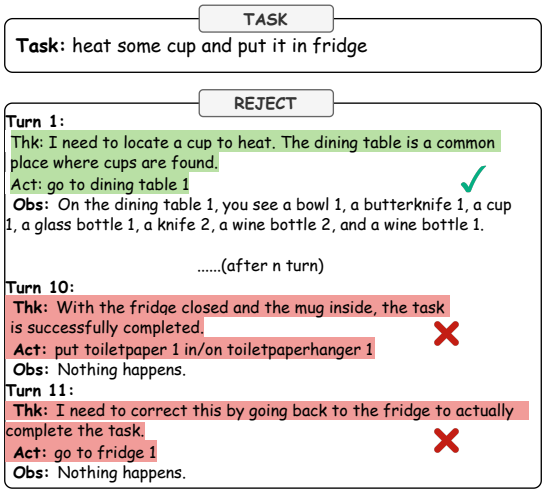

Figure 5: Case analysis across different turns.

Fig. 6 and 7 show how the key turn improved after alignment plays an important role for the task.

TASK

Task: Put a hot mug in coffeemachine

Before tlm-DRE

Turn 1: You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 3, a countertop 2, a countertop 1, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a shelf 3, a shelf 2, a shelf 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 hot mug in coffeemachine.
Thought: I need to put a hot mug in the coffee machine. First, I should find a mug. The fridge is a common place where mugs are stored to keep them cool or for storage.
Action : Go to fridge 1.

Turn 2: The fridge 1 is closed.
Thought: The fridge is closed. I need to open it to see if there's a mug inside.
Action : Open fridge 1.

Turn 3: You open the fridge 1. The fridge 1 is open. In it, you see a bowl 1, a egg 2, a egg 1, a mug 1, a plate 1, a potato 1, and a tomato 1.
Thought: The fridge is now open, and I see a mug inside. I should take it out to use it for the coffee machine.
Action : take mug 1 from fridge 1.

Turn 4: You pick up the mug 1 from the fridge 1.
Thought: I have the mug, but it needs to be hot. I can use the microwave to heat it up. I should go to the microwave to heat the mug.
Action : Go to microwave 1.

Turn 5: The microwave 1 is closed.
Thought: The microwave is closed. I need to open it to see if it's available for use.
Action : Open microwave 1.

Turn 6: You open the microwave 1. The microwave 1 is open. In it, you see a cup 1.
Thought: There's already a cup inside the microwave. I should close the microwave without using it since I can't heat the mug with something else inside.
Action : Close microwave 1.

Turn 7: You close the microwave 1.
Thought: The mug is still in my hand, and I need to put it in the coffee machine. I should go to the coffee machine.
Action : Go to coffeemachine 1.

Turn 8: On the coffeemachine 1, you see nothing.
Thought: I'm at the coffee machine, and it's time to place the mug in or on the coffee machine to complete the task.
Action : Put mug 1 in/on coffeemachine 1.

Turn 9: You put the mug 1 in/on the coffeemachine 1.
Thought: The mug is now in the coffee machine, and I can proceed to make the coffee.
Action : Make coffee 1.

Turn 10: Nothing happens.
Thought: There seems to be an issue with the coffee machine. I should check the coffee machine to see if it's functioning properly.
Action : Check coffeemachine 1.(Fail)
...

Figure 6: Trajectory sample before **tlm-DRE**.

Turn 1: You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 3, a countertop 2, a countertop 1, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a shelf 3, a shelf 2, a shelf 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 hot mug in coffeemachine.
Thought: I need to put a hot mug in the coffee machine. First, I should find a mug. The fridge is a common place where mugs are stored to keep them cool or for storage.
Action : Go to fridge 1.

Turn 2: The fridge 1 is closed.
Thought: The fridge is closed. I need to open it to see if there's a mug inside.
Action : Open fridge 1.

Turn 3: You open the fridge 1. The fridge 1 is open. In it, you see a bowl 1, a egg 2, a egg 1, a mug 1, a plate 1, a potato 1, and a tomato 1.
Thought: The fridge is now open, and I see a mug inside. I should take it out to use it for the coffee machine.
Action : take mug 1 from fridge 1.

Turn 4: You pick up the mug 1 from the fridge 1.
Thought: I have the mug, but it needs to be hot. I can use the microwave to heat it up. I should go to the microwave.
Action : Go to microwave 1.

Turn 5: The microwave 1 is closed.
Thought: The microwave is closed. I need to open it to see if it's available for use.
Action : Open microwave 1.

Turn 6: You open the microwave 1. The microwave 1 is open. In it, you see a cup 1.
Thought: There's already a cup inside the microwave. I should close the microwave without using it since I can't heat the mug with the cup inside.
Action : Close microwave 1.

Turn 7: You close the microwave 1.
Thought: The mug needs to be hot, but I haven't heated it yet. I should have removed the cup from the microwave before attempting to heat the mug. I need to correct this mistake.
Action : Heat mug 1 with microwave 1.

Turn 8: You heat the mug 1 using the microwave 1.
Thought: Now that the mug is hot, I can put it in the coffee machine. I should go to the coffee machine.
Action : Go to coffeemachine 1.

Turn 9: On the coffeemachine 1, you see nothing.
Thought: I'm at the coffee machine with the hot mug. It's time to place the mug in the coffee machine to complete the task.
Action : Put mug 1 in/on coffeemachine 1.

Turn 10: You put the mug 1 in/on the coffeemachine 1.(Success)

Figure 7: Trajectory sample after **tlm-DRE**.