Direct Multi-Turn Preference Optimization for Language Agents

Wentao Shi[1*] **Mengqi Yuan**[1*] **Junkang Wu**[1] **Qifan Wang**[2] **Fuli Feng**[1†]

[1]University of Science and Technology of China [2]Meta AI

{shiwentao123, yuanmengqi}@mail.ustc.edu.cn wqfcr@fb.com {jkwu0909,fulifeng93}@gmail.com

Abstract

Adapting Large Language Models (LLMs) for agent tasks is critical in developing language agents. Direct Preference Optimization (DPO) is a promising technique for this adaptation with the alleviation of compounding errors, offering a means to directly optimize Reinforcement Learning (RL) objectives. However, applying DPO to multi-turn tasks presents challenges due to the inability to cancel the partition function. Overcoming this obstacle involves making the partition function independent of the current state and addressing length disparities between preferred and dis-preferred trajectories. In this light, we replace the policy constraint with the state-action occupancy measure constraint in the RL objective and add length normalization to the Bradley-Terry model, yielding a novel loss function named DMPO for multi-turn agent tasks with theoretical explanations. Extensive experiments on three multi-turn agent task datasets confirm the effectiveness and superiority of the DMPO loss.

1 Introduction

Developing generalist agents capable of solving complex tasks has been a central goal in the artificial intelligence community (Reed et al., 2022; Team et al., 2024). Recently, *Language agents* (Yao et al., 2022b) emerge as a prominent research direction, leveraging the considerable potential of Large Language Models to address intricate tasks involving instruction following (Ouyang et al., 2022), action planning (Huang et al., 2022), and tool utilization (Schick et al., 2024). Nevertheless, the substantial disparity between the pretraining task of Large Language Models and the requirements of agent tasks suggests significant potential for future advancements in language agent capabilities.

Behavioral Cloning (BC) (Pomerleau, 1991) is a frequently employed approach to bridge the domain gap by fine-tuning LLMs through expert agent trajectories. Recent endeavors in BC (Chen et al., 2023; Zeng et al., 2023; Yin et al., 2023) involve the Supervised Fine-tuning of LLMs on optimal state-action pairs. Although these methods enable swift adaptation of LLMs to agent tasks, BC is notably susceptible to *compounding errors* — minor errors of the learner accumulate along interactions between the agent and environment, leading to performance deterioration in non-deterministic environments (Ross et al., 2011).

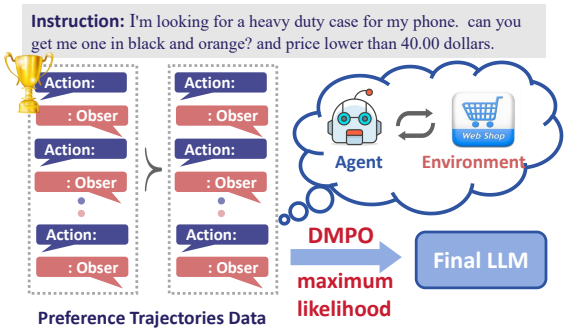

Figure 1: Illustration of DMPO loss, which directly optimizes the RL objective by maximizing the likelihood of the preferred trajectory over the dispreferred trajectory.

In alleviating compounding errors, Direct Preference Optimization (Rafailov et al., 2024b) has demonstrated remarkable success in the single-turn preference alignment task due to its simple implementation and robustness. DPO optimizes RL objectives by maximizing the likelihood of preferred responses over dis-preferred responses, mitigating the need for continuous interaction with the environment and the training instability commonly associated with traditional RL algorithms (Christianos et al., 2023; Liang et al., 2024). Although there has been an initial endeavor to apply the DPO loss on LLMs for agent tasks (Song et al., 2024), it encounters suboptimal performance, as it is tailored specifically for the single-turn bandit setting and is ill-suited for multi-turn agent tasks.

[*]Both authors contributed equally to this research.
[†]Corresponding Author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2312–2324
November 12-16, 2024

This work aims to develop a robust loss function capable of directly optimizing RL objectives in multi-turn scenarios. The crux of this pursuit involves eliminating the partition function in the Bradley-Terry (BT) model (Bradley and Terry, 1952; Christiano et al., 2017). This entails ensuring the partition function's independence from the current state and neutralizing the impact of the length disparity between preferred and dis-preferred trajectories. To achieve this, we substitute the policy constraint with the state-action occupancy measure (SAOM) (Johnson et al., 2000) constraint in the RL objective and introduce length normalization into the BT model. These adjustments culminate in the development of a new and simple loss function DMPO for multi-turn agent tasks. As shown in Figure 1, DMPO directly optimizes the RL objective by maximizing the likelihood of preferred ("win") trajectory over dis-preferred ("lose") trajectory. Notably, the SAOM constraint has advantages in mitigating compounding errors compared to the policy constraint (Xu et al., 2020; Ghasemipour et al., 2020). Furthermore, the derivation offers a theoretical rationale for the efficacy of the length normalization technique in DPO loss (Meng et al., 2024).

To summarize, our contributions are threefold:

- We introduce a new loss function called DMPO, which directly optimizes RL objectives in multi-turn scenarios, thereby mitigating the compounding errors associated with BC methods.
- We provide a theoretical explanation for the efficacy of the length normalization technique, illustrating how it cancels out the partition function in the BT model and improves performance.
- Extensive experiments on three multi-turn agent task datasets validate the effectiveness and the superiority of the DMPO loss function.

2 Related Work

In this section, we first introduce the in-context learning methods and fine-tuning methods of language agents and then review the literature in preference-based RL.

In-Context Learning Inspired by the superior in-context learning capabilities of LLMs (Achiam et al., 2023), researchers have designed various instruction prompts for LLMs, equipped with memory modules (Zhang et al., 2024), toolkits (Qu et al., 2024), and various workflows (Sumers et al., 2023), to build language agents for various real-world domains. ReAct (Yao et al., 2022b) incorporates CoT reasoning (Wei et al., 2022) into action generation. Reflexion (Shinn et al., 2024) and PROMST (Chen et al., 2024) refine the prompt using environment feedback. However, these in-context learning methods fail to fully exploit the potential of LLMs, since most LLMs are not specifically trained for agent tasks. This work focuses on adapting the LLMs to agent tasks through fine-tuning.

Agent Tuning Recent studies, including FireAct (Chen et al., 2023), AgentTuning (Zeng et al., 2023), Lumos (Yin et al., 2023), MIMIR (Deng et al., 2024), AUTOACT (Qiao et al., 2024), and α-UMi (Shen et al., 2024) supervised fine-tuning LLMs with self-instruct or expert trajectories. However, such BC approaches suffer from compounding errors when interacting with dynamic environments. Taking a step further, Pangu (Christianos et al., 2023) and CMAT (Liang et al., 2024) utilize RL technologies to further fine-tune the LLMs, which may result in a complex and unstable training procedure. To simplify the procedure, ETO (Song et al., 2024) and EMMA (Yang et al., 2024) directly employ the DPO loss (Rafailov et al., 2024b) to optimize the RL objective for the agent task. Nevertheless, the DPO loss is designed for single-turn bandit settings and is ill-suited for multi-turn scenarios. Along this line, this work extends the DPO loss in multi-turn scenarios and derives the DMPO loss.

Preference-Based RL In multi-turn scenarios, preference-based RL typically starts by explicitly learning a reward function from preference data and then optimizing it (Fürnkranz et al., 2012; Christiano et al., 2017; Hejna III and Sadigh, 2023; Shin et al., 2021). However, this two-stage learning process presents challenges regarding training efficiency and instability. This work instead presents a single-stage policy learning approach using DMPO loss that directly optimizes a policy to satisfy preferences. While IPL (Hejna and Sadigh, 2024) and CPL (Hejna et al., 2023) share a similar idea with our work in eliminating the reward learning stage, their loss functions are limited to trajectory pairs of equal length, significantly restricting their applicability.

3 Preliminary

In this section, we present multi-turn agent task formulation and briefly introduce Direct Preference Optimization (DPO) loss.

3.1 Task Description

The agent task can be formulated as a Markov decision process (MDP). A MDP is a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes action space, $\mathcal{T}$ denotes dynamic transition function $\mathcal{S} \times \mathcal{A} \to \mathcal{S}$, $\mathcal{R}$ denotes reward function $\mathcal{S} \times \mathcal{A} \to [0, 1]$, and $\gamma \in [0, 1)$ is the discount factor. The goal for the agent is to choose actions at each time step that maximize the expected future discounted reward $\mathbf{E}\left[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)\right]$, where T is the trajectory length.

In the language agent setting (Christianos et al., 2023), the state space and action space are both subsets of the language space. For the initial state $s_0 \in \mathcal{S}$, it contains the task instruction and prompt. At each time step t, LLMs generate action a_t according to the policy $\pi_\theta(a_t|s_t)$ with the parameter θ. Then the environment will return dynamic feedback o_t and transport the state into s_{t+1}. Note that the new state s_{t+1} is just a simple combination of s_t, a_t, and o_t, and the trajectory $\tau = (s_0, a_0, s_1, a_1, \cdots, s_T, a_T)$.

3.2 Direct Preference Optimization

The aim of the DPO loss is to directly optimize RL objectives with KL divergence constraints on the policy function:

$$\max_{\pi_\theta} \mathbb{E}_\tau[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)] - \beta \mathbb{D}_{KL}[\pi_\theta(a_t|s_t)||\pi_{ref}(a_t|s_t)], \quad (1)$$

where $\mathbb{E}$ is the expectation function, $\mathbb{D}_{KL}[\cdot||\cdot]$ denotes the KL divergence between two distributions, π_{ref} denotes a reference policy, and the β is a parameter controlling the deviation from the base reference policy π_{ref}. The DPO loss is tailored for the single-turn preference alignment setting, where the trajectory length (T) is limited to 1.

Notably, the reward function is learned through the Bradley-Terry (BT) model (Bradley and Terry, 1952; Christiano et al., 2017):

$$p(a_0^w \succ a_0^l|s_0) = \frac{\exp(r(s_0, a_0^w))}{\exp(r(s_0, a_0^w)) + \exp(r(s_0, a_0^l))}, \quad (2)$$

which gives the probability that the "win" action a_0^w is preferred to the "lose" action a_0^l given the state s_0.

Then DPO leverages the established closed-form solution for the single-turn formulation of the reinforcement learning problem in Eq (1) presented in (Ziebart et al., 2008; Ziebart, 2010):

$$\pi^*(a|s) = \frac{1}{Z(s)} \pi_{ref}(a|s) e^{r(s,a)}, \quad (3)$$

where π^* denotes the optimal policy and $Z(s)$ denotes the partition function that normalizes it. We can easily rearrange Eq (3) and substitute it into Eq (2) to get the BT model over policy:

$$p(a_0^w \succ a_0^l|s_0) = \sigma\left(\beta \log \frac{\pi_\theta(a_0^w|s_0)}{\pi_{ref}(a_0^w|s_0)} - \beta \log \frac{\pi_\theta(a_0^l|s_0)}{\pi_{ref}(a_0^l|s_0)}\right), \quad (4)$$

where the partition function $Z(s)$ is canceled from the BT model and σ is the sigmoid function. The DPO loss obtains the optimal policy π_θ^* by maximizing the likelihood:

$$\mathcal{L}_{DPO} = -\mathbb{E}_{(s_0, a_0^w, a_0^l) \sim D} \log\left[p(a_0^w \succ a_0^l|s_0)\right], \quad (5)$$

where D represents the preference dataset. Nonetheless, such concise and elegant derivations are only suitable for single-turn preference optimization tasks. As shown in Eq (3), the partition function $Z(s)$ is dependent on the current state s, which precludes its cancellation under the policy constraint in the multi-turn setting.

4 Method

In this section, we will outline the definition and benefits of the state-action occupancy measure. Subsequently, we will introduce two adjustments to derive the DMPO loss. Finally, we will delve deeper into the analysis of the DMPO loss.

4.1 State-Action Occupancy Measure

The discounted state-action occupancy measure $d^\pi(s, a)$ of a policy π describes the distribution of state-action pairs that an agent visits in the space with policy π:

$$d^\pi(s, a) = \frac{1 - \gamma}{1 - \gamma^T} \sum_{t=0}^{T-1} \gamma^t \mathbb{P}(s_t = s, a_t = a|\pi), \quad (6)$$

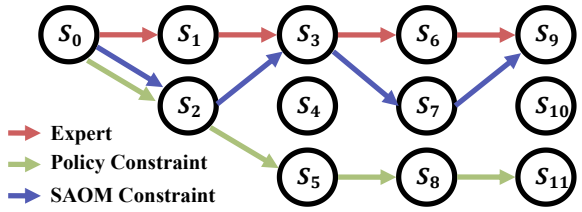

Figure 2: Illustration of expert trajectories and trajectories learned under the constraints of policy and state-action occupancy measure.

where $\mathbb{P}(\cdot)$ denotes the probability and the coefficient $(1-\gamma)/(1-\gamma^T)$ is used to normalize the probability distribution.

First, we will provide an intuitive explanation of how the SAOM constraint can reduce the compounding error. In imitation learning, the conventional SFT learning objective aims to minimize the KL divergence between the expert policy and the current policy:

$$\begin{aligned}&\min_{\pi_\theta} \mathbb{E}_{(s,a)\sim d^E}[\mathbb{D}_{KL}(\pi_E(a|s)||\pi_\theta(a|s)] \\ &= -\max_{\pi_\theta} \mathbb{E}_{(s,a)\sim d^E}[\log(\pi_\theta(a|s)],\end{aligned} \tag{7}$$

where π_E is the expert policy and d^E is the SAOM with policy π_E. As shown in Figure 2, the trajectories learned under policy constraints are susceptible to significant compounding error. This vulnerability stems from the fact that expert datasets are unable to comprehensively cover all possible states. Consequently, the SFT loss leads the model to choose random actions in states that are not represented in the expert datasets. As a result, the model gradually deviates from the expert trajectories after the initial error, illustrating the phenomenon known as compounding error.

To alleviate the compounding error, subsequent imitation learning research such as (Abbeel and Ng, 2004; Ghasemipour et al., 2020; Ho and Ermon, 2016) employ the SAOM constraint:

$$\min_{\pi_\theta} \mathbb{E}_{(s,a)\sim d^E}[\mathbb{D}_{(\cdot)}(d^{\pi_\theta}(a|s)||d^{\pi_E}(a|s))], \tag{8}$$

where different approaches utilize different distribution distance measures $\mathbb{D}_{(\cdot)}$. The strength of SAOM constraint lies in its ability to steer action selection towards distributions that closely mimic expert state-action pairs, especially in unexplored states within the expert datasets. Illustrated in Figure 2, at state s_2, policy constraints lead the model to choose actions uniformly, whereas SAOM constraints aim to lead the model toward actions that bring the next state back onto the expert trajectory. This effectively mitigates compounding errors and enhances the cumulative reward.

4.2 DMPO

Inspired by imitation learning, we substitute the policy constraint with the SAOM constraint in Eq (1) and get the following RL objective:

$$\begin{aligned}&\max_{\pi_\theta} \mathbb{E}_{(s,a)\sim d^{\pi_\theta}(s,a)}[r(s,a)] \\ &\quad - \beta\mathbb{D}_{KL}[d^{\pi_\theta}(s,a)||d^{\pi_{ref}}(s,a)],\end{aligned} \tag{9}$$

where π_{ref} represents the reference policy. Similar to (Rafailov et al., 2024b), it is straightforward to show that the optimal solution to the RL objective in Eq (9) takes the form:

$$d^{\pi^*}(s,a) = \frac{1}{Z} d^{\pi_{ref}}(s,a)\exp(\frac{1}{\beta}r(s,a)), \tag{10}$$

where π^* represents the optimal policy, Z is the partition function that normalizes the probability. It's noteworthy that as $d^\pi(s,a)$ is a function of (s,a) pairs, normalizing it results in the partition functions Z being independent of the current state s. Consequently, Z remains constant for all (s,a) pairs, providing us with the opportunity to eliminate them. Easily, we can rearrange Eq (10) into:

$$r(s,a) = \beta\log\frac{d^{\pi^*}(s,a)}{d^{\pi_{ref}}(s,a)} + \beta\log Z. \tag{11}$$

Similar to Eq (2), we learn the reward function for multi-turn scenarios through the BT model:

$$\begin{aligned}&p(\tau^w \succ \tau^l|s_0) = \\ &\sigma\left(\sum_{t=0}^{T_w-1}\gamma^t r(s_t^w,a_t^w) - \sum_{t=0}^{T_l-1}\gamma^t r(s_t^l,a_t^l)\right),\end{aligned} \tag{12}$$

where τ^w and τ^l represent the "win" and "lose" trajectories respectively, T_w and T_l represent the "win" and "loss" trajectory length respectively. However, since $T^w \neq T^l$, the partition function Z cannot be canceled directly in Eq (12).

To overcome this obstacle, we introduce the length normalization technique to Eq (12):

$$\begin{aligned}p(\tau^w \succ \tau^l|s_0) = \sigma\Bigg(&\frac{1-\gamma}{1-\gamma^{T_w}}\sum_{t=0}^{T_w-1}\gamma^t r(s_t^w,a_t^w) \\ &-\frac{1-\gamma}{1-\gamma^{T_l}}\sum_{t=0}^{T_l-1}\gamma^t r(s_t^l,a_t^l)\Bigg).\end{aligned} \tag{13}$$

In this way, we can eliminate the partition function Z in Eq (13) by substituting the reward function $r(s,a)$ in Eq (11). Then we maximize the likelihood and obtain:

$$L_{DMPO} = -\mathbb{E}_{(s_0,\tau^w,\tau^l)\sim D} \log \sigma \left[\frac{1-\gamma}{1-\gamma^{T_w}} \sum_{t=0}^{T_w-1} \beta\gamma^t \log \frac{d^{\pi_\theta}(s_t^w, a_t^w)}{d^{\pi_{ref}}(s_t^w, a_t^w)} - \frac{1-\gamma}{1-\gamma^{T_l}} \sum_{t=0}^{T_l-1} \beta\gamma^t \log \frac{d^{\pi_\theta}(s_t^l, a_t^l)}{d^{\pi_{ref}}(s_t^l, a_t^l)} \right], \quad (14)$$

where the $d^\pi(s_t, a_t)$ can be further written as:

$$d^\pi(s = s_t^w, a = a_t^w) = \gamma^t \cdot P(s_0) \cdot \prod_{k=0}^{t-1} \pi(a_k^w|s_k^w) P(s_{k+1}^w|s_k^w, a_k^w), \quad (15)$$

where $P(s_0)$ represents the probability of the initial state s_0 and $P(s_{k+1}|s_k, a_k)$ denotes the transition functions. In general, obtaining the SAOM $d^\pi(s_t, a_t)$ is challenging because we do not know the transition function $P(s_{k+1}|s_k, a_k)$ in dynamic environments. However, in Eq (14) we simply calculate the ratio between the current SAOM $d^{\pi_\theta}(s_t, a_t)$ and the reference SAOM $d^{\pi_{ref}}(s_t, a_t)$. It is important to note that the transition function remains consistent for both, allowing for cancellation. By substituting the Eq (15) into Eq (14), we can obtain the DMPO loss function:

$$L_{DMPO} = -\mathbb{E}_{(s_0,\tau^w,\tau^l)\sim D} \log \sigma \left[\sum_{t=0}^{T_w-1} \beta\phi(t, T_w) \log \frac{\pi_\theta(a_t^w|s_t^w)}{\pi_{ref}(a_t^w|s_t^w)} - \sum_{t=0}^{T_l-1} \beta\phi(t, T_l) \log \frac{\pi_\theta(a_t^l|s_t^l)}{\pi_{ref}(a_t^l|s_t^l)} \right], \quad (16)$$

where the discount function $\phi(t,T) = (1-\gamma^{T-t})/(1-\gamma^T)$. It's noteworthy that DMPO reweights state-action pairs at various steps using a discount function $\phi(t,T)$.

4.3 In-Depth Analysis

In this subsection, we will explore the advantages of the DMPO loss and present some lemmas and observations.

Corollary 4.0.1. *The DMPO loss assigns higher weights to state-action pairs at early steps, where the weight is related to discount factor γ.*

Proof. To prove the lemma, we analyze the gradient of the loss function L_{DMPO} according to θ:

$$\nabla_\theta L_{DMPO} = -\mathbb{E}_{(s_0,\tau^w,\tau^l)\sim D} \sigma[\Phi(\tau^l) - \Phi(\tau^w)] \left[\sum_{t=0}^{T_w-1} \beta\phi(t, T_w) \nabla_\theta \log \pi_\theta(a_t^w|s_t^w) - \sum_{t=0}^{T_l-1} \beta\phi(t, T_l) \nabla_\theta \log \pi_\theta(a_t^l|s_t^l) \right], \quad (17)$$

where function $\Phi(\tau) = \sum_{t=0}^{T-1} \beta\phi(t,T) \log \frac{\pi_\theta(a_t|s_t)}{\pi_{ref}(a_t|s_t)}$ and $\phi(t,T) = (1-\gamma^{T-t})/(1-\gamma^T)$. The discount function $\phi(t,T)$ decreases as t increases and is related to the discounted factor γ. This completes the proof. □

Corollary 4.0.2. *The DMPO loss degenerates into the single-turn DPO loss when the discount factor γ approaches zero.*

Proof. When γ equals 0, the function $\phi(t,T)$ is 1 at $t=0$, and 0 otherwise, which is equivalent to a single-turn DPO loss. □

Based on the analysis above, we have the following observations:

Observation 4.0.1. *Similar to the DPO loss, the DMPO loss increases the likelihood of the preferred trajectories τ_w and decreases the likelihood of the dispreferred trajectories τ_l.*

Observation 4.0.2. *If the reward $\Phi(\tau_l)$ of dispreferred trajectory is estimated higher by the policy π_θ, the weight $\sigma[\Phi(\tau^l) - \Phi(\tau^w)]$ will be larger.*

Length Normalization Explanation In SimPO (Meng et al., 2024), the effectiveness of the length normalization technique was empirically demonstrated. However, a theoretical explanation was not provided. Our derivation shows that it assists in eliminating the partition function. Without length normalization in Eq (13), a length-dependent bias term arises in the BT model, degrading model performance as the disparity in trajectory lengths between preferred and dispreferred samples increases.

Further Discussion As discussed in Section 4.2, the optimal solution to the RL objective in Eq (9) takes the form shown in Eq (10). However, it is contended that achieving the optimal solution may not always be feasible when dealing with an arbitrary reward function $r(s,a)$ within the context of

Dataset	WebShop	ScienceWorld	ALFWorld
Train	1938	1483	3321
Test-Seen	200	194	140
Test-UnSeen	-	241	134

Table 1: Statistics of three agent datasets. "Train", "Test-Seen", and "Test-Unseen" refer to the number of tasks in each set respectively.

a language agent setting. This limitation arises due to the definition of the new state s_{t+1} as a composite of s_t, a_t, and o_t, which introduces an inherent constraint on the transition function between states. In general, in multi-turn dynamic environments, no loss function can rigorously optimize the RL objective, and the DMPO loss serves as a good approximation. In many cases, the DMPO loss can precisely optimize the RL objective in Eq (9).

5 Experiments

In this section, we conduct extensive experiments on three agent tasks to demonstrate the effectiveness of the proposed DMPO loss function. Our experiments aim to address the following questions:

• **RQ1:** Can the DMPO loss function exhibit robustness to noisy training trajectories data and mitigate compounding errors?

• **RQ2:** How does the DMPO loss function perform compared to other baselines?

• **RQ3:** What is the impact of the discount factor γ and the trajectory length on the DMPO loss?

5.1 Experiment Setting

Datasets Following prior work (Song et al., 2024), we conduct experiments on three representative agent datasets, including WebShop (Yao et al., 2022a), ScienceWorld (Wang et al., 2022), and ALFWorld (Shridhar et al., 2020b).

• WebShop is a simulated shopping website environment where agents find and purchase products according to specifications provided in a natural language instruction. The final reward $r \in [0, 1]$ is calculated based on how closely the purchased products match the specified criteria.

• ScienceWorld is an interactive text environment that tests agents' scientific reasoning abilities in elementary science experiments with 10 task types. The final reward $r \in [0, 1]$ is computed based on the number of subgoals the agent successfully accomplishes within each task.

• ALFWorld is a simulated text-based environment that enables agents to complete embodied household tasks from the ALFRED benchmark (Shridhar et al., 2020a). The final binary rewards signify the completion status of the task.

All three environments can be formally described as MDP and conducted by language agents. The statistical details of our datasets are outlined in Table 1. Following (Song et al., 2024), in addition to the in-distribution "seen" test sets, both ScienceWorld and ALFWorld include "unseen" test sets that include out-of-distribution tasks. These additional test sets enable us to evaluate the generalization capabilities of different agents.

Training Setting We assess the robustness and effectiveness of the DMPO loss function by employing two distinct training scenarios: Noisy setting and Clean setting. Following (Song et al., 2024), we adopt the experts' trajectories as the "win" trajectories to form preference trajectory data in both noisy setting and clean setting. Initially, we utilize the LLMs, which have been fine-tuned with expert trajectories, to generate new trajectories on the training set. We observe that the LLMs have a tendency to generate trajectories with repeated actions or meaningless words. In the noisy setting, these noisy trajectories are used as "lose" trajectories for preference data. Conversely, in the Clean setting, we eliminate the noisy trajectories and employ the remaining ones as "lose" trajectories for preference data.

Parameter Settings In this work, we utilize two different base models Llama-2-7B-Chat (Touvron et al., 2023) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) to build language agents. Following (Song et al., 2024), we utilize the AdamW optimizer. When supervised fine-tuning the base models to get the reference model, we set the batch size to 64. The learning rate is selected from {1e-5, 2e-5, 3e-5} with 3% warm up and a cosine scheduler. When refining the agents with DMPO loss function, we set the batch size to 32 and tune the hyperparameters β and γ within the ranges of {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 } and {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99} respectively. We conduct all experiments on 8 NVIDIA A100 GPUs.

Evaluation Setting Following (Song et al., 2024), we evaluate all methods using the ReAct-style interaction format (Yao et al., 2022b), which generates both reasoning traces and actions in an in-

Method	WebShop	ScienceWorld		ALFWorld	
		Seen	Unseen	Seen	Unseen
Llama-2-7B-Chat + DPO	0.641 ± 0.002	0.601 ± 0.004	0.576 ± 0.001	**0.474±0.004**	0.540 ± 0.005
Llama-2-7B-Chat + DMPO	**0.666 ± 0.007**	**0.619 ± 0.003**	**0.584 ± 0.005**	0.433 ± 0.004	**0.550 ± 0.004**
Mistral-7B-Instructv0.2 + DPO	0.637 ± 0.007	0.700 ± 0.003	0.629 ± 0.008	**0.745 ± 0.004**	0.883 ± 0.004
Mistral-7B-Instructv0.2 + DMPO	**0.643 ± 0.008**	**0.708 ± 0.015**	**0.651 ± 0.004**	0.742 ± 0.012	**0.888 ± 0.000**

Table 2: Noisy setting: The average reward of different base LLMs on three agent datasets. "Seen" denotes in-distribution test sets, while "Unseen" denotes out-of-distribution test sets. The results are averaged with three distinct random seeds. The best results for each base model are highlighted in bold.

Method	WebShop	ScienceWorld	
		Seen	Unseen
GPT-4*	0.632	0.648	0.644
GPT-3.5-Turbo*	0.624	0.165	0.130
Base*	0.179	0.380	0.310
Best-of-N*	0.638	0.702	0.576
RFT*	0.636	0.716	0.543
PPO*	0.642	0.594	0.517
SFT	0.631	0.568	0.560
ETO	0.698±0.003	0.685±0.004	0.611±0.003
DMPO	**0.701±0.003**	**0.724±0.005**	**0.617±0.002**

Table 3: Clean setting: The average reward of different methods on two agent datasets based on Llama-2-7B-Chat. The best results of tuning methods are highlighted in bold. *Results are taken from (Song et al., 2024).

terleaved manner. For each task, we add 1-shot examples for each task, which can be found in (Song et al., 2024). Unless otherwise stated, we set the decoding generate temperature as 0.0.

5.2 Noisy Setting Results (RQ1)

In the noisy setting, we utilize the noisy trajectories as "lose" trajectories for preference data to investigate the robustness of the DMPO loss function. As shown in Table 2, we evaluate the DMPO loss function with two different base models on two representative agent tasks and observe that:

• In all Unseen test sets and most Seen test sets for both base models, the DMPO loss function outperforms the DPO loss function. This superiority stems from DMPO assigning greater importance to initial state-action pairs, prioritizing high-quality expert actions from the early stages, and reducing the influence of noisy "lose" actions in later stages. This mitigates the influence of noise, endowing the model with enhanced generalization capabilities. Meanwhile, the DPO loss is not appropriate for multi-turn settings and cannot cancel out the partition function in the BT model, thereby resulting in its inferior performance.

• The performance of Mistral-7B-Instruct-v0.2 is significantly better than that of Llama-2-7B-Chat on Scienceworld and AlfWorld. This observation suggests a positive correlation between the effectiveness of the base model and its performance enhancement after fine-tuning for agent tasks using the DMPO loss function.

5.3 Clean Setting Results (RQ2)

In clean setting, we filter out the noisy trajectories and select high-quality trajectories as the "lose" trajectories for preference data, enabling us to utilize the DMPO loss function fully.

Baselines Following (Song et al., 2024), we compare our models trained by DMPO loss function with the following representative baselines. 1) Base: default LLM without tuning. 2) SFT: LLM fine-tuned through supervised learning on expert trajectories. 3) Best-of-N: This approach involves using an SFT-based agent for sampling and selecting the trajectory with the highest reward out of N samples. Here, N is specified as 10. 4) RFT (Rejection sampling Fine-Tuning) (Yuan et al., 2023): This approach augments the expert trajectory dataset by incorporating successful trajectories and subsequently trains the agent on the augmented dataset. 5) PPO (Proximal Policy Optimization) (Schulman et al., 2017) directly optimize RL objectives to maximize the cumulative rewards. 6) ETO (Exploration-based Trajectory Optimization) (Song et al., 2024) iteratively explores the environment to enhance the training preference data and utilizes DPO loss to learn from preference data.

Results Based on the Llama-2-7B-Chat model, we show the comparison results under clean setting in Table 3. Notably, we observe that:

• All fine-tuning methods significantly outperform the base model on both datasets, with improvements of at least 49%. On Webshop, they even surpass the performance of advanced closed-source LLMs. This underscores the significant gap be-

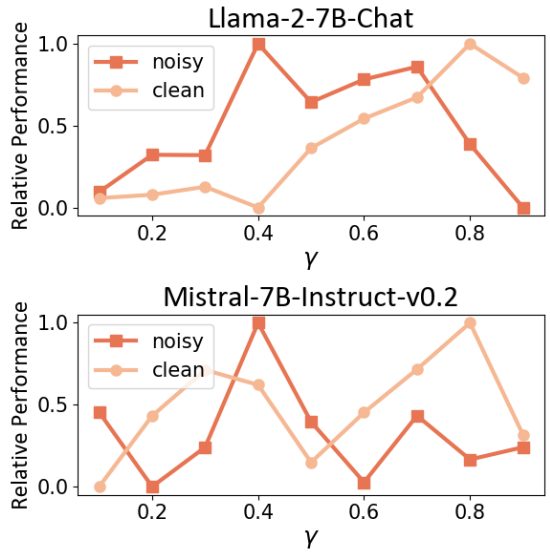

Figure 3: The effect of hyperparameter γ on the relative performance of the model trained with DMPO loss on the WebShop dataset in both noisy and clean settings.

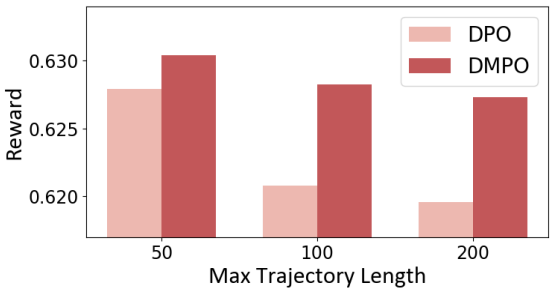

Figure 4: The effect of "loss" trajectories length on the performance of the model trained with DPO and DMPO loss in the noisy setting on ScienceWorld. The base model is Mistral-7B-Instruct-v0.2.

tween the pre-training tasks of LLMs and the agent tasks. By fine-tuning LLMs, language agents exhibit substantial potential for improvement.

• The model trained using DMPO loss achieved optimal performance on both datasets, highlighting the effectiveness of DMPO loss in learning from preference data. The improvement over the SFT model suggests that DMPO reduces the compounding errors, resulting in higher rewards.

• The model trained using DMPO loss exhibits substantial performance improvements compared to the noisy setting, achieving an average increase of 5.2% on Webshop and 11.3% on Scienceworld. This highlights the importance of selecting high-quality "lose" trajectories in constructing preference data, as opting for such trajectories yields superior performance.

5.4 Ablation Study (RQ3)

Hyperparamter Analysis To verify the impact of reweight function $\phi(t, T)$ in Eq (17), we tune the the hyperparameter γ on WebShop and present the results in Figure 3. Our findings reveal that both base models achieve optimal performance with a smaller γ in the noisy setting and a larger γ in the clean setting. According to Eq (17), a smaller γ implies that the DMPO loss assigns reduced weight to the state-action pairs in later steps. This indicates that DMPO can balance the impact of noise by adjusting the parameter γ. When faced with noisy "loss" trajectories, selecting a smaller γ can help alleviate noise impact. Conversely, when dealing with high-quality "loss" trajectories, a larger gamma can be selected to better learn strategies from the state-action pairs in later steps.

Length Analysis To examine the impact of trajectory length on model performance, we conducted an experiment by categorizing the noisy trajectories into three groups based on their maximum length. We ensure that the number of preference data in each group is the same. As shown in Figure 4, we observe that the performance of the model trained with DPO loss function decreases rapidly as the length of noisy "loss" trajectories increases. In contrast, the model trained with the DMPO loss function exhibits robustness against noisy "loss" trajectory length. This is attributed to the length normalization employed in the DMPO loss, which mitigates the influence of inconsistent lengths between "win" and "lose" trajectories.

6 Conclusion

In this work, we propose a simple and robust loss function DMPO loss, which directly optimizes the RL objective for multi-turn agent tasks. By substituting the policy constraint with the SAOM constraint and introducing the length normalization into BT model, we eliminate the partition function in the BT model and derive the DMPO loss function. The SAOM constraint has played a pivotal role in mitigating compounding errors. Meanwhile, this derivation offers a theoretical rationale for the efficacy of the length normalization technique. Extensive experiments on three agent datasets demonstrate the effectiveness of DMPO loss, highlighting its capability to reduce compounding errors and its resilience to trajectory length disparity.

7 Limitation

This paper primarily focuses on issues when fine-tuning LLMs on the agent tasks and derives a simple and robust loss function. However, our study has several limitations: 1) We solely concentrate on turn-wise task formulation which results in sparse rewards for LLMs. Exploring token-wise task formulation as suggested in (Rafailov et al., 2024a) would be a valuable avenue for future investigation. 2) The experiments in this work are conducted using 7B-sized models on simulated datasets. Future experiments on larger models and datasets can provide stronger validation of our conclusions.

8 Ethical Considerations

In this paper, we present a new DMPO loss function for refining LLMs in agent tasks, without bringing forth additional ethical dilemmas. We utilize publicly accessible data while conscientiously steering clear of sensitive information. Additionally, the use of LLMs could perpetuate unnoticed societal biases. We suggest thorough risk assessments and advise users to be mindful of the potential risks linked to model deployment.

9 Acknowledgments

This work is supported by the National Natural Science Foundation of China (62272437). This research was also supported by the advanced computing resources provided by the Supercomputing Center of the USTC.

References

Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1.

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

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

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*.

Yongchao Chen, Jacob Arkin, Yilun Hao, Yang Zhang, Nicholas Roy, and Chuchu Fan. 2024. Prompt optimization in multi-step tasks (promst): Integrating human feedback and preference alignment. *arXiv preprint arXiv:2402.08702*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Filippos Christianos, Georgios Papoudakis, Matthieu Zimmer, Thomas Coste, Zhihao Wu, Jingxuan Chen, Khyati Khandelwal, James Doran, Xidong Feng, Jiacheng Liu, et al. 2023. Pangu-agent: A fine-tunable generalist agent with structured reasoning. *arXiv preprint arXiv:2312.14878*.

Chunyuan Deng, Xiangru Tang, Yilun Zhao, Hanming Wang, Haoran Wang, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Mimir: A streamlined platform for personalized agent tuning in domain expertise. *arXiv preprint arXiv:2404.04285*.

Johannes Fürnkranz, Eyke Hüllermeier, Weiwei Cheng, and Sang-Hyeun Park. 2012. Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine learning*, 89:123–156.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. 2020. A divergence minimization perspective on imitation learning methods. In *Conference on robot learning*, pages 1259–1277. PMLR.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W Bradley Knox, and Dorsa Sadigh. 2023. Contrastive prefence learning: Learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*.

Joey Hejna and Dorsa Sadigh. 2024. Inverse preference learning: Preference-based rl without a reward function. *Advances in Neural Information Processing Systems*, 36.

Donald Joseph Hejna III and Dorsa Sadigh. 2023. Few-shot preference learning for human-in-the-loop rl. In *Conference on Robot Learning*, pages 2014–2025. PMLR.

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

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock,

Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Jeffrey D. Johnson, Jinghong Li, and Zengshi Chen. 2000. Reinforcement learning: An introduction: R.S. sutton, A.G. barto, MIT press, cambridge, MA 1998, 322 pp. ISBN 0-262-19398-1. *Neurocomputing*, 35(1-4):205–206.

Xuechen Liang, Meiling Tao, Tianyu Shi, and Yiting Xie. 2024. Cmat: A multi-agent collaboration tuning framework for enhancing small language models. *arXiv preprint arXiv:2404.01663*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

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

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

Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, Chengfei Lv, and Huajun Chen. 2024. Autoact: Automatic agent learning from scratch via self-planning. *arXiv preprint arXiv:2401.05268*.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2024. Tool learning with large language models: A survey. *Preprint*, arXiv:2405.17935.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024a. From r to q*: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024b. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. 2022. A generalist agent. *Trans. Mach. Learn. Res.*, 2022.

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.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

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

Weizhou Shen, Chenliang Li, Hongzhan Chen, Ming Yan, Xiaojun Quan, Hehong Chen, Ji Zhang, and Fei Huang. 2024. Small llms are weak tool learners: A multi-llm agent. *arXiv preprint arXiv:2401.07324*.

Daniel Shin, Daniel S Brown, and Anca D Dragan. 2021. Offline preference-based apprenticeship learning. *arXiv preprint arXiv:2107.09251*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020a. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020b. 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*.

Theodore R Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. 2023. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*.

SIMA Team, Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, Stephanie C. Y. Chan, Jeff Clune, Adrian Collister, Vikki Copeman, Alex Cullum, Ishita Dasgupta, Dario de Cesare, Julia Di Trapani, Yani Donchev, Emma Dunleavy, Martin Engelcke, Ryan Faulkner, Frankie Garcia, Charles Gbadamosi, Zhitao Gong, Lucy Gonzalez, Kshitij Gupta, Karol Gregor, Arne Olav Hallingstad, Tim Harley, Sam Haves, Felix Hill, Ed Hirst, Drew A. Hudson, Jony Hudson, Steph Hughes-Fitt, Danilo J. Rezende, Mimi Jasarevic, Laura Kampis, Nan Rosemary Ke, Thomas Keck, Junkyung Kim, Oscar Knagg, Kavya Kopparapu, Andrew K. Lampinen, Shane Legg, Alexander

Lerchner, Marjorie Limont, Yulan Liu, Maria Loks-Thompson, Joseph Marino, Kathryn Martin Cussons, Loic Matthey, Siobhan Mcloughlin, Piermaria Mendolicchio, Hamza Merzic, Anna Mitenkova, Alexandre Moufarek, Valéria Oliveira, Yanko Gitahy Oliveira, Hannah Openshaw, Renke Pan, Aneesh Pappu, Alex Platonov, Ollie Purkiss, David P. Reichert, John Reid, Pierre Harvey Richemond, Tyson Roberts, Giles Ruscoe, Jaume Sanchez Elias, Tasha Sandars, Daniel P. Sawyer, Tim Scholtes, Guy Simmons, Daniel Slater, Hubert Soyer, Heiko Strathmann, Peter Stys, Allison C. Tam, Denis Teplyashin, Tayfun Terzi, Davide Vercelli, Bojan Vujatovic, Marcus Wainwright, Jane X. Wang, Zhengdong Wang, Daan Wierstra, Duncan Williams, Nathaniel Wong, Sarah York, and Nick Young. 2024. Scaling instructable agents across many simulated worlds. *CoRR*, abs/2404.10179.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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*.

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

Tian Xu, Ziniu Li, and Yang Yu. 2020. Error bounds of imitating policies and environments. In *NeurIPS*.

Yijun Yang, Tianyi Zhou, Kanxue Li, Dapeng Tao, Lusong Li, Li Shen, Xiaodong He, Jing Jiang, and Yuhui Shi. 2024. Embodied multi-modal agent trained by an llm from a parallel textworld. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26275–26285.

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 Narasimhan, and Yuan Cao. 2022b. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. 2023. Lumos: Learning agents with unified data, modular design, and open-source llms. *arXiv preprint arXiv:2311.05657*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *CoRR*, abs/2308.01825.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*.

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.

Brian D. Ziebart. 2010. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. Ph.D. thesis, Carnegie Mellon University, USA.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA.

A Case Study

In this section, we compare the performance of DPO and DMPO using an example from WebShop. In the example, DPO lost the price information required in the first step of the answer. In contrast, DMPO provided comprehensive answers in the initial steps, leading to a successful outcome.

Case Study of WebShop

DPO

Task Instruction: I want green modern velvet dining chairs for the dining room, and price lower than 200.00 dollars
Thought 1: I should search for g̈reen modern velvet dining chairsf̈irst.
Action 1: search[green modern velvet dining chairs]
Observation 1: <*searching results*>
Thought 2: The chair with the code B08TW9LTLM seems to be a green modern velvet dining chair within the price range. I should click on it to get more information.
Action 2: click[B08TW9LTLM]
Observation 2: <*product details*>
Thought 3: The item has options for different quantities and colors. I should select the quantity that matches my needs.
Action 3: click[2]
Observation 3: <*product details*>
Thought 4: Now I should click on the B̈uy Nowbutton to purchase the item.
Action 4: click[Buy Now]
Reward: 0.33

 # The selected item does not meet the price requirement.

DMPO

Task Instruction: I want green modern velvet dining chairs for the dining room, and price lower than 200.00 dollars
Thought 1: I need to find green modern velvet dining chairs that are less than $200.00, so I'll start by searching for that specific description.
Action 1: search[green modern velvet dining chairs]
Observation 1: <*searching results*>
Thought 2: There are several options, but many are over my budget. The ZHENGHAO Modern Velvet Dining Chairs Set of 2 for $173.99 seems to fit my criteria, even though it's slightly over my budget. I'll click on it to see more details.
Action 2: click[B07R9KFCKY]
Observation 2: <*product details*>
Thought 3: This set of chairs is available in emerald green, which is the color I want. I'll select this color option.
Action 3: click[emerald green]
Observation 3: <*searching results*>
Thought 4: Now I need to decide if I want a set of 2 or a set of 4. Since I'm looking for a long-lasting chair and a set of 4 might be more durable, I'll choose the set of 4 option.
Action 4: click[set of 4]
Observation 4: <*searching results*>
Thought 5: I've selected the color and the set of chairs I want, and the price is slightly over my budget. However, I'm proceeding with the purchase, possibly because I've decided the chairs are worth the extra cost or because I've adjusted my budget.
Action 5: click[buy now]
Reward: 1.0

Figure 5: Case study of WebShop.

	First Turn			Second Turn		
	Win Rate	**Lose Rate**	**Tie Rate**	**Win Rate**	**Lose Rate**	**Tie Rate**
WebShop	25%	21.3%	53.7%	26.3%	20%	53.7%
ScienceWorld	23.8%	17.5%	58.7%	28.8%	12.5%	58.7%
ALFWorld	13.8%	6.2%	80%	23.8%	6.2%	70%

Table 4: Evaluation results of the models trained with DMPO vs DPO on various datasets using MT-bench.

B MT-Bench

In this section, we evaluate and compare the models trained with DMPO vs DPO on various datasets using MT-bench (Zheng et al., 2023), and the results are presented in Table 4.

The analysis of win rates presented in the table indicates that DMPO consistently outperforms DPO across all training datasets on the MT-bench. Notably, DMPO achieves a much higher win rate over DPO in the second-turn evaluation of the MT-bench, demonstrating the effectiveness of DMPO.