LEARNING TO SHAPE THE REWARD FOR AGENTIC REINFORCEMENT LEARNING VIA BILEVEL OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning for agentic systems often relies on trajectory-level rewards, leading to poor credit assignment in multi-turn tasks. A common approach is to introduce turn-level rewards, such as heuristic signals, LLM judges, or learned rewards, to provide dense feedback. However, existing reward learning approaches typically optimize the reward based on trajectories generated by the current policy, while treating the trajectory distribution as independent of the learned reward. In reality, the learned reward determines how the policy is optimized, which in turn affects the trajectories generated by the policy. Ignoring this dependency results in an incomplete gradient for reward learning. This paper proposes Bilevel Agentic Reward Shaping (BARS), a framework that explicitly captures this reward-policy-trajectory dependency through bilevel optimization. The upper level learns a turn-level reward from outcome feedback, while the lower level optimizes the policy under the learned reward. We develop an efficient algorithm that alternates between reward learning and policy updates, and establish a convergence rate of $O(1/\sqrt{K})$. Empirical results demonstrate that BARS achieves higher accuracy and improved training stability compared with baseline methods.

1 INTRODUCTION

Large Language Models (LLMs) are evolving from passive text generators into autonomous agents capable of reasoning, acting, and interacting with complex environments through multi-turn interactions. Recent systems, including search agents (Jin et al., 2025), web navigators (Furuta et al., 2024), and coding assistants (Zhang et al., 2024), demonstrate that augmenting LLMs with external tools (e.g., search engines, databases, and code interpreters) can extend their capabilities beyond pure text-based reasoning. These agentic systems require making sequential and multi-turn decisions, where each turn involves generating reasoning traces, selecting actions, and incorporating environmental feedback into their subsequent states.

Reinforcement learning (RL) has emerged as a key paradigm for improving LLM behavior in agentic settings. Existing agentic RL approaches largely rely on trajectory-level rewards, where a single reward is assigned at the end of the interaction trajectory. This feedback is typically either an outcome-based reward such as task success (Jin et al., 2025; Feng et al., 2025; Chen et al., 2026), or a composite reward that aggregates intermediate and outcome signals into a single scalar (Li et al., 2025; Wang et al., 2025a;b). However, such trajectory-level rewards provide only sparse supervision, making it difficult for LLM agents to effectively optimize their policies, and often leading to limited performance gains (Wei et al., 2025; Liu et al., 2026). To address this sparsity issue, recent work has introduced dense turn-level rewards to provide more fine-grained feedback at each turn of the interaction. Existing approaches obtain turn-level rewards in three main ways: handcrafted heuristics (Zhang et al., 2026), LLM-based judges (Wei et al., 2025), and learned reward models (Liu et al., 2026). Heuristic rewards rely on manually designed rules that encode desirable intermediate behaviors, while LLM-based judges use manually specified evaluation criteria or prompts to assess individual turns. Although both approaches provide intermediate supervision, their reward design is generally manual and task-specific: applying them to a new task may require additional domain knowledge, new rules, or redesigned evaluation prompts. Reward learning provides a more automated alternative. Given access to an outcome reward, a reward model learns how to assign intermediate feedback from collected trajectories, reducing the need to manually specify how individual turns should be evaluated.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Despite this advantage, we identify a fundamental limitation in existing reward learning approaches. Prior methods, such as PRIME (Cui et al., 2025) and iStar (Liu et al., 2026), typically generate trajectories using the current policy and optimize the reward model based on these trajectories using supervised objectives such as cross-entropy (CE) or direct preference optimization (DPO) losses. However, these approaches do not account for the dependency between the reward and the trajectory distribution induced by the policy, and thus implicitly assume that the trajectory distribution is independent of the reward during optimization. In reality, the trajectory distribution is implicitly determined by the reward, since the policy that generates trajectories is optimized with respect to the reward. Existing approaches optimize reward functions under a trajectory distribution that does not capture this implicit dependency, leading to a gradient for reward updates that fails to account for how changes in the reward propagate through the policy and in turn alter the trajectory distribution.

In this paper, we propose "**B**ilevel **A**gentic **R**eward **S**haping" (BARS), a framework for learning turn-level rewards that explicitly accounts for this implicit dependency. We formulate reward learning as a bilevel optimization problem, where the upper level optimizes the shaping reward r_θ, and the lower level learns the policy π_{r_θ} induced by this reward. A key feature of BARS is that it explicitly captures the full dependency of the upper-level objective on the reward parameter θ through both direct and indirect paths. Specifically, the shaping reward r_θ influences the policy π_{r_θ}, which in turn determines the generated trajectories y, and ultimately affects the upper-level loss $L(\theta, \pi_{r_\theta})$. By accounting for this dependency path $r_\theta \to \pi_{r_\theta} \to y \to L(\theta, \pi_{r_\theta})$, BARS computes the full hypergradient (Figure 1), where the indirect term captures how changes in the reward influence the learned policy and, consequently, the data distribution used to evaluate the objective. Prior methods such as PRIME and iStar only consider the direct term. The upper-level objective is compatible with standard supervised losses such as CE and DPO, while the lower-level optimization can be instantiated with standard RL algorithms such as PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), and RLOO (Ahmadian et al., 2024). Our key contributions are as follows:

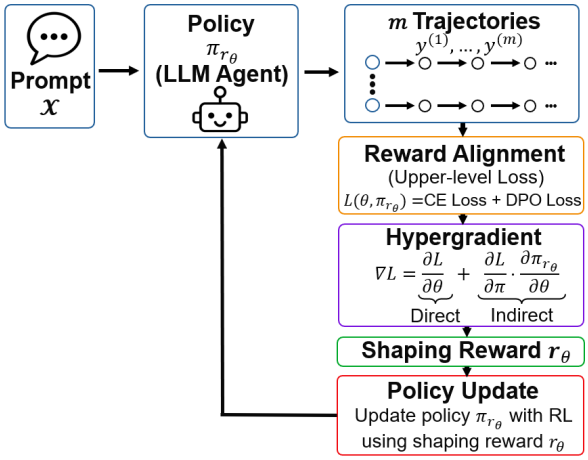

Figure 1: Overview of BARS. The shaping reward r_θ is learned in the upper level, while the policy π_{r_θ} is optimized in the lower level. BARS computes the full hypergradient, which captures both the direct dependency on r_θ and the indirect dependency through the path $r_\theta \to \pi_{r_\theta} \to y \to L(\theta, \pi_{r_\theta})$.

1. Bilevel Reward Shaping Framework. We formulate turn-level reward learning for agentic RL as a bilevel optimization problem that explicitly models the dependency between the learned reward and the policy induced by that reward. Unlike existing reward learning approaches that treat the trajectory distribution as independent of the learned reward, our formulation captures how the learned reward affects the policy and, consequently, the trajectories used for reward learning.

2. Full-Hypergradient Optimization. We develop an efficient algorithm that alternates between reward alignment and policy update within each iteration. Our method computes hypergradients that capture how reward updates propagate through the policy and influence the induced trajectory distribution, resulting in a more accurate and effective optimization objective.

3. Convergence Guarantee and Empirical Gains. We formally guarantee that BARS converges at a rate of $O(1/\sqrt{K})$ under an idealized lower-level optimization setting. Empirically, we instantiate the lower-level optimization of BARS with standard RL optimizers and demonstrate that BARS achieves higher accuracy and improved training stability compared to baseline methods.

2 Related Work

Agentic RL. Conventional agentic RL approaches (Jin et al., 2025; Feng et al., 2025; Chen et al., 2026) treat multi-turn interactive tasks as long-horizon single-turn problems and apply standard RL

algorithms using outcome rewards, such as task success. However, this trajectory-level formulation often proves inadequate for long-horizon multi-turn reasoning, as it overlooks the intermediate dynamics of agent-environment interactions, leading to sparse feedback and poor credit assignment. To address this limitation, subsequent works introduce process rewards (Li et al., 2025; Wang et al., 2025a;b; Qian et al., 2025; Singh et al., 2025; Zhang et al., 2025) that evaluate properties of intermediate turns, such as format quality and answer relevance. Nevertheless, these methods still suffer from sparsity because they aggregate intermediate signals together with the outcome reward into a single scalar reward delivered only at the end of the interaction trajectory. To provide denser supervision, recent approaches have shifted towards assigning turn-level rewards to each agent-environment interaction turn. Such turn-level rewards can be constructed using handcrafted heuristics (Zhang et al., 2026), LLM judges (Wei et al., 2025; Liu et al., 2025; Zha et al., 2025), or learned reward models (Liu et al., 2026). Heuristic and LLM-based rewards require manually designed rules, evaluation criteria, or prompts, which may require task-specific adaptation and domain knowledge when applied to new tasks. In contrast, reward learning derives turn-level supervision from outcome feedback, reducing the need to manually specify intermediate reward signals.

Reward Shaping for RL. Reward shaping improves RL by augmenting the ground-truth reward with additional shaping signals. Conventional methods rely on rule-based heuristics to design shaping rewards, including subgoal-based shaping (Singh et al., 2009; Konidaris & Barto, 2009), distance-based shaping (Mataric, 1994; Laud & DeJong, 2003), and potential-based shaping (Ng et al., 1999; Wiewiora, 2003). Recent learning-based approaches (Liu et al., 2026; Cui et al., 2025; Liu & Zhu, 2025) optimize shaping rewards directly from the ground-truth reward signal. Among them, (Liu et al., 2026; Cui et al., 2025) are most closely related to our work, as they learn dense shaping rewards from sparse outcome rewards for LLMs. However, these methods do not account for the implicit dependency between the shaping reward and the generated trajectory distribution.

3 PRELIMINARIES

Given a prompt x sampled from a dataset $\mathcal{D}$, an LLM agent interacts with the environment for T turns. The resulting interaction trajectory is denoted by $y = (y_1, \ldots, y_T)$, where each turn $y_t = (a_t, f_t)$ consists of an LLM response a_t and the corresponding environment feedback f_t. We model this interaction as a finite-horizon Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{D}, P, \gamma, T, r)$ that includes a state space $\mathcal{S}$, where each state $s_t = (x, y_{<t})$ includes a prompt x and all previous interactive turns $y_{<t}$. We define the initial state $s_1 = x$ as the prompt sampled from the dataset $\mathcal{D}$. The action space $\mathcal{A}$ consists of LLM responses, where on each turn the agent selects an action a_t based on the current state s_t. After the agent generates a response a_t, the environment produces feedback f_t, and the interaction proceeds to the next state $s_{t+1} = (x, y_{<t}, (a_t, f_t))$. The transition function $P(s_{t+1}|s_t, a_t)$ captures this interaction process. The discount factor is $\gamma \in (0, 1)$.

The reward function r assigns a scalar value to each state-action pair (s_t, a_t). In many agentic tasks, the objective is defined by an outcome reward that evaluates the overall success of the interaction. Such rewards are typically sparse and only provided at the end of the trajectory, i.e., $r(s_t, a_t) = 0$ for $t < T$. In this paper, we consider a binary success reward as the outcome reward. Specifically, the success reward r_{succ} assigns a terminal value $r_{\text{succ}}(s_T, a_T) \in \{0, 1\}$, indicating whether the trajectory successfully solves the task. We discuss how our framework can be adapted to non-binary outcome rewards in Appendix A and empirically evaluate on non-binary outcome reward in Appendix D.3.

The LLM agent is modeled as a stochastic policy $\pi(a_t|s_t)$, which generates an action a_t given the current state s_t. The objective is to maximize the expected success rate over prompts sampled from $\mathcal{D}$. Since the reward is only provided at the terminal turn, this objective reduces to maximizing the expected terminal success reward over trajectories induced by the policy:

$$\max_{\pi} \; E^{\pi}_{x\sim\mathcal{D}}\Big[r_{\text{succ}}(S_T, A_T)\Big|S_1 = x\Big], \tag{1}$$

where uppercase (S_t, A_t) denotes random variables and lowercase (s_t, a_t) denotes their realizations.

While this objective (1) precisely captures the task goal, the success reward is sparse, providing supervision only at the end of the trajectory, which makes policy optimization challenging for the LLM agent. To provide denser feedback, we consider a parameterized turn-level reward r_θ that assigns a scalar reward to each interaction turn. Policy optimization under the learned turn-level

reward is then formulated as

$$\max_{\pi} \; E^{\pi}_{x\sim\mathcal{D}}\Big[\sum_{t=1}^{T}\gamma^{t-1} r_\theta(S_t, A_t)\Big| S_1 = x\Big]. \tag{2}$$

The turn-level reward r_θ provides intermediate supervision for policy optimization, while its parameters θ must be learned from the available outcome feedback. This coupling between reward learning and policy optimization motivates the bilevel formulation introduced in the next section.

4 Problem formulation

We consider the problem of learning a dense turn-level reward r_θ from sparse outcome feedback. The learned reward provides intermediate supervision for policy optimization, while its parameters are learned according to the observed task outcomes. Importantly, reward learning and policy optimization are coupled: the learned reward determines how the policy is optimized, while the resulting policy determines the trajectory distribution used to further learn the reward. We therefore formulate turn-level reward learning as a bilevel optimization problem that explicitly captures this reward-policy-trajectory dependency. We first introduce the parameterization of the turn-level reward and then present the bilevel formulation.

4.1 Turn-level reward parameterization

Our goal is to construct a learnable turn-level reward that retains the original outcome feedback while providing dense intermediate supervision. A classical approach for introducing intermediate rewards is potential-based reward shaping (Ng et al., 1999), which augments the original reward according to

$$r_{\text{succ}}(s_t, a_t) \to r_{\text{succ}}(s_t, a_t) + \gamma\Phi(s_{t+1}) - \Phi(s_t),$$

where $\Phi(\cdot)$ is a state-dependent potential function. This construction motivates a useful design principle for our setting: rather than replacing the original outcome reward, the turn-level reward can retain the outcome signal and augment it with a structured dense signal at intermediate turns.

Motivated by this principle and recent reward learning approaches (Liu et al., 2026; Cui et al., 2025), we consider a state-dependent potential parameterized by a language model:

$$\Phi_\theta(s_1) = 0, \qquad \Phi_\theta(s_t) = \gamma^{1-t}\sum_{i=1}^{t-1}\log\frac{\pi_\theta(a_i|s_i)}{\pi_{\text{ref}}(a_i|s_i)} \quad t \geq 2,$$

where π_θ is a language model with parameter θ, and π_{ref} is a reference policy. Note that the potential function $\Phi_\theta(s_t)$ depends only on the state s_t since the trajectory history $\{(s_i, a_i)\}_{i=1}^{t-1}$ is fully contained in s_t. Using this parameterization, the resulting parameterized shaping reward is

$$r_\theta(s_t, a_t) = r_{\text{succ}}(s_t, a_t) + \gamma\Phi_\theta(s_{t+1}) - \Phi_\theta(s_t) = r_{\text{succ}}(s_t, a_t) + \gamma^{1-t}\log\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{ref}}(a_t|s_t)}. \tag{3}$$

Note that potential-based reward shaping serves only as motivation for the structure of (3). Our objective is not to preserve policy invariance under potential-based reward transformation. In particular, we deliberately do not impose the terminal boundary condition on the potential required by (Ng et al., 1999), allowing the cumulative shaping signal to remain nonzero and thus influence reward learning. Consequently, our parameterization should be viewed as a learnable dense reward inspired by potential-based shaping rather than a strict potential-based reward transformation.

4.2 Bilevel Optimization

We formulate a bilevel optimization problem where the upper level optimizes shaping reward to align with the success reward and the lower level computes the policy induced by the shaping reward:

$$\min_{\theta} L(\theta, \pi_{r_\theta}) = L_{\text{ce}}(\theta, \pi_{r_\theta}) + \lambda L_{\text{dpo}}(\theta, \pi_{r_\theta}) \tag{4}$$

$$\text{s.t. } \pi_{r_\theta} = \arg\max_{\pi} \; E^{\pi}_{x\sim\mathcal{D}}\Big[\sum_{t=1}^{T}\gamma^{t-1}\Big(r_\theta(S_t, A_t) - \beta\log\frac{\pi(A_t|S_t)}{\pi_{\text{ref}}(A_t|S_t)}\Big)\Big| S_1 = x\Big]. \tag{5}$$

The lower-level problem (5). Given a shaping reward r_θ, the lower level solves a regularized RL problem to obtain the corresponding policy π_{r_θ}, where β is the KL regularization coefficient. Note that π_{r_θ} is distinct from the policy π_θ used to parameterize the shaping reward. In particular, π_{r_θ} has its own model parameters. We use the notation π_{r_θ} to highlight the implicit dependency of the lower-level induced policy on the shaping reward r_θ.

The upper-level problem (4). To align the shaping reward with the binary success reward, we optimize a combination of a cross-entropy (CE) loss and a DPO-style contrastive loss: $L(\theta, \pi_{r_\theta}) = L_{\text{ce}}(\theta, \pi_{r_\theta}) + \lambda L_{\text{dpo}}(\theta, \pi_{r_\theta})$, where λ is a weighting coefficient. These two losses capture complementary aspects of reward learning: absolute outcome alignment and contrastive discrimination.

The CE loss encourages the shaping reward to reflect the success likelihood of individual trajectories. Given a prompt x, a trajectory y, and the success reward $r_{\text{succ}}(x, y) \in \{0, 1\}$, the CE loss is

$$L_{\text{ce}}(\theta, \pi_{r_\theta}) \triangleq -E_{x\sim\mathcal{D}, y\sim\pi_{r_\theta}(\cdot|x)}\Big[r_{\text{succ}}(x,y)\cdot\log\sigma(r_\theta(x,y))+(1-r_{\text{succ}}(x,y))\cdot\log[1-\sigma(r_\theta(x,y))]\Big],$$

where $\sigma(\cdot)$ is the sigmoid function, $\pi_{r_\theta}(y|x) \triangleq \prod_{t=1}^{T} \pi_{r_\theta}(y_t|x, y_{<t})$ is the probability of policy π_{r_θ} generating a trajectory y to the prompt x, $r_\theta(x, y) \triangleq \sum_{t=1}^{T} \gamma^{t-1} r_\theta(s_t, a_t)$, and $r_{\text{succ}}(x, y) \triangleq r_{\text{succ}}(s_T, a_T)$. While the CE loss aligns with success likelihood, it provides weak supervision for distinguishing between trajectories. To address this, the DPO loss accounts for relative preference, encouraging successful trajectories to receive higher rewards than unsuccessful ones:

$$L_{\text{dpo}}(\theta, \pi_{r_\theta}) \triangleq -E_{x\sim\mathcal{D}, (y^+, y^-)\sim\pi_{r_\theta}(\cdot|x)}\Big[\log\sigma\Big(r_\theta(x, y^+) - r_\theta(x, y^-)\Big)\Big],$$

where y^+ and y^- denote trajectories with success labels 1 and 0, respectively. When π_{r_θ} generates all positive or all negative trajectories, the DPO loss is 0 and the upper level relies on the CE loss.

Remark (distinctions from prior works). While PRIME (Cui et al., 2025) and iStar (Liu et al., 2026) also use CE or DPO losses for reward learning, our bilevel formulation explicitly accounts for the dependency between the shaping reward and generated trajectories through the hyperparameterized policy π_{r_θ}. This allows us to compute the full gradient for reward optimization, which captures not only the direct effect of r_θ but also its indirect effect through the induced policy (see Section 5 for details).

5 Algorithm and Convergence Guarantee

We develop Algorithm 1 to solve the bilevel optimization problem (4)–(5). At a high level, the algorithm alternates between (i) a policy update step that solves the lower-level problem (5), and (ii) a reward alignment step that solves the upper-level problem (4). A key component of our method is the full hypergradient (i.e., the full gradient of the upper-level objective with respect to θ), which captures both the direct effect of the shaping reward and its indirect effect through the induced policy.

Algorithm 1 Bilevel Agentic Reward Shaping (BARS)

Input: Dataset $\mathcal{D}$, success reward r_{succ}, initial shaping parameter θ_0, and reference policy π_{ref}.
Output: Learned shaping parameter θ_K and policy π_K.

1: Initialize policy $\pi_0 = \pi_{\text{ref}}$.
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: **Data sample**: Sample a batch of B prompts $\{x^{(n)}\}_{n=1}^{B}$ from $\mathcal{D}$.
4: **Policy rollout**: For each prompt $x^{(n)}$, generate m trajectories $\{y^{(n,j)}\}_{j=1}^{m}$ using policy π_k.
5: **Policy update**: Perform a one-step soft policy iteration under r_{θ_k} to obtain π_{k+1}.
6: **Reward alignment**: Compute the hypergradient in Lemmas 1–2 and update shaping parameter via $\theta_{k+1} = \theta_k - \alpha[g_{\text{ce}}(\theta_k) + \lambda g_{\text{dpo}}(\theta_k)]$.
7: **end for**

At each iteration k, BARS first samples B prompts from the dataset $\mathcal{D}$ and then uses the current policy π_k to generate m trajectories $\{y^{(n,j)}\}_{j=1}^{m}$ for each prompt $x^{(n)}$, where each trajectory corresponds to a sequence of state-action pairs $\{(s_t^{(n,j)}, a_t^{(n,j)})\}_{t=1}^{T}$. These m trajectories are reused for both policy update and reward alignment, enabling efficient training without additional rollouts.

Policy Update. The lower-level problem (5) can be completely solved by soft policy iteration (Haarnoja et al., 2017). However, it is computationally expensive to run multiple soft policy iterations to fully solve the lower-level problem at each iteration k. In contrast, we use only a one-step soft policy iteration under r_{θ_k} to partially solve the lower-level problem to obtain an updated policy π_{k+1}.

Reward Alignment. To solve the upper-level problem (4), we compute the full hypergradient of $L(\theta, \pi_{r_\theta})$ with respect to θ. Taking the CE loss L_{ce} as an example, we can rewrite it as $L_{\text{ce}}(\theta, \pi_{r_\theta}) = E_{x\sim\mathcal{D}}[\sum_y \pi_{r_\theta}(y|x) \cdot l_{\text{ce}}(\theta; (x, y))]$, where $l_{\text{ce}}(\theta; (x, y)) \triangleq -r_{\text{succ}}(x, y) \cdot \log \sigma(r_\theta(x, y)) - (1 - r_{\text{succ}}(x, y)) \cdot \log[1 - \sigma(r_\theta(x, y))]$ is the individual CE loss of (x, y). Since the trajectory distribution depends on θ through the induced policy π_{r_θ}, the hypergradient consists of two components:

$$\nabla L_{\text{ce}}(\theta, \pi_{r_\theta}) = \underbrace{E_{x\sim\mathcal{D}}\Big[\sum_y \pi_{r_\theta}(y|x) \cdot \nabla_\theta l_{\text{ce}}(\theta; (x, y))\Big]}_{\text{direct term}} + \underbrace{E_{x\sim\mathcal{D}}\Big[\sum_y \nabla_\theta \pi_{r_\theta}(y|x) \cdot l_{\text{ce}}(\theta; (x, y))\Big]}_{\text{indirect term}}.$$

The first term captures the direct dependency of the loss on θ, while the second term captures the indirect effect through the trajectory distribution induced by π_{r_θ}. PRIME (Cui et al., 2025) and iStar (Liu et al., 2026) only consider the first term, ignoring the implicit dependency of π_{r_θ} on θ. In contrast, we consider the full hypergradient and use the sampled trajectories in the "Policy rollout" step to estimate the hypergradient. Since the upper-level loss in (4) consists of a CE loss and a DPO loss, we provide the hypergradients of these two losses separately.

Lemma 1. *The hypergradient of the CE loss can be computed as*

$$g_{\text{ce}}(\theta_k) = \frac{1}{Bm} \sum_{n=1}^{B} \sum_{j=1}^{m} \Big[\nabla_\theta l_{\text{ce}}(\theta_k; (x^{(n)}, y^{(n,j)})) + l_{\text{ce}}(\theta_k; (x^{(n)}, y^{(n,j)})) \cdot \Delta_{\text{ce}}^{(n,j,k)}\Big],$$

where $l_{\text{ce}}(\theta; (x, y))$ is the CE loss of (x, y) whose expression and gradient are in Appendix B.1, and $\Delta_{\text{ce}}^{(n,j,k)}$ is a correction term whose full expression is in Appendix B.1.

For the DPO loss, we form preference pairs. Let $\bar{m}^{(n)}$ denote the number of successful trajectories among the m samples for the prompt $x^{(n)}$. Then, we construct $\bar{m}^{(n)}(m - \bar{m}^{(n)})$ preference pairs $\{(y^{(n,\bar{j},+)}, y^{(n,\bar{j},-)})\}_{\bar{j}=1}^{\bar{m}^{(n)}(m-\bar{m}^{(n)})}$, where each pair consists of a successful trajectory y^+ and an unsuccessful trajectory y^-. If the m trajectories are all positive or all negative, the DPO loss with respect to this prompt $x^{(n)}$ is 0.

Lemma 2. *The hypergradient of the DPO loss can be computed as*

$$g_{\text{dpo}}(\theta_k) = \frac{1}{B} \sum_{n=1}^{B} \frac{1}{\bar{m}^{(n)}(m - \bar{m}^{(n)})} \sum_{\bar{j}=1}^{\bar{m}^{(n)}(m-\bar{m}^{(n)})} \Big[\nabla_\theta l_{\text{dpo}}(\theta_k; (x^{(n)}, y^{(n,\bar{j},+)}, y^{(n,\bar{j},-)}))$$
$$+ l_{\text{dpo}}(\theta_k; (x^{(n)}, y^{(n,\bar{j},+)}, y^{(n,\bar{j},-)})) \cdot \Delta_{\text{dpo}}^{(n,\bar{j},k)}\Big],$$

where $l_{\text{dpo}}(\theta; (x, y^+, y^-))$ is the DPO loss of (x, y^+, y^-) whose expression and gradient are in Appendix B.2, and $\Delta_{\text{dpo}}^{(n,\bar{j},k)}$ is a correction term whose full expression is in Appendix B.2.

The correction terms in Lemma 1 and Lemma 2 account for how changes in the shaping reward affect the trajectory distribution. We then update the shaping parameter via gradient descent: $\theta_{k+1} = \theta_k - \alpha[g_{\text{ce}}(\theta_k) + \lambda g_{\text{dpo}}(\theta_k)]$, where α is a constant step size.

Assumption 1. *The policy satisfies $||\nabla_\theta \log \pi_\theta(a|s)|| \le C$ and $||\nabla^2_{\theta\theta} \log \pi_\theta(a|s)|| \le \bar{C}$, and the reward satisfies $|r_\theta(s, a)| \le \tilde{C}$ for all (s, a), where C, $\bar{C}$, and $\tilde{C}$ are positive constants.*

Assumption 1 is a standard assumption in RL (Zhang et al., 2020; Kumar et al., 2023).

Theorem 1. *Suppose Assumption 1 holds and the step size is $\alpha \propto 1/\sqrt{K}$. Then it holds that*

$$\frac{1}{K} \sum_{k=0}^{K-1} |\log \pi_{k+1}(a|s) - \log \pi_{r_{\theta_k}}(a|s)| \le O(\frac{1}{\sqrt{K}}), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}$$

$$\frac{1}{K} \sum_{k=0}^{K-1} E\Big[||\nabla L(\theta_k, \pi_k)||^2\Big] \le O(\frac{1}{\sqrt{K}}).$$

Theorem 1 shows that (i) the iteratively updated policy π_k tracks the optimal lower-level solution $\pi_{r_{\theta_k}}$, despite only performing one-step policy updates; and (ii) the shaping parameter θ_k converges to the set of stationary points of the upper-level objective. The proof is in Appendix B.

Practical Implementation. Algorithm 1 presents an idealized implementation of BARS that facilitates theoretical analysis. Our practical implementation introduces two modifications to make BARS compatible with practical training. First, instead of the one-step soft policy iteration used for the lower level in Algorithm 1, we optimize the policy using standard RL optimizers, such as one-step GRPO/PPO/RLOO. Second, we employ a computationally efficient approximation to compute the hypergradient. Consequently, the theoretical guarantees established for Algorithm 1 do not directly apply to these practical modifications. The complete practical algorithm is provided in Appendix C.

6 EXPERIMENTS

In this section, we demonstrate that BARS achieves higher accuracy and improved training stability compared to baseline methods. Training details and additional experiments are in Appendix D.

Benchmarks. We evaluate BARS on two search benchmarks and a long-horizon interactive e-commerce benchmark. Our main experiments consider two search benchmarks, NQ-Search and NQ-HotpotQA (Jin et al., 2025), and we also evaluate BARS on WebShop (Yao et al., 2022) in Appendix D.3. NQ-Search serves as the in-domain benchmark, where both the training and test sets are drawn from NQ (Karpukhin et al., 2020). NQ-HotpotQA evaluates generalization to out-of-domain, multi-step reasoning settings: its training set is constructed from NQ and HotpotQA (Yang et al., 2018), while its test set spans multiple sources, including general question answering (NQ, TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023)), and multi-hop question answering (HotpotQA and 2WikiMultiHopQA (Ho et al., 2020)). For both search benchmarks, the outcome reward is a binary "exact match" signal indicating whether the generated answer exactly matches the ground-truth answer. The number of turns is 2 for NQ-Search and 4 for NQ-HotpotQA, since HotpotQA and 2WikiMultiHopQA involve multi-hop questions.

Models. For the search experiments, we consider two base models of different scales: Qwen2.5-3B and Llama3.1-8B. For the WebShop experiment, we use the newer-generation Qwen3-4B model.

RL Optimizers. Our method is compatible with standard RL optimizers. In the main experiments, we use GRPO (Shao et al., 2024) as the default optimizer. To validate optimizer compatibility, we further evaluate our method with PPO (Schulman et al., 2017) and RLOO (Ahmadian et al., 2024).

Baselines. We compare against four representative baselines: (1) **RL+outcome reward (OR)** maximizes the binary "exact match" reward; (2) **RL+turn-level LLM judge (LLMJ)** (Wei et al., 2025) uses OpenAI's GPT-OSS-120B (OpenAI, 2025) as a judge to provide turn-level rewards based on format, information quality, and search efficiency (the judge prompt is provided in Appendix D); (3) **PRIME** (Cui et al., 2025) learns a shaping reward using the CE loss (although it is originally designed for RLOO, we adapt it to support GRPO and PPO); and (4) **iStar** (Liu et al., 2026) learns a shaping reward using the DPO loss.

Evaluation. We report success rate, i.e., the proportion of prompts for which the model's predicted answer exactly matches the ground truth. All results are computed from three random seeds.

6.1 MAIN RESULTS

The results in Table 1 demonstrate that BARS achieves the best performance across both models and benchmarks. Taking the NQ-Search benchmark as an example, on Qwen2.5-3B, BARS improves over OR by 6.9% and surpasses the strongest baseline PRIME by 2.5%. On Llama3.1-8B, BARS further reaches 48.7%, achieving the best overall success rate. In contrast, OR performs poorly, which highlights the difficulty of learning from sparse binary supervision without intermediate credit assignment. While LLM-based turn-level rewards (LLMJ) provide denser feedback, they yield only marginal improvement because they suffer from training instability due to the internal bias and inconsistency of the LLM judge. Methods that learn reward models (PRIME, iStar, and BARS) generally perform better, underscoring the importance of learned shaping rewards. Among these reward learning methods, BARS consistently achieves the strongest performance as its bilevel formulation captures the complete dependency between reward learning and policy optimization.

Table 1: Performance comparison (success rate) on NQ-Search and NQ-HotpotQA benchmarks. † denotes the best result before training collapse. All scores are reported as percentages.

Model	Method	NQ-Search	NQ-HotpotQA				
			NQ	HotpotQA	TriviaQA	PopQA	2WikiMultiHopQA
Qwen2.5-3B	OR	40.9 ± 0.3	43.0 ± 0.2†	33.0 ± 0.4†	59.3 ± 0.3†	40.9 ± 0.5†	27.5 ± 0.2†
	LLMJ	41.9 ± 0.5†	36.2 ± 0.4†	30.7 ± 0.4†	54.5 ± 0.5†	38.5 ± 0.6†	29.9 ± 0.4†
	PRIME	45.3 ± 0.3†	46.6 ± 0.2†	34.3 ± 0.2†	59.2 ± 0.3†	44.7 ± 0.2†	29.2 ± 0.3†
	iStar	44.0 ± 0.2	43.3 ± 0.2	32.1 ± 0.3	58.1 ± 0.2	42.7 ± 0.3	26.1 ± 0.4
	BARS	$\mathbf{47.8 \pm 0.3}$	$\mathbf{46.9 \pm 0.3}$	$\mathbf{42.1 \pm 0.2}$	$\mathbf{62.2 \pm 0.2}$	$\mathbf{46.8 \pm 0.3}$	$\mathbf{38.6 \pm 0.2}$
Llama3.1-8B	OR	40.7 ± 0.3†	42.4 ± 0.2†	30.1 ± 0.2†	58.6 ± 0.3†	40.8 ± 0.3†	27.1 ± 0.2†
	LLMJ	38.7 ± 0.5†	32.3 ± 0.3†	27.8 ± 0.5†	59.0 ± 0.4†	27.1 ± 0.3†	27.9 ± 0.4†
	PRIME	44.6 ± 0.2	44.5 ± 0.6	31.7 ± 0.3	60.6 ± 0.4	43.5 ± 0.3	28.8 ± 0.3
	iStar	45.9 ± 0.3†	42.2 ± 0.5†	32.4 ± 0.4†	59.3 ± 0.2†	43.9 ± 0.4†	27.6 ± 0.3†
	BARS	$\mathbf{48.7 \pm 0.2}$	$\mathbf{46.8 \pm 0.2}$	$\mathbf{36.6 \pm 0.1}$	$\mathbf{62.8 \pm 0.2}$	$\mathbf{46.4 \pm 0.2}$	$\mathbf{30.8 \pm 0.2}$

6.2 Training Dynamics

Figure 2 further illustrates the training dynamics of BARS and the baselines. On both training and test curves, BARS consistently maintains stable performance throughout training, avoiding the large fluctuations observed in several baselines. In contrast, OR improves slowly due to sparse outcome-level supervision. LLMJ exhibits significant instability, with large fluctuations and even severe performance drops, suggesting that LLM-based turn-level rewards can be noisy and inconsistent. While PRIME and iStar both improve upon OR, their optimization dynamics differ: PRIME shows noticeable instability during training, whereas iStar exhibits relatively smoother early-stage behavior but plateaus at a lower performance than BARS after around 100 training steps. We include the training and test curves on the NQ-HotpotQA benchmark in Appendix D.1. Across benchmarks, BARS consistently demonstrates more stable training dynamics compared to the baselines. These results indicate that BARS provides a more reliable optimization signal, leading to stable and consistent policy improvement.

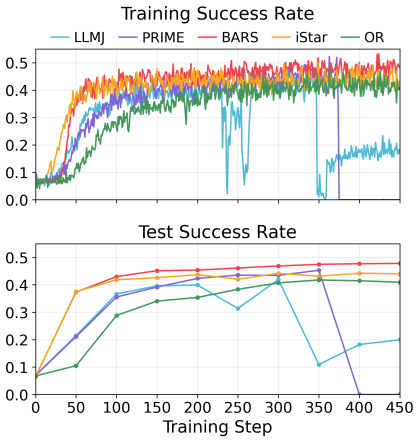

Figure 2: Training and test curves on Qwen2.5-3B and NQ-Search.

6.3 Compatibility with RL Optimizers

Table 2: Performance comparison on Qwen2.5-3B under different RL optimizers.

RL optimizer	Method	NQ-Search	NQ-HotpotQA				
			NQ	HotpotQA	TriviaQA	PopQA	2WikiMultiHopQA
PPO	OR	42.4 ± 0.2	40.1 ± 0.3	31.5 ± 0.2	59.0 ± 0.3	39.7 ± 0.2	27.2 ± 0.4
	LLMJ	43.0 ± 0.3	40.8 ± 0.3	32.2 ± 0.2	59.6 ± 0.2	40.9 ± 0.2	28.3 ± 0.3
	PRIME	45.4 ± 0.3	41.6 ± 0.2	29.2 ± 0.4	59.9 ± 0.2	39.8 ± 0.3	29.5 ± 0.2
	iStar	45.7 ± 0.5	41.4 ± 0.3	32.6 ± 0.2	59.8 ± 0.3	40.7 ± 0.4	31.1 ± 0.3
	BARS	$\mathbf{48.1 \pm 0.2}$	$\mathbf{43.0 \pm 0.2}$	$\mathbf{35.6 \pm 0.3}$	$\mathbf{61.1 \pm 0.2}$	$\mathbf{43.9 \pm 0.3}$	$\mathbf{33.1 \pm 0.3}$
RLOO	OR	38.2 ± 0.3†	40.4 ± 0.2†	32.2 ± 0.3†	57.3 ± 0.2†	40.1 ± 0.3†	26.9 ± 0.3†
	LLMJ	39.6 ± 0.6†	39.2 ± 0.2†	33.7 ± 0.3†	55.3 ± 0.2†	31.9 ± 0.2†	24.7 ± 0.2†
	PRIME	44.9 ± 0.2†	44.2 ± 0.3†	33.3 ± 0.2†	60.1 ± 0.3†	42.6 ± 0.4†	26.7 ± 0.2†
	iStar	44.7 ± 0.1	43.8 ± 0.2	33.5 ± 0.3	57.6 ± 0.3	41.7 ± 0.1	27.2 ± 0.4
	BARS	$\mathbf{47.5 \pm 0.3}$	$\mathbf{46.0 \pm 0.3}$	$\mathbf{35.7 \pm 0.1}$	$\mathbf{62.7 \pm 0.3}$	$\mathbf{46.1 \pm 0.4}$	$\mathbf{29.2 \pm 0.2}$

While Table 1 reports results using GRPO, BARS is agnostic to the choice of RL optimizers and remains effective across different optimizers. Table 2 shows that BARS achieves the best performance under both PPO and RLOO. For example, under RLOO, BARS improves over the strongest baseline PRIME by 2.6% on NQ-Search and yields consistent gains across NQ-HotpotQA subsets. Similar trends hold under PPO, where BARS outperforms the baselines.

6.4 Ablation Studies

We conduct ablation studies to understand the contribution of two components in BARS: (i) the full hypergradient, and (ii) our turn-level reward parameterization. To evaluate the role of the hypergradient, we compare BARS with a variant that removes the indirect term and optimizes the reward using only the direct gradient. To study the impact of reward parameterization, we replace our shaping parameterization with a standard reward model implemented as a linear head on top of the final transformer layer of π_{ref} (Ouyang et al., 2022). The results in Table 3 show that removing either component consistently degrades performance across models and benchmarks. In particular, using only the direct gradient leads to noticeable drops, confirming the importance of accounting for the dependency between the reward and the induced policy. Replacing our reward parameterization with a linear reward model also results in large performance degradation.

Table 3: Ablation studies (with GRPO as the RL optimizer).

Model	Ablation	NQ-Search	NQ-HotpotQA				
			NQ	HotpotQA	TriviaQA	PopQA	2WikiMultiHopQA
Qwen2.5-3B	direct gradient	46.1(↓ 1.7)	46.2(↓ 0.7)	37.2(↓ 4.9)	59.8(↓ 2.4)	43.8(↓ 3.0)	30.4(↓ 8.2)
	linear parameterization	45.8(↓ 2.0)	45.8(↓ 1.1)	36.9(↓ 5.2)	60.1(↓ 2.1)	40.7(↓ 6.1)	27.9(↓ 10.7)
Llama3.1-8B	direct gradient	46.2(↓ 2.5)	45.0(↓ 1.8)	34.4(↓ 2.2)	61.3(↓ 1.5)	44.7(↓ 1.7)	29.4(↓ 1.4)
	linear parameterization	45.6(↓ 3.1)	44.2(↓ 2.6)	32.7(↓ 3.9)	58.7(↓ 4.1)	43.0(↓ 3.4)	27.7(↓ 3.1)

6.5 Sensitivity Analyses

We conduct sensitivity analyses to evaluate BARS with respect to key hyperparameters, including (i) the number of rollouts m, (ii) the KL regularization coefficient β, and (iii) the weighting coefficient λ in the upper-level objective. We focus on the Qwen2.5-3B model on the NQ-Search benchmark. The results in Table 4 suggest the following. (i) Increasing the number of rollouts improves the performance as it provides more diverse trajectories for reward learning and policy optimization. However, the gains gradually plateau after $m = 8$. (ii) Performance improves as β increases from 0.001 to 0.005, indicating that moderate regularization helps training. However, overly large β values (e.g., 0.015 and 0.020) degrade performance, as strong regularization restricts policy updates. (iii) A moderate weighting coefficient (e.g., $\lambda = 0.05$ or 0.10) achieves the best performance, while only considering CE loss ($\lambda = 0$) or DPO loss leads to large degradation. This indicates that combining the complementary supervision (absolute alignment and contrastive discrimination) is important for reward learning.

Table 4: Sensitivity analysis using Qwen2.5-3B with GRPO as the RL optimizer.

Rollout number m	4	6	8	10	12
Success rate	44.9 ± 0.3	46.6 ± 0.2	47.8 ± 0.3	48.2 ± 0.2	48.4 ± 0.3
KL coefficient β	0.001	0.005	0.010	0.015	0.020
Success rate	45.8 ± 0.4	47.8 ± 0.3	47.6 ± 0.2	46.2 ± 0.3	44.5 ± 0.2
Weighting coefficient λ	0	0.05	0.10	0.15	DPO only
Success rate	46.8 ± 0.2	47.5 ± 0.2	47.8 ± 0.3	46.5 ± 0.5	45.8 ± 0.6

7 Conclusion

This paper proposes BARS, a bilevel framework for learning dense turn-level rewards from sparse outcome feedback in agentic RL. BARS explicitly captures the dependency between rewards, policies, and the induced trajectory distribution through a full hypergradient. We develop an idealized BARS algorithm based on soft policy iteration and establish a convergence rate of $O(1/\sqrt{K})$ under standard assumptions. For practical training, BARS can be instantiated with standard RL optimizers and an efficient approximation of the hypergradient. Empirical results demonstrate that BARS achieves improved training stability and performance across multiple models and benchmarks.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Annual Meeting of the Association for Computational Linguistics*, pp. 12248–12267, 2024.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, et al. Learning to reason with search for LLMs via reinforcement learning. In *Advances in Neural Information Processing Systems*, 2026.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in LLMs. *arXiv preprint arXiv:2504.11536*, 2025.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models. In *International Conference on Learning Representations*, 2024.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361, 2017.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training llms to reason and leverage search engines with reinforcement learning. In *Conference on Language Modeling*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020.

George Konidaris and Andrew Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. *Advances in Neural Information Processing Systems*, 22, 2009.

Harshat Kumar, Alec Koppel, and Alejandro Ribeiro. On the sample complexity of actor-critic method for reinforcement learning with function approximation. *Machine Learning*, 112(7): 2433–2467, 2023.

Adam Laud and Gerald DeJong. The influence of reward on the speed of reinforcement learning: An analysis of shaping. In *International Conference on Machine Learning*, pp. 440–447, 2003.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. ToRL: Scaling tool-integrated RL. *arXiv preprint arXiv:2503.23383*, 2025.

Shicheng Liu and Minghui Zhu. UTILITY: Utilizing explainable reinforcement learning to improve reinforcement learning. In *International Conference on Learning Representations*, 2025.

Xiaoqian Liu, Ke Wang, Yongbin Li, Yuchuan Wu, Wentao Ma, Aobo Kong, Fei Huang, Jianbin Jiao, and Junge Zhang. Epo: Explicit policy optimization for strategic reasoning in llms via reinforcement learning. In *Annual Meeting of the Association for Computational Linguistics*, pp. 15371–15396, 2025.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Junge Zhang, and Jianbin Jiao. Agentic reinforcement learning with implicit step rewards. In *International Conference on Learning Representations*, 2026.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Annual Meeting of the Association for Computational Linguistics*, pp. 9802–9822, 2023.

Maja J Mataric. Reward functions for accelerated learning. In *International Conference on Machine Learning*, pp. 181–189. 1994.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, volume 99, pp. 278–287, 1999.

OpenAI. gpt-oss-120b. `https://huggingface.co/openai/gpt-oss-120b`, 2025.

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

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. *Advances in Neural Information Processing Systems*, 2025.

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

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

Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. Agentic reasoning and tool integration for LLMs via reinforcement learning. *arXiv preprint arXiv:2505.01441*, 2025.

Satinder Singh, Richard L Lewis, and Andrew G Barto. Where do rewards come from. In *Annual Conference of the Cognitive Science Society*, pp. 2601–2606, 2009.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. OTC: Optimal tool calls via reinforcement learning. *arXiv e-prints*, pp. arXiv–2504, 2025a.

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

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, et al. Reinforcing multi-turn reasoning in LLM agents via turn-level reward design. *arXiv preprint arXiv:2505.11821*, 2025.

Eric Wiewiora. Potential-based shaping and q-value initialization are equivalent. *Journal of Artificial Intelligence Research*, 19:205–208, 2003.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

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

Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. Maximum-likelihood inverse reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 35:10122–10135, 2022.

Kaiwen Zha, Zhengqi Gao, Maohao Shen, Zhang-Wei Hong, Duane S Boning, and Dina Katabi. RL tango: Reinforcing generator and verifier together for language reasoning. *Advances in Neural Information Processing Systems*, 2025.

Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Basar. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6): 3586–3612, 2020.

Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. In *Annual Meeting of the Association for Computational Linguistics*, pp. 13643–13658, 2024.

Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. Nemotron-research-tool-n1: Tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*, 2(10), 2025.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. RLVMR: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. In *International Conference on Learning Representations*, 2026.

A Adaptation to Non-binary Outcome Rewards

While this paper considers the case where the outcome reward is a binary success reward, the proposed framework can be directly applied to cases where the outcome reward is non-binary. In our framework, we need the outcome reward in the upper level (4) to learn a shaping reward. The upper-level loss consists of a cross-entropy (CE) loss and a DPO loss.

The CE loss also supports non-binary outcome rewards as long as the outcome reward $r_{\text{out}}(x, y) \in [0, 1]$. In this case, the CE loss becomes:

$$L_{\text{ce}}(\theta, \pi_{r_\theta}) \triangleq -E_{x\sim\mathcal{D}, y\sim\pi_{r_\theta}(\cdot|x)}\Big[r_{\text{out}}(x, y)\cdot\log\sigma(r_\theta(x, y))+(1-r_{\text{out}}(x, y))\cdot\log[1-\sigma(r_\theta(x, y))]\Big].$$

In the case where the original outcome reward is beyond $[0, 1]$, e.g., $r_{\text{out}}(x, y) \in [C_l, C_r]$ where C_l and C_r are some bounded constants, we can normalize the outcome reward to $[0, 1]$ and get the normalized outcome reward $\bar{r}_{\text{out}} = \frac{r_{\text{out}}-C_l}{C_r-C_l}$. We can use the normalized outcome reward in the CE loss. Note that $\bar{r}_{\text{out}}$ and r_{out} have the same optimal policy because any linear transformation of the reward function does not change the corresponding optimal policy (Ng et al., 1999).

The DPO loss also works for the non-binary outcome reward because we can still formulate preference pairs where a trajectory with a higher outcome reward is preferred to a trajectory with a lower outcome reward.

B Proofs

In this section, we provide the theoretical foundations and proof for the theoretical statements in the paper.

In standard RL (Sutton & Barto, 1998), the policy is stationary for infinite-horizon MDPs, while the policy should depend on t for finite-horizon MDPs. This leads to a mismatch for agentic RL and RL for LLMs. Specifically, the generation of LLMs is always finite-horizon (finite number of tokens or finite number of turns), which fits finite-horizon MDPs and thus the policy model should be time-dependent. However, in practice, only a stationary LLM policy is learned. To address this issue, in the proof, we consider a unified infinite-horizon MDP $(\mathcal{S}, \mathcal{A}, \mathcal{D}, P, \gamma, r)$ that does not include the time horizon T, and its time horizon is infinite. However, we treat s_{T+1} as an absorbing state, i.e., it can only transit to itself regardless of the action. Moreover, the reward $r(s_t, a_t) \equiv 0$ for $t > T$. By this way, we use an infinite-horizon MDP to model the finite-horizon MDP in the paper. The discount factor is $\gamma \in (0, 1)$ for infinite-horizon MDPs. Moreover, the next state is $s_{t+1} = (s_t, a_t, f_t)$, where f_t is the environment feedback that depends on the action a_t. In the proof, for simplicity, we omit f_t and write $s_{t+1} = (s_t, a_t)$.

Lemma 3 ((Haarnoja et al., 2017; 2018)). *The optimal policy π_{r_θ} of the lower-level optimization problem (5) is the soft policy:*

$$\begin{aligned}
\pi_{r_\theta}(a_t|s_t) &= \frac{\pi_{\text{ref}}(a_t|s_t)\exp(\frac{1}{\beta}Q^*_{r_\theta}(s_t, a_t))}{\sum_{a'\in\mathcal{A}}\pi_{\text{ref}}(a'|s_t)\exp(\frac{1}{\beta}Q^*_{r_\theta}(s_t, a'))}\\
Q^*_{r_\theta}(s_t, a_t) &= r_\theta(s_t, a_t) + \gamma V^*_{r_\theta}(s_{t+1})\\
V^*_{r_\theta}(s_t) &= \beta\log\Big[\sum_{a'\in\mathcal{A}}\pi_{\text{ref}}(a'|s_t)\exp(\frac{1}{\beta}Q^*_{r_\theta}(s_t, a'))\Big],
\end{aligned}$$

where $Q^*_{r_\theta}$ and $V^*_{r_\theta}$ are respectively the optimal action-value function and state-value function under the reward r_θ.

B.1 Proof of Lemma 1

The full hypergradient needs to consider the implicit dependency of the policy π_{r_θ} on θ. Therefore, the following claim computes the gradient $\nabla_\theta \log \pi_{r_\theta}$.

Claim 1. *We have the following gradient:*

$\nabla_\theta \log \pi_{r_\theta}(a_t|s_t)$

$$= \frac{1}{\beta} E^{\pi_{r_\theta}}\Big[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_t = s_t, A_t = a_t\Big] - \frac{1}{\beta} E^{\pi_{r_\theta}}\Big[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_t = s_t\Big].$$

Proof. Define $Z_{r_\theta}(s_t, a_t) \triangleq \pi_{\text{ref}}(a_t|s_t) \exp(\frac{1}{\beta} Q^*_{r_\theta}(s_t, a_t))$ and $Z_{r_\theta}(s_t) \triangleq \exp(\frac{1}{\beta} V^*_{r_\theta}(s_t))$.

$$\begin{aligned}
\nabla_\theta \log Z_{r_\theta}(s_t) &= \frac{\sum_{a_t \in \mathcal{A}} \nabla_\theta Z_{r_\theta}(s_t, a_t)}{Z_{r_\theta,}(s_t)}, \\
&= \sum_{a_t \in \mathcal{A}} \frac{Z_{r_\theta}(s_t, a_t)}{Z_{r_\theta}(s_t)} \nabla_\theta \log Z_{r_\theta}(s_t, a_t), \\
&= \sum_{a_t \in \mathcal{A}} \pi_{r_\theta}(a_t|s_t) \Big[\frac{\nabla_\theta r_\theta(s_t, a_t)}{\beta} + \gamma \nabla_\theta \log Z_{r_\theta}(s_{t+1})\Big], \\
&= \sum_{a_t \in \mathcal{A}} \pi_{r_\theta}(a_t|s_t) \Big[\frac{\nabla_\theta r_\theta(s_t, a_t)}{\beta} \\
&+ \gamma \sum_{a_{t+1} \in \mathcal{A}} \pi_{r_\theta}(a_{t+1}|s_{t+1}) \Big(\frac{\nabla_\theta r_\theta(s_{t+1}, a_{t+1})}{\beta} + \gamma \nabla_\theta \log Z_{r_\theta}(s_{t+2})\Big)\Big].
\end{aligned}$$

By continuing to expand $\nabla_\theta \log Z_{r_\theta}(s_{t+2})$, we can get

$$\nabla_\theta \log Z_{r_\theta}(s_t) = \frac{1}{\beta} E^{\pi_{r_\theta}}[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_t = s_t],$$

and similarly we can get

$$\nabla_\theta \log Z_{r_\theta}(s_t, a_t) = \frac{1}{\beta} E^{\pi_{r_\theta}}[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_t = s_t, A_t = a_t].$$

Thus, we have the gradient $\nabla_\theta \log \pi_{r_\theta}(a_t|s_t) = \nabla_\theta \log Z_{r_\theta}(s_t, a_t) - \nabla_\theta \log Z_{r_\theta}(s_t)$. □

Now, we can proceed to obtain the hypergradient. Define the pointwise cross-entropy loss

$$l_{\text{ce}}(\theta; (x, y)) \triangleq -\Big[r_{\text{succ}}(x, y) \log \sigma(r_\theta(x, y)) + \big(1 - r_{\text{succ}}(x, y)\big) \log\big(1 - \sigma(r_\theta(x, y))\big)\Big].$$

Then, the overall loss is

$$L_{\text{ce}}(\theta, \pi_{r_\theta}) \triangleq E_{x \sim \mathcal{D},\, y \sim \pi_{r_\theta}(\cdot|x)}\big[l_{\text{ce}}(\theta; (x, y))\big].$$

To compute the full gradient with respect to θ, we differentiate both the explicit dependence of l_{ce} on θ and the implicit dependence through $y \sim \pi_{r_\theta}(\cdot|x)$:

$$\begin{aligned}
\nabla_\theta L_{\text{ce}}(\theta, \pi_{r_\theta}) &= \nabla_\theta E_{x \sim \mathcal{D},\, y \sim \pi_{r_\theta}(\cdot|x)}\big[l_{\text{ce}}(\theta; (x, y))\big] \\
&= E_{x \sim \mathcal{D},\, y \sim \pi_{r_\theta}(\cdot|x)}\Big[l_{\text{ce}}(\theta; (x, y))\, \nabla_\theta \log \pi_{r_\theta}(y|x) + \nabla_\theta l_{\text{ce}}(\theta; (x, y))\Big]. \qquad (6)
\end{aligned}$$

Recall from Claim 1 that

$$\begin{aligned}
\nabla_\theta \log \pi_{r_\theta}(a_t|s_t) &= \nabla_\theta \log Z_{r_\theta}(s_t, a_t) - \nabla_\theta \log Z_{r_\theta}(s_t) \\
\nabla_\theta \log Z_{r_\theta}(s_t, a_t) &= \frac{1}{\beta} E^{\pi_{r_\theta}}[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_t = s_t, A_t = a_t] \\
\nabla_\theta \log Z_{r_\theta}(s_t) &= \frac{1}{\beta} E^{\pi_{r_\theta}}[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_t = s_t].
\end{aligned}$$

Therefore, the term $\nabla_\theta \log \pi_{r_\theta}(y|x)$ in the CE hypergradient (6) is

$\nabla_\theta \log \pi_{r_\theta}(y|x)$

$$
\begin{aligned}
&= \sum_{t=1}^{\infty} \nabla_\theta \log \pi_{r_\theta}(a_t|s_t) \\
&= \sum_{t=1}^{\infty} \nabla_\theta \log Z_{r_\theta}(s_t, a_t) - \nabla_\theta \log Z_{r_\theta}(s_t) \\
&= \frac{1}{\beta} \sum_{t=1}^{\infty} \Big[E^{\pi_{r_\theta}}[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i)|S_t = s_t, A_t = a_t] - E^{\pi_{r_\theta}}[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i)|S_t = s_t] \Big] \\
&\overset{(a)}{=} \frac{1}{\beta} \sum_{t=1}^{T} \Big[E^{\pi_{r_\theta}}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i)|S_t = s_t, A_t = a_t] - E^{\pi_{r_\theta}}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i)|S_t = s_t] \Big],
\end{aligned} \tag{7}
$$

where (a) follows from the fact that $r_\theta(s_t, a_t) \equiv 0$ for $t > T$ in our infinite-time MDP formulation. Therefore, the CE hypergradient (6) becomes

$$
\begin{aligned}
&\nabla_\theta L_{\text{ce}}(\theta, \pi_{r_\theta}) = E_{x\sim\mathcal{D},\, y\sim\pi_{r_\theta}(\cdot|x)} \Big[\nabla_\theta l_{\text{ce}}(\theta; (x, y)) + l_{\text{ce}}(\theta; (x, y)) \cdot \\
&\frac{1}{\beta} \sum_{t=1}^{T} \Big(E^{\pi_{r_\theta}}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i)|S_t = s_t, A_t = a_t] - E^{\pi_{r_\theta}}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i)|S_t = s_t] \Big) \Big].
\end{aligned} \tag{8}
$$

In practice, we sample a finite batch of B prompts and use the policy π_k (rather than $\pi_{r_{\theta_k}}$) to generate m trajectories for each prompt at each iteration. Therefore, we approximate the CE hypergradient $\nabla L_{\text{ce}}(\theta_k, \pi_{r_{\theta_k}})$ via $g_{\text{ce}}(\theta_k)$:

$$
\begin{aligned}
g_{\text{ce}}(\theta_k) = \frac{1}{Bm} \sum_{n=1}^{B} \sum_{j=1}^{m} \Big[&\nabla_\theta l_{\text{ce}}(\theta_k; (x^{(n)}, y^{(n,j)})) + l_{\text{ce}}(\theta_k; (x^{(n)}, y^{(n,j)})) \cdot \\
&\frac{1}{\beta} \sum_{t=1}^{T} \Big(E^{\pi_k}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_{\theta_k}(S_i, A_i)|S_t = s_t^{(n,j)}, A_t = a_t^{(n,j)}] \\
&- E^{\pi_k}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_{\theta_k}(S_i, A_i)|S_t = s_t^{(n,j)}] \Big) \Big].
\end{aligned} \tag{9}
$$

For simplicity of notation, we denote

$$
\begin{aligned}
\Delta_{\text{ce}}^{(n,j,k)} = \frac{1}{\beta} \sum_{t=1}^{T} \Big(&E^{\pi_k}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_{\theta_k}(S_i, A_i)|S_t = s_t^{(n,j)}, A_t = a_t^{(n,j)}] \\
&- E^{\pi_k}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_{\theta_k}(S_i, A_i)|S_t = s_t^{(n,j)}] \Big).
\end{aligned}
$$

Therefore,

$$
g_{\text{ce}}(\theta_k) = \frac{1}{Bm} \sum_{n=1}^{B} \sum_{j=1}^{m} \Big[\nabla_\theta l_{\text{ce}}(\theta_k; (x^{(n)}, y^{(n,j)})) + l_{\text{ce}}(\theta_k; (x^{(n)}, y^{(n,j)})) \cdot \Delta_{\text{ce}}^{(n,j,k)} \Big].
$$

The first term in (9) is the standard gradient of a CE loss $\nabla_\theta l_{\text{ce}}(\theta; (x, y))$. For the sake of completeness, we also include its expression here. Let

$$
s \triangleq r_{\text{succ}}(x, y), \qquad z_\theta \triangleq r_\theta(x, y).
$$

Then

$$
l_{\text{ce}}(\theta; (x, y)) = -\Big[s \log \sigma(z_\theta) + (1 - s) \log(1 - \sigma(z_\theta)) \Big].
$$

Differentiating with respect to θ,

$$\nabla_\theta l_{\text{ce}}(\theta;(x,y)) = -\left[s\nabla_\theta \log\sigma(z_\theta) + (1-s)\nabla_\theta \log(1-\sigma(z_\theta))\right].$$

Using

$$\nabla_\theta \log\sigma(z_\theta) = \big(1-\sigma(z_\theta)\big)\nabla_\theta z_\theta, \qquad \nabla_\theta \log(1-\sigma(z_\theta)) = -\sigma(z_\theta)\nabla_\theta z_\theta,$$

we obtain

$$\nabla_\theta l_{\text{ce}}(\theta;(x,y)) = \big(\sigma(z_\theta)-s\big)\nabla_\theta z_\theta = \big(\sigma(r_\theta(x,y)) - r_{\text{succ}}(x,y)\big)\nabla_\theta r_\theta(x,y). \tag{10}$$

B.2 Proof of Lemma 2

Consider a prompt x together with a preferred response y^+ and a dispreferred response y^-. Define the reward-induced preference margin

$$\Delta_\theta(x,y^+,y^-) \triangleq r_\theta(x,y^+) - r_\theta(x,y^-),$$

and the pointwise DPO loss

$$l_{\text{dpo}}(\theta;(x,y^+,y^-)) \triangleq -\log\sigma\big(\Delta_\theta(x,y^+,y^-)\big).$$

The overall DPO loss is

$$L_{\text{dpo}}(\theta,\pi_{r_\theta}) \triangleq E_{x\sim\mathcal{D},\ (y^+,y^-)\sim\pi_{r_\theta}(\cdot,\cdot|x)}\big[l_{\text{dpo}}(\theta;(x,y^+,y^-))\big].$$

Again, the full gradient must account for both the explicit dependence of l_{dpo} on θ and the implicit dependence through sampling from $\pi_{r\theta}$:

$$\begin{aligned}&\nabla_\theta L_{\text{dpo}}(\theta,\pi_{r_\theta}) \\ &= E_{x\sim\mathcal{D},\ (y^+,y^-)\sim\pi_{r_\theta}(\cdot,\cdot|x)}\Big[l_{\text{dpo}}(\theta;(x,y^+,y^-))\,\nabla_\theta \log\pi_{r_\theta}(y^+,y^-|x) + \nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big].\end{aligned} \tag{11}$$

Since the pair is formed by independent samples from the same policy, then

$$\log\pi_{r_\theta}(y^+,y^-|x) = \log\pi_{r_\theta}(y^+|x) + \log\pi_{r_\theta}(y^-|x),$$

so

$$\nabla_\theta \log\pi_{r_\theta}(y^+,y^-|x) = \nabla_\theta \log\pi_{r_\theta}(y^+|x) + \nabla_\theta \log\pi_{r_\theta}(y^-|x). \tag{12}$$

By plugging (7) and (12) into (11) and considering that the trajectories are generated by π_k rather than $\pi_{r_{\theta_k}}$, we can approximate the hypergradient $\nabla_\theta L_{\text{dpo}}(\theta_k,\pi_{r_{\theta_k}})$ via $g_{\text{dpo}}(\theta_k)$:

$$\begin{aligned}g_{\text{dpo}}(\theta_k) = \frac{1}{B}\sum_{n=1}^{B}\frac{1}{\bar{m}^{(n)}(m-\bar{m}^{(n)})}\sum_{\bar{j}=1}^{\bar{m}^{(n)}(m-\bar{m}^{(n)})}&\Big[\nabla_\theta l_{\text{dpo}}(\theta_k;(x^{(n)},y^{(n,\bar{j},+)},y^{(n,\bar{j},-)}))\\
+\, l_{\text{dpo}}(\theta_k;(x^{(n)},y^{(n,\bar{j},+)},y^{(n,\bar{j},-)}))\cdot\frac{1}{\beta}\sum_{t=1}^{T}\Big(&E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t=s_t^{(n,\bar{j},+)},A_t=a_t^{(n,\bar{j},+)}]\\
-\,&E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t=s_t^{(n,\bar{j},+)}]\\
+\,&E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t=s_t^{(n,\bar{j},-)},A_t=a_t^{(n,\bar{j},-)}]\\
-\,&E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t=s_t^{(n,\bar{j},-)}]\Big)\Big].\end{aligned} \tag{13}$$

For simplicity of notation, we denote

$$\Delta_{\text{dpo}}^{(n,\bar{j},k)} = \frac{1}{\beta}\sum_{t=1}^{T}\Big(E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t=s_t^{(n,\bar{j},+)},A_t=a_t^{(n,\bar{j},+)}]$$

$$
\begin{aligned}
&- E^{\pi_k}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_{\theta_k}(S_i, A_i) | S_t = s_t^{(n,\bar{j},+)}] \\
&+ E^{\pi_k}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_{\theta_k}(S_i, A_i) | S_t = s_t^{(n,\bar{j},-)}, A_t = a_t^{(n,\bar{j},-)}] \\
&- E^{\pi_k}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_{\theta_k}(S_i, A_i) | S_t = s_t^{(n,\bar{j},-)}]\Big).
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
g_{\text{dpo}}(\theta_k) = \frac{1}{B}\sum_{n=1}^{B} \frac{1}{\bar{m}^{(n)}(m-\bar{m}^{(n)})} \sum_{\bar{j}=1}^{\bar{m}^{(n)}(m-\bar{m}^{(n)})} \Big[\nabla_\theta l_{\text{dpo}}(\theta_k; (x^{(n)}, y^{(n,\bar{j},+)}, y^{(n,\bar{j},-)})) \\
+ l_{\text{dpo}}(\theta_k; (x^{(n)}, y^{(n,\bar{j},+)}, y^{(n,\bar{j},-)})) \cdot \Delta_{\text{dpo}}^{(n,\bar{j},k)}.
\end{aligned}
$$

The first term in (13) is the standard gradient of a DPO loss $\nabla_\theta l_{\text{dpo}}(\theta; (x, y^+, y^-))$. For the sake of completeness, we compute the gradient of l_{dpo}. Since

$$
l_{\text{dpo}}(\theta; (x, y^+, y^-)) = -\log \sigma(\Delta_\theta),
$$

we have

$$
\nabla_\theta l_{\text{dpo}}(\theta; (x, y^+, y^-)) = -\big(1 - \sigma(\Delta_\theta)\big)\nabla_\theta \Delta_\theta.
$$

Using

$$
\Delta_\theta(x, y^+, y^-) = r_\theta(x, y^+) - r_\theta(x, y^-),
$$

we have that its gradient is

$$
\nabla_\theta \Delta_\theta(x, y^+, y^-) = \nabla_\theta r_\theta(x, y^+) - \nabla_\theta r_\theta(x, y^-).
$$

Therefore,

$$
\nabla_\theta l_{\text{dpo}}(\theta; (x, y^+, y^-)) = -\big(1 - \sigma(\Delta_\theta(x, y^+, y^-))\big)\Big(\nabla_\theta r_\theta(x, y^+) - \nabla_\theta r_\theta(x, y^-)\Big). \tag{14}
$$

B.3 Preliminary Theoretical Results for Theorem 1

Lemma 4. *Suppose Assumption 1 holds, then the shaping reward function r_θ satisfies $||\nabla_\theta r_\theta(s,a)|| \le C$ and $||\nabla^2_{\theta\theta} r_\theta(s,a)|| \le \bar{C}$ for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, where C and $\bar{C}$ are positive constants.*

Proof. This is a direct result based on Assumption 1. From the reward parameterization, we can see that $\nabla_\theta r_\theta(s_t, a_t) = \gamma^{1-t}\nabla_\theta \log \pi_\theta(a_t|s_t)$ and $\nabla^2_{\theta\theta} r_\theta(s_t, a_t) = 1 - \gamma^t \nabla^2_{\theta\theta} \log \pi_\theta(a_t|s_t)$. Therefore, $||\nabla_\theta r_\theta(s,a)|| \le \gamma^{1-T}||\nabla_\theta \log \pi_\theta(a|s)||$ and $||\nabla^2_{\theta\theta} r_\theta(s,a)|| \le \gamma^{1-T}||\nabla^2_{\theta\theta} \log \pi_\theta(a|s)||$. Since γ^{1-T} is a bounded constant, we can always find a value of C to upper bound both $||\nabla_\theta r_\theta(s,a)||$ and $||\nabla_\theta \log \pi_\theta(a_t|s_t)||$, and a value of $\bar{C}$ to upper bound both $||\nabla^2_{\theta\theta} r_\theta(s,a)||$ and $||\nabla^2_{\theta\theta} \log \pi_\theta(a|s)||$. For simplicity, we use C and $\bar{C}$ to bound both. □

Lemma 5. *Suppose Assumption 1 holds. Then, it holds that*

$$
\begin{aligned}
&0 \le l_{\text{ce}}(\theta; (x,y)) \le \log(1 + e^{T\tilde{C}}) \\
&\|\nabla_\theta l_{\text{ce}}(\theta; (x,y))\| \le TC \\
&\|\nabla^2_{\theta\theta} l_{\text{ce}}(\theta; (x,y))\| \le \frac{1}{4}T^2C^2 + T\bar{C}.
\end{aligned}
$$

Proof. Recall that

$$
l_{\text{ce}}(\theta; (x,y)) = -\Big[r_{\text{succ}}(x,y)\log\sigma(r_\theta(x,y)) + \big(1 - r_{\text{succ}}(x,y)\big)\log\big(1 - \sigma(r_\theta(x,y))\big)\Big].
$$

Let

$$s \triangleq r_{\mathrm{succ}}(x,y) \in \{0,1\}, \qquad z_\theta \triangleq r_\theta(x,y).$$

Then

$$l_{\mathrm{ce}}(\theta;(x,y)) = -\big[s\log\sigma(z_\theta) + (1-s)\log(1-\sigma(z_\theta))\big].$$

We first rewrite l_{ce} as a scalar function of z_θ. Define

$$\phi_s(z) \triangleq -\big[s\log\sigma(z) + (1-s)\log(1-\sigma(z))\big].$$

Then

$$l_{\mathrm{ce}}(\theta;(x,y)) = \phi_s(z_\theta).$$

Step 1: boundedness of the loss. Since $s \in \{0,1\}$, there are two cases.

If $s=1$, then

$$\phi_1(z) = -\log\sigma(z) = \log(1+e^{-z}).$$

Since $r_\theta(s,a) \le \tilde{C}$, then $|z_\theta| \le T\tilde{C}$, we have $z_\theta \in [-T\tilde{C}, T\tilde{C}]$, and thus

$$0 \le \phi_1(z_\theta) \le \phi_1(-T\tilde{C}) = \log(1+e^{T\tilde{C}}).$$

If $s=0$, then

$$\phi_0(z) = -\log(1-\sigma(z)) = \log(1+e^{z}),$$

so again using $z_\theta \in [-T\tilde{C}, T\tilde{C}]$,

$$0 \le \phi_0(z_\theta) \le \phi_0(T\tilde{C}) = \log(1+e^{T\tilde{C}}).$$

Therefore, in both cases,

$$0 \le l_{\mathrm{ce}}(\theta;(x,y)) \le \log(1+e^{T\tilde{C}}).$$

Step 2: boundedness of the gradient. The derivative of ϕ_s is

$$\phi_s'(z) = \sigma(z) - s.$$

Hence, by the chain rule,

$$\nabla_\theta l_{\mathrm{ce}}(\theta;(x,y)) = \phi_s'(z_\theta)\nabla_\theta z_\theta = (\sigma(z_\theta)-s)\nabla_\theta r_\theta(x,y).$$

Since $s \in \{0,1\}$ and $\sigma(z_\theta) \in (0,1)$,

$$|\sigma(z_\theta) - s| \le 1.$$

Therefore,

$$\|\nabla_\theta l_{\mathrm{ce}}(\theta;(x,y))\| \le |\sigma(z_\theta)-s|\,\|\nabla_\theta r_\theta(x,y)\| \le TC.$$

Step 3: boundedness of the Hessian. The second derivative of ϕ_s is

$$\phi_s''(z) = \sigma(z)(1-\sigma(z)).$$

Thus,

$$0 \le \phi_s''(z) \le \frac{1}{4}.$$

Differentiating once more and applying the chain rule gives

$$\nabla^2_{\theta\theta} l_{\mathrm{ce}}(\theta;(x,y)) = \phi_s''(z_\theta)\,\nabla_\theta z_\theta \nabla_\theta z_\theta^\top + \phi_s'(z_\theta)\,\nabla^2_{\theta\theta} z_\theta.$$

Substituting back $z_\theta = r_\theta(x,y)$,

$$\nabla^2_{\theta\theta} l_{\mathrm{ce}}(\theta;(x,y)) = \sigma(z_\theta)(1-\sigma(z_\theta))\nabla_\theta r_\theta(x,y)\nabla_\theta r_\theta(x,y)^\top + (\sigma(z_\theta)-s)\nabla^2_{\theta\theta} r_\theta(x,y).$$

Taking norms and using the triangle inequality,

$$\begin{aligned}
\|\nabla^2_{\theta\theta} l_{\mathrm{ce}}(\theta;(x,y))\| &\le \sigma(z_\theta)(1-\sigma(z_\theta))\|\nabla_\theta r_\theta(x,y)\nabla_\theta r_\theta(x,y)^\top\| + |\sigma(z_\theta)-s|\,\|\nabla^2_{\theta\theta} r_\theta(x,y)\| \\
&\le \frac{1}{4}\|\nabla_\theta r_\theta(x,y)\|^2 + \bar{C} \\
&\le \frac{1}{4}T^2C^2 + T\bar{C}.
\end{aligned}$$

Here we use

$$\|\nabla_\theta r_\theta(x,y)\nabla_\theta r_\theta(x,y)^\top\| = \|\nabla_\theta r_\theta(x,y)\|^2$$

for the operator norm. □

Lemma 6. *Suppose Assumption 1 holds, then it holds that*

$$0 \le l_{\mathrm{dpo}}(\theta;(x,y^+,y^-)) \le \log\big(1+e^{2T\tilde{C}}\big)$$
$$\|\nabla_\theta l_{\mathrm{dpo}}(\theta;(x,y^+,y^-))\| \le 2TC$$
$$\|\nabla^2_{\theta\theta} l_{\mathrm{dpo}}(\theta;(x,y^+,y^-))\| \le T^2C^2 + 2T\bar{C}.$$

Proof. Recall that

$$\Delta_\theta(x,y^+,y^-) = r_\theta(x,y^+) - r_\theta(x,y^-), \qquad l_{\mathrm{dpo}}(\theta;(x,y^+,y^-)) = -\log\sigma\big(\Delta_\theta(x,y^+,y^-)\big).$$

Define

$$\phi(z) \triangleq -\log\sigma(z) = \log(1+e^{-z}),$$

so that

$$l_{\mathrm{dpo}}(\theta;(x,y^+,y^-)) = \phi(\Delta_\theta).$$

Step 1: bounds on Δ_θ. By the assumptions on r_θ,

$$|\Delta_\theta| \le |r_\theta(x,y^+)| + |r_\theta(x,y^-)| \le 2T\tilde{C},$$
$$\|\nabla_\theta\Delta_\theta\| = \|\nabla_\theta r_\theta(x,y^+) - \nabla_\theta r_\theta(x,y^-)\| \le 2TC,$$

and

$$\|\nabla^2_{\theta\theta}\Delta_\theta\| = \|\nabla^2_{\theta\theta} r_\theta(x,y^+) - \nabla^2_{\theta\theta} r_\theta(x,y^-)\| \le 2T\bar{C}.$$

Step 2: boundedness of the loss. Since $\phi(z) = \log(1+e^{-z})$ and $\Delta_\theta \in [-2T\tilde{C},\, 2T\tilde{C}]$, we have

$$0 \le \phi(\Delta_\theta) \le \phi(-2T\tilde{C}) = \log(1+e^{2T\tilde{C}}).$$

Therefore,

$$0 \le l_{\mathrm{dpo}}(\theta;(x,y^+,y^-)) \le \log(1+e^{2T\tilde{C}}).$$

Step 3: boundedness of the gradient. The derivative of ϕ is

$$\phi'(z) = \sigma(z) - 1.$$

Hence, by the chain rule,

$$\nabla_\theta l_{\mathrm{dpo}} = \phi'(\Delta_\theta)\nabla_\theta\Delta_\theta = (\sigma(\Delta_\theta)-1)\nabla_\theta\Delta_\theta.$$

Since $|\sigma(\Delta_\theta)-1| \le 1$,

$$\|\nabla_\theta l_{\mathrm{dpo}}\| \le \|\nabla_\theta\Delta_\theta\| \le 2TC.$$

Step 4: boundedness of the Hessian. The second derivative of ϕ is

$$\phi''(z) = \sigma(z)(1-\sigma(z)), \qquad 0 \le \phi''(z) \le \frac{1}{4}.$$

Applying the chain rule again,

$$\nabla^2_{\theta\theta} l_{\mathrm{dpo}} = \phi''(\Delta_\theta)\,\nabla_\theta\Delta_\theta\nabla_\theta\Delta_\theta^\top + \phi'(\Delta_\theta)\,\nabla^2_{\theta\theta}\Delta_\theta.$$

Therefore,

$$\begin{aligned}\|\nabla^2_{\theta\theta} l_{\mathrm{dpo}}\| &\le \phi''(\Delta_\theta)\,\|\nabla_\theta\Delta_\theta\nabla_\theta\Delta_\theta^\top\| + |\phi'(\Delta_\theta)|\,\|\nabla^2_{\theta\theta}\Delta_\theta\| \\ &\le \frac{1}{4}\|\nabla_\theta\Delta_\theta\|^2 + \|\nabla^2_{\theta\theta}\Delta_\theta\| \\ &\le \frac{1}{4}(2TC)^2 + 2T\bar{C} \\ &= T^2C^2 + 2T\bar{C}.\end{aligned}$$

□

Lemma 7. *Suppose Assumption 1 holds, then it holds that* $||\nabla L(\theta, \pi_{r_\theta})|| \leq (1 + 2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}})+4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}$ *and* $||g_{\text{ce}}(\theta) + \lambda g_{\text{dpo}}(\theta)|| \leq (1 + 2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}})+4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}$ *for any* θ.

Proof. Recall from (9) that

$$g_{\text{ce}}(\theta_k) = \frac{1}{Bm}\sum_{n=1}^{B}\sum_{j=1}^{m}\Big[\nabla_\theta l_{\text{ce}}(\theta_k;(x^{(n)},y^{(n,j)})) + l_{\text{ce}}(\theta_k;(x^{(n)},y^{(n,j)}))\cdot$$
$$\frac{1}{\beta}\sum_{t=1}^{T}\Big(E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t = s_t^{(n,j)}, A_t = a_t^{(n,j)}]$$
$$- E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t = s_t^{(n,j)}]\Big)\Big].$$

Since $|l_{\text{ce}}(\theta;(x,y))| \leq \log(1+e^{T\tilde{C}})$ and $\|\nabla_\theta l_{\text{ce}}(\theta;(x,y))\| \leq TC$ (Lemma 5) and $||\nabla_\theta r(s,a)|| \leq C$ (Lemma 4), we can easily see that $||g_{\text{ce}}(\theta)|| \leq TC + \frac{2\log(1+e^{T\tilde{C}})T^2C}{\beta}$.

Recall from (13) that

$$g_{\text{dpo}}(\theta) = \frac{1}{B}\sum_{n=1}^{B}\frac{1}{\bar{m}^{(n)}(m-\bar{m}^{(n)})}\sum_{\bar{j}=1}^{\bar{m}^{(n)}(m-\bar{m}^{(n)})}\Big[\nabla_\theta l_{\text{dpo}}(\theta_k;(x^{(n)},y^{(n,\bar{j},+)},y^{(n,\bar{j},-)}))$$
$$+ l_{\text{dpo}}(\theta_k;(x^{(n)},y^{(n,\bar{j},+)},y^{(n,\bar{j},-)}))\cdot\frac{1}{\beta}\sum_{t=1}^{T}\Big(E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t = s_t^{(n,\bar{j},+)}, A_t = a_t^{(n,\bar{j},+)}]$$
$$- E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t = s_t^{(n,\bar{j},+)}]$$
$$+ E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t = s_t^{(n,\bar{j},-)}, A_t = a_t^{(n,\bar{j},-)}]$$
$$- E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t = s_t^{(n,\bar{j},-)}]\Big)\Big].$$

Since $|l_{\text{dpo}}(\theta,(x,y^+,y^-))| \leq \log(1+e^{2T\tilde{C}})$ and $||\nabla_\theta l_{\text{dpo}}(\theta,(x,y^+,y^-))|| \leq 2TC$ (Lemma 6) and $||\nabla_\theta r(s,a)|| \leq C$ (Lemma 4), we can easily see that $||g_{\text{dpo}}(\theta)|| \leq 2TC + \frac{4\log(1+e^{2T\tilde{C}})T^2C}{\beta}$.

Therefore, we have that $||g_{\text{ce}}(\theta) + \lambda g_{\text{dpo}}(\theta)|| \leq (1+2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}})+4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}$. Similarly, we can get that $||\nabla L(\theta,\pi_{r_\theta})|| \leq (1+2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}})+4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}$. □

Lemma 8. *Suppose Assumption 1 holds, then* $||\nabla_\theta \log \pi_{r_\theta}(y|x)|| \leq \frac{2CT^2}{\beta}$ *and* $||\nabla^2_{\theta\theta}\log\pi_{r_\theta}(y|x)|| \leq \frac{2\tilde{C}T^2\beta+4C^2T^4}{\beta^2}$.

Proof. From Claim 1, we know that

$$\nabla_\theta \log \pi_{r_\theta}(a_t|s_t) = \nabla_\theta \log Z_{r_\theta}(s_t,a_t) - \nabla_\theta \log Z_{r_\theta}(s_t),$$

where the terms $\nabla_\theta \log Z_{r_\theta}(s_t,a_t) = \frac{1}{\beta}E^{\pi_{r_\theta}}[\sum_{i=t}^{\infty}\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t = s_t, A_t = a_t]$ and $\nabla_\theta \log Z_{r_\theta}(s_t) = \frac{1}{\beta}E^{\pi_{r_\theta}}[\sum_{i=t}^{\infty}\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t = s_t]$. Note that $r_\theta(s_t,a_t) = 0$ for $t > T$, thus we can equivalently write $\nabla_\theta \log Z_{r_\theta}(s_t,a_t) = \frac{1}{\beta}E^{\pi_{r_\theta}}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t = s_t, A_t =$

$a_t]$ and $\nabla_\theta \log Z_{r_\theta}(s_t) = \frac{1}{\beta} E^{\pi_{r_\theta}}[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_t = s_t]$. Therefore, we have

$$||\nabla_\theta \log \pi_{r_\theta}(y|x)|| \leq \sum_{t=1}^{\infty} ||\nabla_\theta \log Z_{r_\theta}(s_t, a_t)|| + ||\nabla_\theta \log Z_{r_\theta}(s_t)|| \leq \frac{2CT^2}{\beta}.$$

Next, we consider the Hessian:

$$\begin{aligned}
&\nabla^2_{\theta\theta} \log Z_{r_\theta}(s,a) = \frac{1}{\beta} \nabla_\theta E^{\pi_{r_\theta}} \Big[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_t = s, A_t = a \Big] \\
&= \frac{1}{\beta} \nabla^2_{\theta\theta} r_\theta(s,a) + \frac{1}{\beta} \nabla_\theta E^{\pi_{r_\theta}} \Big[\sum_{i=t+1}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_{t+1} = s' \Big] \\
&= \frac{1}{\beta} \nabla^2_{\theta\theta} r_\theta(s,a) + \frac{1}{\beta} \nabla_\theta \sum_{a' \in \mathcal{A}} \pi_{r_\theta}(a'|s') E^{\pi_{r_\theta}} \Big[\sum_{i=t+1}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_{t+1} = s', A_{t+1} = a' \Big] \\
&= \frac{1}{\beta} \nabla^2_{\theta\theta} r_\theta(s,a) + \frac{1}{\beta} \sum_{a' \in \mathcal{A}} \nabla_\theta \pi_{r_\theta}(a'|s') \cdot E^{\pi_{r_\theta}} \Big[\sum_{i=t+1}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_{t+1} = s', A_{t+1} = a' \Big] \\
&+ \frac{1}{\beta} \sum_{a' \in \mathcal{A}} \pi_{r_\theta}(a'|s') \cdot \nabla_\theta E^{\pi_{r_\theta}} \Big[\sum_{i=t+1}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_{t+1} = s', A_{t+1} = a' \Big].
\end{aligned}$$

By expanding $\nabla_\theta E^{\pi_{r_\theta}} \Big[\sum_{i=t+1}^{\infty} \gamma^{i-t} \nabla_\theta r_\theta(S_i, A_i) | S_{t+1} = s', A_{t+1} = a' \Big]$, we have

$$\begin{aligned}
&\nabla^2_{\theta\theta} \log Z_{r_\theta}(s,a) = \frac{1}{\beta} E^{\pi_{r_\theta}} \Big[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla^2_{\theta\theta} r_\theta(S_i, A_i) | S_t = s, A_t = a \Big] \\
&+ \frac{1}{\beta} E^{\pi_{r_\theta}} \Big[\sum_{i=t}^{\infty} \gamma^{i-t} \nabla_\theta \pi_{r_\theta}(A_i|S_i) \cdot E^{\pi_{r_\theta}} \Big[\sum_{j=i}^{\infty} \gamma^{j-t} \nabla_\theta r_\theta(S_j, A_j) \Big] \Big| S_t = s, A_t = a \Big].
\end{aligned}$$

Now we take a look at the second term in the above equation:

$$\begin{aligned}
&\nabla_\theta \pi_{r_\theta}(A_i|S_i) \cdot E^{\pi_{r_\theta}} \Big[\sum_{j=i}^{\infty} \gamma^{j-t} \nabla_\theta r_\theta(S_j, A_j) \Big] \\
&= \pi_{r_\theta}(A_i|S_i) \nabla_\theta \log \pi_{r_\theta}(A_i|S_i) \cdot E^{\pi_{r_\theta}} \Big[\sum_{j=i}^{\infty} \gamma^{j-t} \nabla_\theta r_\theta(S_j, A_j) \Big] \\
&\overset{(a)}{=} \pi_{r_\theta}(A_i|S_i) \Big[\nabla_\theta \log Z_{r_\theta}(S_i, A_i) - \nabla_\theta \log Z_{r_\theta}(S_i) \Big] E^{\pi_{r_\theta}} \Big[\sum_{j=i}^{T} \gamma^{j-t} \nabla_\theta r_\theta(S_j, A_j) \Big] \\
&\Rightarrow \Big|\Big| \nabla_\theta \pi_{r_\theta}(A_i|S_i) \cdot E^{\pi_{r_\theta}} \Big[\sum_{j=i}^{\infty} \gamma^{j-t} \nabla_\theta r_\theta(S_j, A_j) \Big] \Big|\Big| \leq \frac{2CT}{\beta} \cdot CT = \frac{2C^2T^2}{\beta}, \qquad (15)
\end{aligned}$$

where (a) follows from Claim 1 and that $r_\theta(s_t, a_t) \equiv 0$ for $t > T$. Therefore, we have

$$\begin{aligned}
||\nabla^2_{\theta\theta} \log Z_{r_\theta}(s,a)|| &\leq \frac{1}{\beta} \Big|\Big| E^{\pi_{r_\theta}} \Big[\sum_{i=t}^{T} \gamma^{i-t} \nabla^2_{\theta\theta} r_\theta(S_i, A_i) | S_t = s, A_t = a \Big] \Big|\Big| \\
&+ \frac{1}{\beta} \Big|\Big| E^{\pi_{r_\theta}} \Big[\sum_{i=t}^{T} \gamma^{i-t} \nabla_\theta \pi_{r_\theta}(A_i|S_i) \cdot E^{\pi_{r_\theta}} \Big[\sum_{j=i}^{T} \gamma^{j-t} \nabla_\theta r_\theta(S_j, A_j) \Big] \Big| S_t = s, A_t = a \Big] \Big|\Big| \\
&\overset{(b)}{\leq} \frac{\bar{C}T}{\beta} + \frac{1}{\beta} \cdot T \cdot \frac{2C^2T^2}{\beta} = \frac{\bar{C}T\beta + 2C^2T^3}{\beta^2},
\end{aligned}$$

where (b) follows from Lemma 4 and (15). Similarly, we can get $||\nabla^2_{\theta\theta} \log Z_{r_\theta}(s)|| \leq \frac{\bar{C}T + 2C^2T^3}{\beta}$. Therefore, we have

$$||\nabla^2_{\theta\theta} \log \pi_{r_\theta}(y|x)|| \leq \sum_{t=1}^{\infty} ||\nabla^2_{\theta\theta} \log \pi_{r_\theta}(a_t|s_t)||$$

$$\leq \sum_{t=1}^{\infty} \Big[||\nabla^2_{\theta\theta} \log Z_{r_\theta}(s,a)|| + ||\nabla^2_{\theta\theta} \log Z_{r_\theta}(s)|| \Big]$$
$$\leq \frac{2\bar{C}T^2\beta + 4C^2T^4}{\beta^2}.$$

□

Lemma 9. *Suppose Assumption 1 holds, then* $||\nabla^2 L(\theta, \pi_{r_\theta})|| \leq C_L$ *where* C_L *is a positive constant whose expression is in (18).*

Proof. Since $L(\theta, \pi_{r_\theta}) = L_{\text{ce}}(\theta, \pi_{r_\theta}) + \lambda L_{\text{dpo}}(\theta, \pi_{r_\theta})$, we compute the Hessian of $L_{\text{ce}}(\theta, \pi_{r_\theta})$ and $L_{\text{dpo}}(\theta, \pi_{r_\theta})$ respectively.

Recall from (6) that

$$\nabla_\theta L_{\text{ce}}(\theta, \pi_{r_\theta}) = E_{x\sim\mathcal{D},\, y\sim\pi_{r_\theta}(\cdot|x)} \Big[l_{\text{ce}}(\theta;(x,y))\, \nabla_\theta \log \pi_{r_\theta}(y|x) + \nabla_\theta l_{\text{ce}}(\theta;(x,y)) \Big].$$

Define

$$\phi(\theta;(x,y)) \triangleq l_{\text{ce}}(\theta;(x,y))\, \nabla_\theta \log \pi_{r_\theta}(y|x) + \nabla_\theta l_{\text{ce}}(\theta;(x,y)).$$

Then

$$\nabla^2_{\theta\theta} L_{\text{ce}}(\theta, \pi_{r_\theta}) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_{r_\theta}(\cdot|x)} \Big[\phi(\theta;(x,y)) \nabla_\theta \log \pi_{r_\theta}(y|x)^\top + \nabla_\theta \phi(\theta;(x,y)) \Big].$$

Expanding the first term,

$$\phi(\theta;(x,y)) \nabla_\theta \log \pi_{r_\theta}(y|x)^\top = l_{\text{ce}}(\theta;(x,y)) \nabla_\theta \log \pi_{r_\theta}(y|x) \nabla_\theta \log \pi_{r_\theta}(y|x)^\top + \nabla_\theta l_{\text{ce}}(\theta;(x,y)) \nabla_\theta \log \pi_{r_\theta}(y|x)^\top.$$

Moreover,

$$\nabla_\theta \phi(\theta;(x,y)) = \nabla_\theta \Big(l_{\text{ce}}(\theta;(x,y))\, \nabla_\theta \log \pi_{r_\theta}(y|x) + \nabla_\theta l_{\text{ce}}(\theta;(x,y)) \Big)$$
$$= \nabla_\theta \log \pi_{r_\theta}(y|x)\, \nabla_\theta l_{\text{ce}}(\theta;(x,y))^\top + l_{\text{ce}}(\theta;(x,y))\, \nabla^2_{\theta\theta} \log \pi_{r_\theta}(y|x) + \nabla^2_{\theta\theta} l_{\text{ce}}(\theta;(x,y)).$$

Therefore,

$$\nabla^2_{\theta\theta} L_{\text{ce}}(\theta, \pi_{r_\theta}) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_{r_\theta}(\cdot|x)} \Big[l_{\text{ce}}(\theta;(x,y)) \nabla_\theta \log \pi_{r_\theta}(y|x) \nabla_\theta \log \pi_{r_\theta}(y|x)^\top + \nabla_\theta l_{\text{ce}}(\theta;(x,y)) \nabla_\theta \log \pi_{r_\theta}(y|x)^\top + \nabla_\theta \log \pi_{r_\theta}(y|x) \nabla_\theta l_{\text{ce}}(\theta;(x,y))^\top + l_{\text{ce}}(\theta;(x,y))\, \nabla^2_{\theta\theta} \log \pi_{r_\theta}(y|x) + \nabla^2_{\theta\theta} l_{\text{ce}}(\theta;(x,y)) \Big].$$

Since $|l_{\text{ce}}(\theta;(x,y))| \leq \log(1+e^{T\tilde{C}})$, $\|\nabla_\theta l_{\text{ce}}(\theta;(x,y))\| \leq TC$, $\|\nabla^2_{\theta\theta} l_{\text{ce}}(\theta;(x,y))\| \leq \frac{1}{4}T^2C^2 + T\bar{C}$ (Lemma 5), and $||\nabla_\theta \log \pi_{r_\theta}(y|x)|| \leq \frac{2CT^2}{\beta}$, $||\nabla^2_{\theta\theta} \log \pi_{r_\theta}(y|x)|| \leq \frac{2\bar{C}T^2\beta + 4C^2T^4}{\beta^2}$ (Lemma 8).

Therefore, we can see that

$$||\nabla^2_{\theta\theta} L_{\text{ce}}(\theta, \pi_{r_\theta})||$$
$$\leq \frac{4\log(1+e^{T\tilde{C}})C^2T^4}{\beta^2} + \frac{4C^2T^3}{\beta} + \frac{(2\bar{C}T^2\beta + 4C^2T^4)\log(1+e^{T\tilde{C}})}{\beta^2} + \frac{1}{4}T^2C^2 + T\bar{C}. \quad (16)$$

Next, we compute the Hessian of $L_{\text{dpo}}(\theta, \pi_{r_\theta})$. Recall from (11) that

$$\nabla_\theta L_{\text{dpo}}(\theta, \pi_{r_\theta})$$
$$= E_{x\sim\mathcal{D},\, (y^+,y^-)\sim\pi_{r_\theta}(\cdot,\cdot|x)} \Big[l_{\text{dpo}}(\theta;(x,y^+,y^-))\, \nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x) + \nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-)) \Big].$$

Define

$$\phi(\theta;(x,y^+,y^-)) \triangleq l_{\text{dpo}}(\theta;(x,y^+,y^-))\, \nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x) + \nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-)).$$

Then,

$$\nabla^2_{\theta\theta} L_{\text{dpo}}(\theta, \pi_{r_\theta}) = \mathbb{E}_{x\sim\mathcal{D},\ (y^+,y^-)\sim\pi_{r_\theta}(\cdot,\cdot|x)}\Big[\phi(\theta;(x,y^+,y^-))\,\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)^\top + \nabla_\theta\phi(\theta;(x,y^+,y^-))\Big].$$

Expanding the first term,

$$\begin{aligned}&\phi(\theta;(x,y^+,y^-))\,\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)^\top\\ &= l_{\text{dpo}}(\theta;(x,y^+,y^-))\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)^\top\\ &+ \nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)^\top.\end{aligned}$$

Moreover,

$$\begin{aligned}\nabla_\theta\phi(\theta;(x,y^+,y^-)) &= \nabla_\theta\Big(l_{\text{dpo}}(\theta;(x,y^+,y^-))\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x) + \nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big)\\ &= \nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)\,\nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))^\top\\ &+ l_{\text{dpo}}(\theta;(x,y^+,y^-))\nabla^2_{\theta\theta}\log \pi_{r_\theta}(y^+,y^-|x) + \nabla^2_{\theta\theta} l_{\text{dpo}}(\theta;(x,y^+,y^-)).\end{aligned}$$

Therefore,

$$\begin{aligned}&\nabla^2_{\theta\theta} L_{\text{dpo}}(\theta, \pi_{r_\theta})\\ &= \mathbb{E}_{x\sim\mathcal{D},\ (y^+,y^-)\sim\pi_{r_\theta}(\cdot,\cdot|x)}\Big[l_{\text{dpo}}(\theta;(x,y^+,y^-))\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)^\top\\ &+ \nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)^\top\\ &+ \nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)\nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))^\top\\ &+ l_{\text{dpo}}(\theta;(x,y^+,y^-))\nabla^2_{\theta\theta}\log \pi_{r_\theta}(y^+,y^-|x) + \nabla^2_{\theta\theta} l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big].\end{aligned}$$

Since $\log \pi_{r_\theta}(y^+,y^-|x) = \log \pi_{r_\theta}(y^+|x) + \log \pi_{r_\theta}(y^-|x)$, then from Lemma 8, we can see that $||\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)|| \le \frac{4CT^2}{\beta}$ and $||\nabla^2_{\theta\theta}\log \pi_{r_\theta}(y^+,y^-|x)|| \le \frac{4\bar{C}T^2\beta+8C^2T^4}{\beta^2}$. From Lemma 6, we know that $|l_{\text{dpo}}(\theta;(x,y^+,y^-))| \le \log\left(1+e^{2T\tilde{C}}\right)$, $\|\nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\| \le 2TC$, and $\|\nabla^2_{\theta\theta} l_{\text{dpo}}(\theta;(x,y^+,y^-))\| \le T^2C^2 + 2T\bar{C}$. Therefore, we can bound the Hessian:

$$\begin{aligned}&||\nabla^2_{\theta\theta} L_{\text{dpo}}(\theta, \pi_{r_\theta})|| \qquad (17)\\ &\le \frac{16\log\left(1+e^{2T\tilde{C}}\right)C^2T^4}{\beta^2} + \frac{16C^2T^3}{\beta} + \frac{(4\bar{C}T^2\beta+8C^2T^4)\log\left(1+e^{2T\tilde{C}}\right)}{\beta^2} + T^2C^2 + 2T\bar{C}.\end{aligned}$$

Combining (16) and (17), we have

$$||\nabla^2 L(\theta, \pi_{r_\theta})|| \le C_L,$$

where

$$\begin{aligned}C_L = &\frac{4\log(1+e^{T\tilde{C}})C^2T^4}{\beta^2} + \frac{4C^2T^3}{\beta} + \frac{(2\bar{C}T^2\beta+4C^2T^4)\log(1+e^{T\tilde{C}})}{\beta^2} + \frac{1}{4}T^2C^2 + T\bar{C} +\\ &\lambda\Big[\frac{16\log\left(1+e^{2T\tilde{C}}\right)C^2T^4}{\beta^2} + \frac{16C^2T^3}{\beta} + \frac{(4\bar{C}T^2\beta+8C^2T^4)\log\left(1+e^{2T\tilde{C}}\right)}{\beta^2} + T^2C^2 + 2T\bar{C}\Big]. \qquad (18)\end{aligned}$$

□

B.4 Proof of Theorem 1 (lower-level convergence)

Theorem 1 includes both lower- and upper-level convergence. We start with proving the lower-level convergence.

We use soft policy iteration as the lower-level policy update for analysis. At each iteration k, we have a policy π_k and a reward function r_{θ_k}. We use a one-step policy update to get π_{k+1} as follows:

$$\pi_{k+1}(a|s) = \frac{\pi_{\text{ref}}(a|s)\exp(\frac{1}{\beta}Q^{\pi_k}_{r_{\theta_k}}(s,a))}{\sum_{\bar{a}\in\mathcal{A}}\pi_{\text{ref}}(\bar{a}|s)\exp(\frac{1}{\beta}Q^{\pi_k}_{r_{\theta_k}}(s,\bar{a}))}$$

$$Q^{\pi_k}_{r_{\theta_k}}(s,a) = r_{\theta_k}(s,a) + \gamma V^{\pi_k}_{r_{\theta_k}}(s')$$
$$V^{\pi_k}_{r_{\theta_k}}(s) = \beta \log\Big[\sum_{\bar{a}\in\mathcal{A}} \pi_{\text{ref}}(\bar{a}|s)\exp(\frac{1}{\beta}Q^{\pi_k}_{r_{\theta_k}}(s,\bar{a}))\Big],$$

where $s' = (s,a)$.

Define the soft Bellman operator

$$(\mathcal{T}_{\theta_k}Q)(s,a) \triangleq r_{\theta_k}(s,a) + \gamma\beta \log\Big[\sum_{\bar{a}'\in\mathcal{A}} \pi_{\text{ref}}(\bar{a}'|s')\exp(\frac{1}{\beta}Q(s',\bar{a}'))\Big].$$

Then a one-step soft policy iteration is equivalently written as

$$Q_{k+1} = \mathcal{T}_{\theta_k}(Q_k), \qquad \pi_k(s,a) \propto \pi_{\text{ref}}(a|s)\exp(\frac{1}{\beta}Q_k(s,a)).$$

Claim 2. *For two action-value functions Q and Q' where $Q, Q' \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, it holds that*

$$||\mathcal{T}_{\theta_k}(Q) - \mathcal{T}_{\theta_k}(Q')||_\infty \le \gamma||Q - Q'||_\infty.$$

Proof. We define $||Q-Q'||_\infty \triangleq \max_{s\in\mathcal{S},a\in\mathcal{A}}|Q(s,a)-Q'(s,a)|$ and denote $\epsilon = ||Q-Q'||_\infty$. Therefore, we have

$$\begin{aligned}\gamma\beta\log\Big[\sum_{a\in\mathcal{A}}\pi_{\text{ref}}(a|s)\exp(\frac{1}{\beta}Q(s,a))\Big] &\le \gamma\beta\log\Big[\sum_{a\in\mathcal{A}}\pi_{\text{ref}}(a|s)\exp(\frac{1}{\beta}(Q'(s,a)+\epsilon))\Big]\\ &= \gamma\epsilon + \gamma\beta\log\Big[\sum_{a\in\mathcal{A}}\pi_{\text{ref}}(a|s)\exp(\frac{1}{\beta}Q'(s,a))\Big].\end{aligned}$$

Similarly, we have

$$\gamma\beta\log\Big[\sum_{a\in\mathcal{A}}\pi_{\text{ref}}(a|s)\exp(\frac{1}{\beta}Q(s,a))\Big] \ge -\gamma\epsilon + \gamma\beta\log\Big[\sum_{a\in\mathcal{A}}\pi_{\text{ref}}(a|s)\exp(\frac{1}{\beta}Q'(s,a))\Big].$$

Therefore, $||\mathcal{T}_{\theta_k}(Q) - \mathcal{T}_{\theta_k}(Q')||_\infty \le \gamma\epsilon = \gamma||Q-Q'||_\infty$. □

Claim 3. *It holds that $Q^{\pi_{k+1}}_{r_{\theta_k}}(s,a) \ge \mathcal{T}_{\theta_k}(Q^{\pi_k}_{r_{\theta_k}}(s,a))$ for any (s,a).*

Proof.

$$\begin{aligned}Q^{\pi_{k+1}}_{r_{\theta_k}}(s,a) &\overset{(a)}{=} r_{\theta_k}(s,a) + \gamma\sum_{a'\in\mathcal{A}}\pi_{k+1}(a'|s')\Big[Q^{\pi_{k+1}}_{r_{\theta_k}}(s',a') - \beta\log\frac{\pi_{k+1}(a'|s')}{\pi_{\text{ref}}(a'|s')}\Big]\\ &\overset{(b)}{\ge} r_{\theta_k}(s,a) + \gamma\sum_{a'\in\mathcal{A}}\pi_{k+1}(a'|s')\Big[Q^{\pi_k}_{r_{\theta_k}}(s',a') - \beta\log\frac{\pi_{k+1}(a'|s')}{\pi_{\text{ref}}(a'|s')}\Big]\\ &= r_{\theta_k}(s,a) + \gamma\beta\log\Big[\sum_{a'\in\mathcal{A}}\pi_{\text{ref}}(a'|s')\exp(\frac{1}{\beta}Q^{\pi_k}_{r_{\theta_k}}(s',a'))\Big]\\ &= \mathcal{T}_{\theta_k}(Q^{\pi_k}_{r_{\theta_k}}(s,a)),\end{aligned}$$

where (a) follows from equations (2)-(3) in (Haarnoja et al., 2018) and (b) follows from policy improvement theorem (Theorem 4 in (Haarnoja et al., 2017)). □

Claim 4. *Suppose Assumption 1 holds. For an arbitrary policy π and reward parameters θ_1 and θ_2, it holds that*

$$|Q^*_{r_{\theta_1}}(s,a) - Q^*_{r_{\theta_2}}(s,a)| \le \hat{C}||\theta_1-\theta_2||$$
$$|Q^\pi_{r_{\theta_1}}(s,a) - Q^\pi_{r_{\theta_2}}(s,a)| \le \hat{C}||\theta_1-\theta_2||,$$

for any $(s,a) \in \mathcal{S}\times\mathcal{A}$, and $\hat{C} = CT$ is a positive constant.

Proof. Recall from the proof of Claim 1 that $Z_{r_\theta}(s,a) \triangleq \pi_{\text{ref}}(a|s)\exp(\frac{1}{\beta}Q^*_{r_\theta}(s,a))$ and $\nabla_\theta \log Z_{r_\theta}(s,a) = \frac{1}{\beta}E^{\pi_{r_{\theta_k}}}[\sum_{i=t}^{\infty}\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s,A_t=a]$. Therefore, we have

$$\begin{aligned}\nabla_\theta Q^\pi_{r_\theta}(s,a) &= \beta\nabla_\theta \log Z^\pi_{r_\theta}(s,a)\\ &= E^\pi[\sum_{i=t}^{\infty}\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s,A_t=a]\\ &\overset{(c)}{\leq} CT,\end{aligned}$$

where (c) follows from Lemma 4 that $||\nabla_\theta r_\theta(s,a)|| \leq C$ and that $r_\theta(s_t,a_t)\equiv 0$ for $t>T$.

The action value function can be equivalently reformulated as (Haarnoja et al., 2017):

$$Q^\pi_r(s,a) = E^\pi\Big[\sum_{i=t}^{\infty}\gamma^{i-t}r(S_i,A_i) - \beta\log\frac{\pi(A_i|S_i)}{\pi_{\text{ref}}(A_i|S_i)}\Big|S_t=s,A_t=a\Big]. \tag{19}$$

Therefore, we have

$$\begin{aligned}&\Big|Q^\pi_{r_{\theta_1}}(s,a) - Q^\pi_{r_{\theta_2}}(s,a)\Big|\\ &\overset{(d)}{=} \Big|E^\pi\Big[\sum_{i=t}^{\infty}\gamma^{i-t}r_{\theta_1}(S_i,A_i) - \beta\log\frac{\pi(A_i|S_i)}{\pi_{\text{ref}}(A_i|S_i)}\Big|S_t=s,A_t=a\Big]\\ &- E^\pi\Big[\sum_{i=t}^{\infty}\gamma^{i-t}r_{\theta_2}(S_i,A_i) - \beta\log\frac{\pi(A_i|S_i)}{\pi_{\text{ref}}(A_i|S_i)}\Big|S_t=s,A_t=a\Big]\Big|\\ &= \Big|E^\pi\Big[\sum_{i=t}^{\infty}\gamma^{i-t}r_{\theta_1}(S_i,A_i) - r_{\theta_2}(S_i,A_i)\Big|S_t=s,A_t=a\Big]\Big|\\ &\overset{(e)}{\leq} CT||\theta_1-\theta_2||,\end{aligned}$$

where (d) follows from (19) and (e) follows from Lemma 4. Note that on a convex domain, $|f(x_1)-f(x_2)|\leq L||x_1-x_2||$ and $||\nabla f(x)||\leq L$ are equivalent (Nesterov, 2013; Bubeck, 2015), where L is a positive constant. □

Now, we proceed to quantify the convergence of the lower-level policy in Algorithm 1. For any $(s,a)\in\mathcal{S}\times\mathcal{A}$, we know that

$$\begin{aligned}&|\log\pi_{k+1}(a|s) - \log\pi_{r_{\theta_k}}(a|s)|\\ &\leq \frac{1}{\beta}(|Q^{\pi_k}_{r_{\theta_k}}(s,a) - Q^*_{r_{\theta_k}}(s,a)| + |V^{\pi_k}_{r_{\theta_k}}(s) - V^*_{r_{\theta_k}}(s)|)\\ &\overset{(f)}{\leq} \frac{1}{\beta}(|Q^{\pi_k}_{r_{\theta_k}}(s,a) - Q^*_{r_{\theta_k}}(s,a)| + \max_{(s,a)\in\mathcal{S}\times\mathcal{A}}\{|Q^{\pi_k}_{r_{\theta_k}}(s,a) - Q^*_{r_{\theta_k}}(s,a)|\})\\ &\leq \frac{2}{\beta}||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty,\end{aligned} \tag{20}$$

where (f) follows from equation (48) in (Zeng et al., 2022).

Now we look at the term $||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty$:

$$\begin{aligned}||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty &\leq ||Q^{\pi_k}_{r_{\theta_k}} - Q^{\pi_k}_{r_{\theta_{k-1}}}||_\infty + ||Q^{\pi_k}_{r_{\theta_{k-1}}} - Q^*_{r_{\theta_{k-1}}}||_\infty + ||Q^*_{r_{\theta_{k-1}}} - Q^*_{r_{\theta_k}}||_\infty\\ &\overset{(g)}{\leq} \hat{C}||\theta_{k-1}-\theta_k|| + ||Q^{\pi_k}_{r_{\theta_{k-1}}} - Q^*_{r_{\theta_{k-1}}}||_\infty + \hat{C}||\theta_{k-1}-\theta_k||\\ &\overset{(h)}{=} 2\hat{C}||\theta_{k-1}-\theta_k|| + Q^*_{r_{\theta_{k-1}}} - Q^{\pi_k}_{r_{\theta_{k-1}}}\\ &\overset{(i)}{\leq} 2\hat{C}||\theta_{k-1}-\theta_k|| + Q^*_{r_{\theta_{k-1}}} - \mathcal{T}_{\theta_{k-1}}(Q^{\pi_{k-1}}_{r_{\theta_{k-1}}})\end{aligned}$$

$$\overset{(j)}{=} 2\hat{C}||\theta_{k-1}-\theta_k|| + \mathcal{T}_{\theta_{k-1}}(Q^*_{r_{\theta_{k-1}}}) - \mathcal{T}_{\theta_{k-1}}(Q^{\pi_{k-1}}_{r_{\theta_{k-1}}})$$
$$\overset{(k)}{\leq} 2\hat{C}||\theta_{k-1}-\theta_k|| + \gamma||Q^*_{r_{\theta_{k-1}}} - Q^{\pi_{k-1}}_{r_{\theta_{k-1}}}||_\infty, \tag{21}$$

where (g) follows from Claim 4, (h) follows from the fact that $Q^*_{r_{\theta_{k-1}}}(s,a) \geq Q^{\pi_k}_{r_{\theta_{k-1}}}(s,a)$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$ because $Q^*_{r_{\theta_{k-1}}}$ is the optimal action-value function under $r_{\theta_{k-1}}$, (i) follows from Claim 3, (j) follows from the fact that $Q^*_{r_{\theta_{k-1}}} = \mathcal{T}_{\theta_{k-1}}(Q^*_{r_{\theta_{k-1}}})$ because $Q^*_{r_{\theta_{k-1}}}$ is the fixed point of the contraction map $\mathcal{T}_{\theta_{k-1}}$ (Haarnoja et al., 2017), and (k) follows from Claim 2. Therefore, we have

$$\sum_{k=1}^{K-1} ||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty$$
$$\leq 2\hat{C}\sum_{k=1}^{K-1}||\theta_{k-1}-\theta_k|| + \gamma\sum_{k=1}^{K-1}||Q^*_{r_{\theta_{k-1}}} - Q^{\pi_{k-1}}_{r_{\theta_{k-1}}}||_\infty$$
$$= 2\hat{C}\sum_{k=1}^{K-1}\alpha||g_{\text{ce}}(\theta_{k-1}) + \lambda g_{\text{dpo}}(\theta_{k-1})||_\infty + \gamma\sum_{k=1}^{K-1}||Q^*_{r_{\theta_{k-1}}} - Q^{\pi_{k-1}}_{r_{\theta_{k-1}}}||_\infty$$
$$\overset{(l)}{\leq} 2\hat{C}\sum_{k=1}^{K-1}\alpha\Big[(1+2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}}) + 4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}\Big]$$
$$+\gamma\sum_{k=1}^{K-1}||Q^*_{r_{\theta_{k-1}}} - Q^{\pi_{k-1}}_{r_{\theta_{k-1}}}||_\infty,$$

where (l) follows from Lemma 7. By subtracting $\gamma\sum_{k=1}^{K-1}||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty$ on both sides, we have

$$(1-\gamma)\sum_{k=1}^{K-1}||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty$$
$$\leq 2\hat{C}\Big[(1+2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}}) + 4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}\Big]\sum_{k=1}^{K}\alpha$$
$$+\gamma\Big(||Q^*_{r_{\theta_0}} - Q^{\pi_0}_{r_{\theta_0}}||_\infty - ||Q^*_{r_{\theta_{K-1}}} - Q^{\pi_{K-1}}_{r_{\theta_{K-1}}}||_\infty\Big).$$

Dividing both sides by $(1-\gamma)$, we have

$$\sum_{k=0}^{K-1}||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty = ||Q^{\pi_0}_{r_{\theta_0}} - Q^*_{r_{\theta_0}}||_\infty + \sum_{k=1}^{K-1}||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty$$
$$\leq \frac{2\hat{C}}{1-\gamma}\Big[(1+2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}}) + 4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}\Big]\sum_{k=1}^{K}\alpha$$
$$+\frac{\gamma}{(1-\gamma)}\Big(||Q^*_{r_{\theta_0}} - Q^{\pi_0}_{r_{\theta_0}}||_\infty - ||Q^*_{r_{\theta_K}} - Q^{\pi_K}_{r_{\theta_K}}||_\infty\Big) + ||Q^{\pi_0}_{r_{\theta_0}} - Q^*_{r_{\theta_0}}||_\infty. \tag{22}$$

Finally, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}|\log\pi_{k+1} - \log\pi_{r_{\theta_k}}||_\infty$$
$$\overset{(m)}{\leq} \frac{2}{\beta K}\sum_{k=0}^{K-1}||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty$$
$$\overset{(n)}{\leq} \frac{4\hat{C}}{(1-\gamma)\beta K}\Big[(1+2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}}) + 4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}\Big]\sum_{k=1}^{K}K^{-1/2}$$

$$+\frac{2\gamma}{(1-\gamma)\beta K}\Big(||Q^*_{r_{\theta_0}}-Q^{\pi_0}_{r_{\theta_0}}||_\infty-||Q^*_{r_{\theta_K}}-Q^{\pi_K}_{r_{\theta_K}}||_\infty\Big)+\frac{1}{K}||Q^{\pi_0}_{r_{\theta_0}}-Q^*_{r_{\theta_0}}||_\infty$$
$$=O(1/\sqrt{K})+O(1/K)=O(1/\sqrt{K}),$$

where (m) follows from (20) and (n) follows from (22).

B.5 Proof of Theorem 1 (upper-level convergence)

We next quantify the convergence of the upper level. For simplicity, we use $L(\theta)$ to represent $L(\theta,\pi_{r_\theta})$ and $g(\theta)$ to represent $g_{\text{ce}}(\theta)+\lambda g_{\text{dpo}}(\theta)$:

$$\begin{aligned}L(\theta_{k+1})&\overset{(a)}{\leq}L(\theta_k)+\langle\nabla L(\theta_k),\theta_{k+1}-\theta_k\rangle+\frac{C_L}{2}||\theta_{k+1}-\theta_k||^2\\&=L(\theta_k)-\alpha\langle\nabla L(\theta_k),g(\theta_k)\rangle+\frac{C_L\alpha^2}{2}||g(\theta_k)||^2\\&=L(\theta_k)-\alpha\langle\nabla L(\theta_k),E[g(\theta_k)-\nabla L(\theta_k)]\rangle-\alpha||\nabla L(\theta_k)||^2+\frac{C_L\alpha^2}{2}||g(\theta_k)||^2\\&\overset{(b)}{\leq}L(\theta_k)-\alpha\langle\nabla L(\theta_k),E[g(\theta_k)-\nabla L(\theta_k)]\rangle-\alpha||\nabla L(\theta_k)||^2+\frac{C_LC_g\alpha^2}{2},\end{aligned}$$

where (a) follows from Lemma 9 and (b) follows from Lemma 7 such that $C_g\triangleq[(1+2\lambda)TC+\frac{[2\log(1+e^{T\tilde{C}})+4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}]^2$. Taking the expectation on both sides, we have

$$E[L(\theta_{k+1})]\leq E[L(\theta_k)]-\alpha E[\langle\nabla L(\theta_k),E[g(\theta_k)-\nabla L(\theta_k)|\theta_k]\rangle]-\alpha E[||\nabla L(\theta_k)||^2]+\frac{C_LC_g\alpha^2}{2}. \tag{23}$$

We next start from the following claim, which is proved in the literature:

Claim 5 (Inequality (64) in (Zeng et al., 2022))**.** *It holds that* $\Big|\Big|E_{x\sim\mathcal{D},y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[\sum_{i=t}^\infty\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)\Big]-E_{x\sim\mathcal{D},y\sim\pi_k(\cdot|x)}\Big[\sum_{i=t}^\infty\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)\Big]\Big|\Big|\leq CC_d\sqrt{|\mathcal{S}|\times|\mathcal{A}|}E\Big[||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||_\infty\Big]$ *where* $||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||_\infty\triangleq\max_{(s,a)\in\mathcal{S}\times\mathcal{A}}\{|Q^{\pi_k}_{r_{\theta_k}}(s,a)-Q^*_{r_{\theta_k}}(s,a)|\}$.

Based on Claim 5, we can see that

$$\begin{aligned}&\Big|\Big|E_{x\sim\mathcal{D},y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[\nabla_\theta\log\pi_{r_\theta}(y|x)\Big]-E_{x\sim\mathcal{D},y\sim\pi_k(\cdot|x)}\Big[\nabla_\theta\log\pi_{r_\theta}(y|x)\Big]\Big|\Big|\\&\leq\sum_{t=1}^\infty\Big|\Big|E_{x\sim\mathcal{D},y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[\nabla_\theta\log\pi_{r_\theta}(a_t|s_t)\Big]-E_{x\sim\mathcal{D},y\sim\pi_k(\cdot|x)}\Big[\nabla_\theta\log\pi_{r_\theta}(a_t|s_t)\Big]\Big|\Big|\\&\overset{(c)}{=}\sum_{t=1}^\infty\Big|\Big|E_{x\sim\mathcal{D},y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[E^{\pi_{r_\theta}}\Big[\sum_{i=t}^\infty\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s_t,A_t=a_t\Big]\\&-E^{\pi_{r_\theta}}\Big[\sum_{i=t}^\infty\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s_t\Big]\Big]\\&-E_{x\sim\mathcal{D},y\sim\pi_k(\cdot|x)}\Big[E^{\pi_{r_\theta}}\Big[\sum_{i=t}^\infty\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s_t,A_t=a_t\Big]\\&-E^{\pi_{r_\theta}}\Big[\sum_{i=t}^\infty\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s_t\Big]\Big]\Big|\Big|\\&\leq\sum_{t=1}^\infty\Big\{\Big|\Big|E_{x\sim\mathcal{D},y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[E^{\pi_{r_\theta}}\Big[\sum_{i=t}^\infty\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s_t,A_t=a_t\Big]\Big]\\&-E_{x\sim\mathcal{D},y\sim\pi_k(\cdot|x)}\Big[E^{\pi_{r_\theta}}\Big[\sum_{i=t}^\infty\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s_t,A_t=a_t\Big]\Big]\Big|\Big|\end{aligned}$$

$$
\begin{aligned}
&+ \Big|\Big| E_{x\sim\mathcal{D}, y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[E^{\pi_{r_\theta}}\Big[\sum_{i=t}^{\infty}\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s_t\Big]\Big] \\
&- E_{x\sim\mathcal{D}, y\sim\pi_k(\cdot|x)}\Big[E^{\pi_{r_\theta}}\Big[\sum_{i=t}^{\infty}\gamma^{i-t}\nabla_\theta r_\theta(S_i,A_i)|S_t=s_t\Big]\Big]\Big|\Big|\Big\} \\
&\overset{(d)}{\leq} 2TCC_d\sqrt{|\mathcal{S}|\times|\mathcal{A}|}E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty\Big], \qquad (24)
\end{aligned}
$$

where (c) follows from Claim 1 and (d) follows from Claim 5 and that $r_\theta(s_t,a_t)\equiv 0$ for $t>T$.

Now, we look at the second term in (23):

$$
\begin{aligned}
&E[\langle \nabla L(\theta_k), E[g(\theta_k)-\nabla L(\theta_k)|\theta_k]\rangle] \\
&= E[\langle \nabla L(\theta_k), E[(g_{\text{ce}}(\theta_k)-\nabla L_{\text{ce}}(\theta_k)) + \lambda(g_{\text{dpo}}(\theta_k)-\nabla L_{\text{dpo}}(\theta_k))|\theta_k\rangle]] \\
&\overset{(e)}{\leq} C_{\nabla L}E\Big[\Big\|E[g_{\text{ce}}(\theta_k)-\nabla L_{\text{ce}}(\theta_k))]\Big\| + \lambda\Big\|E[g_{\text{dpo}}(\theta_k)-\nabla L_{\text{dpo}}(\theta_k)]\Big\|\Big], \qquad (25)
\end{aligned}
$$

where (e) follows from Lemma 7 such that $C_{\nabla L} = (1+2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}})+4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}$.

Next, we bound the first and second terms in (25), respectively. We start with the first term:

$$
\begin{aligned}
&E\Big[\Big\|E[g_{\text{ce}}(\theta_k)-\nabla L_{\text{ce}}(\theta_k))]\Big\|\Big] \\
&= \Big\|E_{x\sim\mathcal{D}, y\sim\pi_k(\cdot|x)}\Big[l_{\text{ce}}(\theta;(x,y))\,\nabla_\theta\log\pi_{r_\theta}(y|x) + \nabla_\theta l_{\text{ce}}(\theta;(x,y))\Big] \\
&- E_{x\sim\mathcal{D}, y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[l_{\text{ce}}(\theta;(x,y))\,\nabla_\theta\log\pi_{r_\theta}(y|x) + \nabla_\theta l_{\text{ce}}(\theta;(x,y))\Big]\Big\| \\
&\leq \Big\|E_{x\sim\mathcal{D}, y\sim\pi_k(\cdot|x)}\Big[l_{\text{ce}}(\theta;(x,y))\,\nabla_\theta\log\pi_{r_\theta}(y|x)\Big] \\
&- E_{x\sim\mathcal{D}, y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[l_{\text{ce}}(\theta;(x,y))\,\nabla_\theta\log\pi_{r_\theta}(y|x)\Big]\Big\| \\
&+ \Big\|E_{x\sim\mathcal{D}, y\sim\pi_k(\cdot|x)}\Big[\nabla_\theta l_{\text{ce}}(\theta;(x,y))\Big] - E_{x\sim\mathcal{D}, y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[\nabla_\theta l_{\text{ce}}(\theta;(x,y))\Big]\Big\| \\
&\overset{(f)}{\leq} \log(1+e^{T\tilde{C}})\Big\|E_{x\sim\mathcal{D}, y\sim\pi_k(\cdot|x)}\Big[\nabla_\theta\log\pi_{r_\theta}(y|x)\Big] - E_{x\sim\mathcal{D}, y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[\nabla_\theta\log\pi_{r_\theta}(y|x)\Big]\Big\| \\
&+ \Big\|E_{x\sim\mathcal{D}, y\sim\pi_k(\cdot|x)}\Big[\nabla_\theta l_{\text{ce}}(\theta;(x,y))\Big] - E_{x\sim\mathcal{D}, y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[\nabla_\theta l_{\text{ce}}(\theta;(x,y))\Big]\Big\| \\
&\overset{(g)}{\leq} \log(1+e^{T\tilde{C}})\cdot 2TCC_d\sqrt{|\mathcal{S}|\times|\mathcal{A}|}E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty\Big] \\
&+ \Big\|E_{x\sim\mathcal{D}, y\sim\pi_k(\cdot|x)}\Big[\nabla_\theta l_{\text{ce}}(\theta;(x,y))\Big] - E_{x\sim\mathcal{D}, y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[\nabla_\theta l_{\text{ce}}(\theta;(x,y))\Big]\Big\| \\
&\overset{(h)}{\leq} \log(1+e^{T\tilde{C}})\cdot 2TCC_d\sqrt{|\mathcal{S}|\times|\mathcal{A}|}E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty\Big] \\
&+ \Big\|E_{x\sim\mathcal{D}, y\sim\pi_k(\cdot|x)}\Big[\nabla_\theta r_\theta(x,y)\Big] - E_{x\sim\mathcal{D}, y\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[\nabla_\theta r_\theta(x,y)\Big]\Big\| \\
&\overset{(i)}{\leq} [\log(1+e^{T\tilde{C}})\cdot 2TCC_d + CC_d]\sqrt{|\mathcal{S}|\times|\mathcal{A}|}E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty\Big], \qquad (26)
\end{aligned}
$$

where (f) follows from Lemma 5, (g) follows from (24), (h) follows from the expression (10) of $\nabla_\theta l_{\text{ce}}$, and (i) follows from Claim 5.

Next, we bound the second term in (25):

$$
\begin{aligned}
&E\Big[\Big\|E[g_{\text{dpo}}(\theta_k)-\nabla L_{\text{dpo}}(\theta_k))]\Big\|\Big] \\
&= \Big\|E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_{r_{\theta_k}}(\cdot,\cdot|x)}\Big[l_{\text{dpo}}(\theta;(x,y^+,y^-))\,\nabla_\theta\log\pi_{r_\theta}(y^+,y^-|x) + \nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big] \\
&- E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_k(\cdot,\cdot|x)}\Big[l_{\text{dpo}}(\theta;(x,y^+,y^-))\,\nabla_\theta\log\pi_{r_\theta}(y^+,y^-|x) + \nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big]\Big\| \\
&\leq \Big\|E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_{r_{\theta_k}}(\cdot,\cdot|x)}\Big[l_{\text{dpo}}(\theta;(x,y^+,y^-))\,\nabla_\theta\log\pi_{r_\theta}(y^+,y^-|x)\Big]
\end{aligned}
$$

$$
\begin{aligned}
&- E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_k(\cdot,\cdot|x)}\Big[l_{\text{dpo}}(\theta;(x,y^+,y^-))\,\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)\Big]\Big\| \\
&+ \Big\|E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_{r_{\theta_k}}(\cdot,\cdot|x)}\Big[\nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big] \\
&- E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_k(\cdot,\cdot|x)}\Big[\nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big]\Big\| \\
&\overset{(j)}{\leq} \log\left(1+e^{2T\tilde{C}}\right)\Big\|E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_{r_{\theta_k}}(\cdot,\cdot|x)}\Big[\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)\Big] \\
&- E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_k(\cdot,\cdot|x)}\Big[\nabla_\theta \log \pi_{r_\theta}(y^+,y^-|x)\Big]\Big\| \\
&+ \Big\|E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_{r_{\theta_k}}(\cdot,\cdot|x)}\Big[\nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big] \\
&- E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_k(\cdot,\cdot|x)}\Big[\nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big]\Big\| \\
&\overset{(k)}{\leq} \log\left(1+e^{2T\tilde{C}}\right)\cdot 4TCC_d\sqrt{|\mathcal{S}|\times|\mathcal{A}|}E\Big[||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||_\infty\Big] \\
&+ \Big\|E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_{r_{\theta_k}}(\cdot,\cdot|x)}\Big[\nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big] \\
&- E_{x\sim\mathcal{D},\,(y^+,y^-)\sim\pi_k(\cdot,\cdot|x)}\Big[\nabla_\theta l_{\text{dpo}}(\theta;(x,y^+,y^-))\Big]\Big\| \\
&\overset{(l)}{\leq} \log\left(1+e^{2T\tilde{C}}\right)\cdot 4TCC_d\sqrt{|\mathcal{S}|\times|\mathcal{A}|}E\Big[||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||_\infty\Big] \\
&+ \Big\|E_{x\sim\mathcal{D},(y^+y^-)\sim\pi_k(\cdot|x)}\Big[\nabla_\theta r_\theta(x,y^+)-\nabla_\theta r_\theta(x,y^-)\Big] \\
&- E_{x\sim\mathcal{D},(y^+y^-)\sim\pi_{r_{\theta_k}}(\cdot|x)}\Big[\nabla_\theta r_\theta(x,y^+)-\nabla_\theta r_\theta(x,y^-)\Big]\Big\| \\
&\overset{(m)}{\leq} [\log\left(1+e^{2T\tilde{C}}\right)\cdot 4TCC_d\cdot 2CC_d]\sqrt{|\mathcal{S}|\times|\mathcal{A}|}E\Big[||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||_\infty\Big], \qquad (27)
\end{aligned}
$$

where (j) follows from Lemma 6, (k) follows from (24) and the fact that $\log\pi(y^+,y^-|x) = \log\pi(y^+|x)+\log\pi(y^-|x)$, (l) follows from the expression (14) of $\nabla_\theta l_{\text{dpo}}$, and (m) follows from Claim 5.

By plugging (26) and (27) into (25), we have

$$
E[\langle\nabla L(\theta_k), E[g(\theta_k)-\nabla L(\theta_k)|\theta_k]\rangle] \leq C_\Delta E\Big[||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||_\infty\Big], \qquad (28)
$$

where $C_\Delta = [\log(1+e^{T\tilde{C}})\cdot 2TCC_d + CC_d]\sqrt{|\mathcal{S}|\times|\mathcal{A}|} + \lambda[\log\left(1+e^{2T\tilde{C}}\right)\cdot 4TCC_d 2CC_d]\sqrt{|\mathcal{S}|\times|\mathcal{A}|}$.

Recall from (23) that

$$
\begin{aligned}
&E[L(\theta_{k+1})] \leq E[L(\theta_k)] - \alpha E[\langle\nabla L(\theta_k), E[g(\theta_k)-\nabla L(\theta_k)|\theta_k]\rangle] - \alpha E[||\nabla L(\theta_k)||^2] + \frac{C_L C_g\alpha^2}{2} \\
&\overset{(n)}{\leq} E[L(\theta_k)] + \alpha C_{\nabla_L} C_\Delta E\Big[||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||_\infty\Big] - \alpha E[||\nabla L(\theta_k)||^2] + \frac{C_L C_g\alpha^2}{2},
\end{aligned}
$$

where (n) follows from Lemma 7 and (28). Therefore, we have

$$
\alpha E[||\nabla L(\theta_k)||^2] \leq E[L(\theta_k)] - E[L(\theta_{k+1})] + \alpha C_{\nabla_L} C_\Delta E\Big[||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||_\infty\Big] + \frac{C_L C_g\alpha^2}{2}.
$$

Telescoping from 0 to $K-1$ on both sides, we have

$$
\begin{aligned}
&\alpha\sum_{k=0}^{K-1} E[||\nabla L(\theta_k)||^2] \\
&\leq E[L(\theta_0)] - E[L(\theta_K)] + \alpha C_{\nabla_L} C_\Delta \sum_{k=0}^{K-1} E\Big[||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||_\infty\Big] + \sum_{k=0}^{K-1}\frac{C_L C_g\alpha^2}{2}.
\end{aligned}
$$

Dividing both sides by αK, we have

$$\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1} E[||\nabla L(\theta_k)||^2] \\
&\leq \frac{1}{\alpha K}\Big[E[L(\theta_0)] - E[L(\theta_K)]\Big] + \frac{C_{\nabla_L} C_\Delta}{K}\sum_{k=0}^{K-1} E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||_\infty\Big] + \frac{C_L C_g \alpha}{2} \\
&\overset{(o)}{\leq} \frac{1}{\alpha K}\Big[E[L(\theta_0)] - E[L(\theta_K)]\Big] \\
&+ \frac{2\hat{C} C_{\nabla_L} C_\Delta}{1-\gamma}\Big[(1+2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}}) + 4\lambda\log(1+e^{2T\tilde{C}})]T^2 C}{\beta}\Big]\cdot\frac{1}{K}\sum_{k=1}^{K}\alpha \\
&+ \frac{\gamma C_{\nabla_L} C_\Delta}{(1-\gamma)K}\Big(||Q^*_{r_{\theta_0}} - Q^{\pi_0}_{r_{\theta_0}}||_\infty - ||Q^*_{r_{\theta_K}} - Q^{\pi_K}_{r_{\theta_K}}||_\infty\Big) + \frac{C_{\nabla_L} C_\Delta}{K}||Q^{\pi_0}_{r_{\theta_0}} - Q^*_{r_{\theta_0}}||_\infty + \frac{C_L C_g \alpha}{2} \\
&= O(\frac{1}{\sqrt{K}}) + O(\frac{1}{K}),
\end{aligned} \tag{29}$$

where (o) follows from (22).

Note that $\nabla L(\theta_k) = \nabla L(\theta_k, \pi_{r_{\theta_k}})$ and we need to upper bound $||\nabla L(\theta_k, \pi_k)||$. Therefore, we need to bound the difference:

$$\begin{aligned}
&E[||\nabla L(\theta_k, \pi_k) - \nabla L(\theta_k)||^2] \\
&\leq E[2||\nabla L_{\text{ce}}(\theta_k, \pi_k) - \nabla L_{\text{ce}}(\theta_k)||^2 + 2\lambda^2||\nabla L_{\text{dpo}}(\theta_k, \pi_k) - \nabla L_{\text{dpo}}(\theta_k)||^2] \\
&\overset{(p)}{\leq} C_1 E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||^2_\infty\Big],
\end{aligned} \tag{30}$$

where (p) follows from (26) and (27) such that $C_1 = 2\Big[\Big(\log(1+e^{T\tilde{C}})\cdot 2TCC_d + CC_d\Big)^2 + \lambda^2\Big(\log\big(1+e^{2T\tilde{C}}\big)\cdot 4TCC_d 2CC_d\Big)^2\Big]\Big(|\mathcal{S}|\times|\mathcal{A}|\Big)$.

Telescoping $E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||^2_\infty\Big]$ from 0 to $K-1$, we have

$$\begin{aligned}
&\sum_{k=0}^{K-1} E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||^2_\infty\Big] = E\Big[||Q^{\pi_0}_{r_{\theta_0}} - Q^*_{r_{\theta_0}}||^2_\infty\Big] + \sum_{k=1}^{K-1} E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||^2_\infty\Big] \\
&\overset{(q)}{\leq} E\Big[||Q^{\pi_0}_{r_{\theta_0}} - Q^*_{r_{\theta_0}}||^2_\infty\Big] + \sum_{k=1}^{K-1} 4(1+\epsilon)\hat{C}^2 E[||\theta_{k-1} - \theta_k||^2] + (1+1/\epsilon)\gamma^2 E[||Q^*_{r_{\theta_{k-1}}} - Q^{\pi_{k-1}}_{r_{\theta_{k-1}}}||^2_\infty] \\
&= E\Big[||Q^{\pi_0}_{r_{\theta_0}} - Q^*_{r_{\theta_0}}||^2_\infty\Big] + \sum_{k=1}^{K-1} 4(1+\epsilon)\hat{C}^2\alpha^2 E\Big[||g(\theta_{k-1})||^2\Big] + (1+1/\epsilon)\gamma^2 E\Big[||Q^*_{r_{\theta_{k-1}}} - Q^{\pi_{k-1}}_{r_{\theta_{k-1}}}||^2_\infty\Big] \\
&\overset{(r)}{\leq} \sum_{k=1}^{K-1} 4(1+\epsilon)\hat{C}^2\alpha^2\Big[(1+2\lambda)TC + \frac{[2\log(1+e^{T\tilde{C}}) + 4\lambda\log(1+e^{2T\tilde{C}})]T^2 C}{\beta}\Big]^2 \\
&+ (1+1/\epsilon)\gamma^2 E[||Q^*_{r_{\theta_{k-1}}} - Q^{\pi_{k-1}}_{r_{\theta_{k-1}}}||^2_\infty] + E\Big[||Q^{\pi_0}_{r_{\theta_0}} - Q^*_{r_{\theta_0}}||^2_\infty\Big],
\end{aligned}$$

where (q) follows from (21) and Young's inequality, and (r) follows from Lemma 7. Subtracting both sides by $(1+1/\epsilon)\sum_{k=0}^{K-1}\gamma^2 E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||^2_\infty\Big]$, we have

$$\begin{aligned}
&(1-(1+1/\epsilon)\gamma^2)\sum_{k=0}^{K-1} E\Big[||Q^{\pi_k}_{r_{\theta_k}} - Q^*_{r_{\theta_k}}||^2_\infty\Big] \\
&\leq E[||Q^*_{r_{\theta_0}} - Q^{\pi_0}_{r_{\theta_0}}||^2_\infty] - (1+1/\epsilon)\gamma^2 E[||Q^*_{r_{\theta_{K-1}}} - Q^{\pi_{K-1}}_{r_{\theta_{K-1}}}||^2_\infty]
\end{aligned}$$

$$+\sum_{k=1}^{K-1} 4(1+\epsilon)\hat{C}^2\alpha^2\Big[(1+2\lambda)TC+\frac{[2\log(1+e^{T\tilde{C}})+4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}\Big]^2. \tag{31}$$

Note that for any $\gamma \in (0,1)$, we can always choose ϵ such that $(1-(1+1/\epsilon)\gamma^2) > 0$. Therefore, we can see that

$$\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1} E[||\nabla L(\theta_k,\pi_k)-\nabla L(\theta_k)||^2] \\
&\overset{(s)}{\leq} \frac{1}{K}\sum_{k=0}^{K-1} C_1 E\Big[||Q^{\pi_k}_{r_{\theta_k}}-Q^*_{r_{\theta_k}}||^2_\infty\Big] \\
&\overset{(t)}{\leq} \frac{C_1}{(1-(1+1/\epsilon)\gamma^2)K}\Big\{E[||Q^*_{r_{\theta_0}}-Q^{\pi_0}_{r_{\theta_0}}||^2_\infty]-(1+1/\epsilon)\gamma^2 E[||Q^*_{r_{\theta_{K-1}}}-Q^{\pi_{K-1}}_{r_{\theta_{K-1}}}||^2_\infty] \\
&+\sum_{k=1}^{K-1} 4(1+\epsilon)\hat{C}^2\alpha^2\Big[(1+2\lambda)TC+\frac{[2\log(1+e^{T\tilde{C}})+4\lambda\log(1+e^{2T\tilde{C}})]T^2C}{\beta}\Big]^2\Big\} \\
&= O(\frac{1}{K}),
\end{aligned} \tag{32}$$

where (s) follows from (30) and (t) follows from (31).

By combining (29) and (32), we have

$$\begin{aligned}
\frac{1}{K}\sum_{k=0}^{K-1} E[||\nabla L(\theta_k,\pi_k)||^2] &\leq \frac{2}{K}\sum_{k=0}^{K-1} E[||\nabla L(\theta_k,\pi_k)-\nabla L(\theta_k)||^2]+\frac{2}{K}\sum_{k=0}^{K-1} E[||\nabla L(\theta_k)||^2] \\
&= O(\frac{1}{\sqrt{K}})+O(\frac{1}{K}) = O(\frac{1}{\sqrt{K}}).
\end{aligned}$$

C PRACTICAL IMPLEMENTATIONS

Algorithm 2 BARS (practical implementation)

Input: Dataset $\mathcal{D}$, success reward r_{succ}, initial shaping parameter θ_0, and reference policy π_{ref}.
Output: Learned shaping parameter θ_K and policy π_K.

1: Initialize policy $\pi_0 = \pi_{\text{ref}}$.
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: **Data sample**: Sample a batch of B prompts $\{x^{(n)}\}_{n=1}^B$ from $\mathcal{D}$.
4: **Policy rollout**: For each prompt $x^{(n)}$, generate m trajectories $\{y^{(n,j)}\}_{j=1}^m$ using policy π_k.
5: **Policy update**: Perform a one-step PPO/GRPO/RLOO update under r_{θ_k} to obtain π_{k+1}.
6: **Reward alignment**: Compute the hypergradient in (33)-(34) and update shaping parameter via $\theta_{k+1} = \theta_k - \alpha[\hat{g}_{\text{ce}}(\theta_k) + \lambda\hat{g}_{\text{dpo}}(\theta_k)]$.
7: **end for**

Algorithm 1 is an idealized implementation of BARS, and requires two modifications for practical implementations. First, Algorithm 1 requires an idealized one-step soft policy iteration in the lower level to update the policy. In the practical implementation (Algorithm 2), we replace soft policy iteration with off-the-shelf RL algorithms, such as one-step PPO/GRPO/RLOO. Second, the computation of the hypergradient $[g_{\text{ce}}(\theta_k) + \lambda g_{\text{dpo}}(\theta_k)]$ in Algorithm 1 is expensive, and we use an efficient approximation $[\hat{g}_{\text{ce}}(\theta_k) + \lambda\hat{g}_{\text{dpo}}(\theta_k)]$ in practice. Below, we elaborate on the efficient approximation. We start with the CE hypergradient:

$$\begin{aligned}
g_{\text{ce}}(\theta_k) = \frac{1}{Bm}\sum_{n=1}^{B}\sum_{j=1}^{m}\Big[\nabla_\theta l_{\text{ce}}(\theta_k;(x^{(n)},y^{(n,j)}))+l_{\text{ce}}(\theta_k;(x^{(n)},y^{(n,j)}))\cdot \\
\frac{1}{\beta}\sum_{t=1}^{T}\Big(E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t=s_t^{(n,j)},A_t=a_t^{(n,j)}]
\end{aligned}$$

$$- E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t=s_t^{(n,j)}]\Big)\Big].$$

For simplicity of notation, we denote

$$Z_{t,k}(s,a)=\frac{1}{\beta}\sum_{t=1}E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t=s,A_t=a]$$
$$Z_{t,k}(s)=\frac{1}{\beta}E^{\pi_k}[\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_{\theta_k}(S_i,A_i)|S_t=s].$$

To precisely compute the above CE hypergradient, at each state $s_t^{(n,j)}$, we need to sample multiple trajectories, e.g., sample $\tilde{m}$ trajectories $\{s_t,a_t^j,s_{t+1}^j,\cdots,s_T^j,a_T^j,\cdots\}_{j=1}^{m}$. Then, we can estimate

$$Z_{t,k}(s_t,a_t^j)\approx\frac{1}{\beta}\Big[\nabla_\theta r_\theta(s_t,a_t^j)+\sum_{i=t+1}^{T}\gamma^{i-t}\nabla_\theta r_\theta(s_i^j,a_i^j)\Big]$$
$$Z_{t,k}(s_t)\approx\frac{1}{\beta}\Big[\frac{1}{\tilde{m}}\sum_{j=1}^{\tilde{m}}\nabla_\theta r_\theta(s_t,a_t^j)+\sum_{i=t+1}^{T}\gamma^{i-t}\nabla_\theta r_\theta(s_i^j,a_i^j)\Big].$$

However, it is intractable to estimate $Z_{t,k}(s_t,a_t^j)$ and $Z_{t,k}(s_t)$ at each time t in this way because it requires to sample $\tilde{m}$ trajectories at each time t. To address this issue, we use an approximation method proposed in Subsection 4.1.3 of (Shao et al., 2024). Specifically, we only sample m trajectories at the initial state $s_1^{(n)}=x^{(n)}$: $\{s_1^{(n,j)},a_1^{(n,j)},\ldots,s_T^{(n,j)},a_T^{(n,j)},\cdots\}_{j=1}^{m}$. We can approximate $Z_{t,k}(s_t)$ and $Z_{t,k}(s_t,a_t)$ via

$$Z_{t,k}(s_t^{(n,j)})\approx\frac{1}{\beta m}\sum_{j'=1}^{m}\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_\theta(s_i^{(n,j')},a_i^{(n,j')})$$
$$Z_{t,k}(s_t^{(n,j)},a_t^{(n,j)})\approx\frac{1}{\beta}\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_\theta(s_i^{(n,j)},a_i^{(n,j)}).$$

Therefore, we can approximate the CE hypergradient $g_{\text{ce}}(\theta)$ as

$$\hat{g}_{\text{ce}}(\theta)=\frac{1}{Bm}\sum_{n=1}^{B}\sum_{j=1}^{m}\Big[\nabla_\theta l_{\text{ce}}(\theta;(x^{(n)},y^{(n,j)}))+l_{\text{ce}}(\theta;(x^{(n)},y^{(n,j)}))\cdot$$
$$\sum_{t=1}^{T}\Big(\frac{1}{\beta}\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_\theta(s_i^{(n,j)},a_i^{(n,j)})-\frac{1}{\beta m}\sum_{j'=1}^{m}\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_\theta(s_i^{(n,j')},a_i^{(n,j')})\Big)\Big]. \tag{33}$$

The above CE gradient approximation (33) does not require new samples, and reuses the m samples for policy updates. With the same idea, we can approximate the DPO hypergradient $g_{\text{dpo}}(\theta)$ as

$$\hat{g}_{\text{dpo}}(\theta)=\frac{1}{B}\sum_{n=1}^{B}\frac{1}{\bar{m}^{(n)}(m-\bar{m}^{(n)})}\sum_{\bar{j}=1}^{\bar{m}^{(n)}(m-\bar{m}^{(n)})}\Big[\nabla_\theta l_{\text{dpo}}(\theta;(x^{(n)},y^{(n,\bar{j},+)},y^{(n,\bar{j},-)}))$$
$$+l_{\text{dpo}}(\theta;(x^{(n)},y^{(n,\bar{j},+)},y^{(n,\bar{j},-)}))\cdot\sum_{t=1}^{T}\Big(\frac{1}{\beta}\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_\theta(s_i^{(n,\bar{j},+)},a_i^{(n,\bar{j},+)})$$
$$+\frac{1}{\beta}\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_\theta(s_i^{(n,\bar{j},-)},a_i^{(n,\bar{j},-)})-\frac{2}{\beta m}\sum_{j=1}^{m}\sum_{i=t}^{T}\gamma^{i-t}\nabla_\theta r_\theta(s_i^{(n,j)},a_i^{(n,j)})\Big)\Big]. \tag{34}$$

The above DPO gradient approximation (34) does not require new samples either, and reuses the m samples for policy updates.

D EXPERIMENT DETAILS

Prompt for LLM judge

```
You are an expert evaluator for multi-turn search-augmented reasoning systems. Given a user prompt,
ground truth answer, and multi-turn generated response, evaluate each turn's effectiveness and compliance.

## EVALUATION TASK

Assess each turn's format compliance, content quality, and contribution toward the ground truth answer.

## SCORING CRITERIA

### FINAL TURN (Last Turn) - Score Range: [-1.0 to 1.0]

**Format Compliance:**
• Required: `<think>...</think><answer>...</answer>` (tags only, once each, in order)
• Answer in `<answer>` tag must not exceed 5 tokens

**Answer Correctness:**
• Correct and complete answer in `<answer>` tag that matches the ground truth

**Scoring Rules:**
• If format is incorrect: Final Turn Score = -1.0
• If format is correct, answer is incorrect: Final Turn Score = 0.2
• If format is correct, answer is correct: Final Turn Score = 1.0

### INTERMEDIATE TURNS - Score Range: [-1.0 to 1.0]

**Format Compliance:**
• Required: `<think>...</think><search>...</search><information>...</information>` (tags only, once each,
in order)
• Correct format: +0.1
• Incorrect format: -0.2

**Information Quality:**
• Relevant information in `<information>` tag that helps toward the ground
truth answer (e.g., ground truth exists in
the retrieved result within `<information>` tag): +0.3
• Irrelevant or unhelpful information in `<information>` tag: +0.0

**Search Efficiency Penalty:**
• Number of searches = Total count of `<search>` tags across all turns from Turn 1 up to and including
the current turn
• Search penalty = Number of searches × (-0.1)
• Encourages finding answers with fewer searches

**Intermediate Turn Score = Format Compliance + Information Quality + Search Penalty**

## OUTPUT FORMAT

Provide your evaluation using ONLY these XML tags:

<reasoning>
Systematically evaluate each turn: check format compliance, assess content quality, calculate scores with
clear explanations
</reasoning>

<score>
Turn1: X.X
Turn2: X.X
Turn3: X.X
...
</score>

REQUIREMENTS:
• Must provide exactly {len(turns)} scores (one per turn)
• Use decimal format (e.g., 0.5, -0.3, 1.0)
• Use only the specified XML tags, no additional text

## EVALUATION DATA

{prompt_text}
{turns_text}
{ground_truth_text}
**TURNS TO EVALUATE: {len(turns)}**

## Your Evaluation
```

We use the practical algorithm (Algorithm 2) for our experiments. The code was run on eight A100 GPUs, each with 80GB of memory. The choices of the hyperparameters are the same across benchmarks and models. The number of rollouts for each prompt is $m = 8$, the KL coefficient is $\beta = 0.005$, the discount factor $\gamma = 0.999$, the weighting coefficient is 0.1, the number of training iterations is $K = 450$, and the batch size of each iteration is $B = 512$. Both the reward model and policy model use AdamW as the optimizer with a learning rate of 10^{-6}. The maximum prompt length

is 4096. For each turn, the maximum response length is 500. When the RL optimizer is PPO, the critic uses AdamW as the optimizer with a learning rate of 10^{-5}. The reward model uses the same base model as the policy model.

The prompt for the turn-level LLM judge is provided in (Wei et al., 2025). For the sake of completeness, we also provide it here.

D.1 Training Dynamics

In this section, we include the training and test curves on Qwen2.5-3B and the NQ-HotpotQA benchmark.

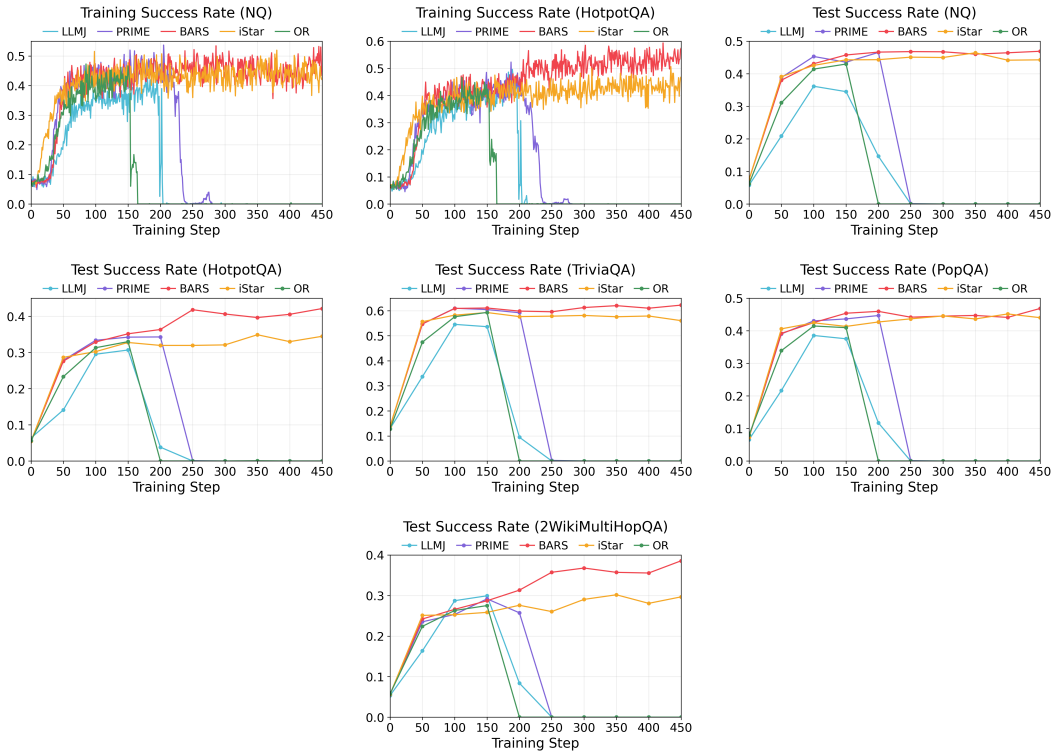

Figure 3: Training and test curves on Qwen2.5-3B and the NQ-HotpotQA benchmark.

Figure 3 shows the training and test success rates across multiple benchmarks. The training set consists of NQ and HotpotQA, where HotpotQA includes multi-hop questions, while the test set covers a broader collection of datasets including NQ, HotpotQA, TriviaQA, PopQA, and 2WikiMultiHopQA. BARS exhibits consistently stable training dynamics, with smooth and monotonic improvements that quickly plateau at a high performance level. In contrast, alternative methods such as OR, LLMJ, and PRIME demonstrate significant instability, including abrupt performance drops and even training collapse. This instability is also reflected in test performance, where these methods often degrade sharply. Although iStar achieves stable training, it converges to a worse performance compared to BARS. Overall, the results demonstrate that BARS not only achieves superior final accuracy but also provides substantially improved training stability and strong generalization to both mixed single-hop (NQ, TriviaQA, and PopQA) and multi-hop evaluation benchmarks (HotpotQA and 2WikiMultiHopQA).

D.2 Training cost comparison

We include a quantitative comparison of the wall-clock training cost and peak GPU memory usage for OR, LLMJ, PRIME, iStar, and BARS under the same optimizer (AdamW), rollout budget ($m = 8$), batch size (512), and hardware configuration (8×A100 80G). Since all methods are trained for the

same number of iterations, we report the average wall-clock time per training iteration as a fair measure of training cost. The following results are obtained using Qwen2.5-3B with GRPO on the NQ-Search benchmark.

Table 5: Training cost comparison

	Time/iteration (s)	Peak GPU memory (GB)
OR	379	73
LLMJ	542	76
PRIME	417	77
iStar	429	75
BARS	441	76

The results in Table 5 demonstrate that OR is the fastest because it directly optimizes the policy without learning an auxiliary shaping reward. Compared with PRIME and iStar, BARS introduces only a modest computational overhead (5.8% and 2.8% additional time per iteration, respectively). BARS requires approximately 16% more time than OR. The modest overhead arises because the dominant cost of each iteration is policy rollout, which is identical across all the methods under the same rollout budget. The additional reward optimization in BARS reuses the same sampled trajectories as the policy update and therefore does not require additional rollouts. LLMJ incurs the highest training cost because it uses GPT-OSS-120B to generate turn-level evaluation and reward labels during training, introducing substantial additional inference overhead. The peak GPU memory usage is comparable across all methods because they share the same policy architecture and rollout pipeline. In addition, our implementation offloads model parameters to the CPU when they are not actively being optimized, preventing the additional reward optimization from significantly increasing GPU memory consumption.

D.3 WebShop experiment

In this section, we evaluate BARS on WebShop (Yao et al., 2022), an interactive e-commerce benchmark. Given a natural language shopping instruction, the agent interacts with the website through a sequence of actions, including searching for products, navigating between pages, inspecting product details, and selecting the final item to purchase. The agent can interact with the environment for up to 10 turns, which is a long horizon compared to the search benchmarks (2 and 4 turns). WebShop reports two primary evaluation metrics, success and score. Success measures whether the agent purchases the exact target product that meets all user requirements. Score provides a finer-grained evaluation by assigning partial credit based on how well the selected product matches the desired attributes (e.g., category, brand, color, price, and other specifications). While success is a binary value that captures strict task completion, score is a continuous value between 0 and 1 that reflects the overall quality of the agent's purchasing decisions, even when the exact target is not found. We use both success and score as the outcome reward for RL training and evaluate both success rate and score. We use Qwen3-4B with GRPO and compare against the three baselines in the paper (OR, PRIME, and iStar). Note that we do not compare to LLMJ as LLMJ (Wei et al., 2025) only provides LLM judge prompts for the search benchmarks.

The number of rollouts for each prompt is $m = 8$, the KL coefficient is $\beta = 0.01$, the discount factor $\gamma = 0.999$, the weighting coefficient is 0.1, the number of training iterations is $K = 200$, and the batch size of each iteration is $B = 64$. The number of turns is 10. Both the reward model and policy model use AdamW as the optimizer with a learning rate of 10^{-6}. The maximum prompt length is 4096. For each turn, the maximum response length is 512.

The results in Table 6 demonstrate that BARS consistently outperforms all baselines under both outcome reward formulations, including binary reward (Success) and non-binary reward (Score). When using success as the outcome reward, BARS achieves a success rate of 83.2% and a score of 92.7%, improving over the strongest baselines by 2.7% and 3.3%, respectively. When using the WebShop score as the outcome reward, BARS achieves a success rate of 82.5% and a score of 93.8%, improving over the strongest baselines by 3.0% and 3.6%, respectively. We also observe a consistent effect of the outcome reward formulation: optimizing the binary success reward generally leads to

higher success rates, whereas optimizing the continuous WebShop score produces higher scores. Importantly, BARS maintains its advantage in both settings, suggesting that its effectiveness is not specific to a particular form of outcome feedback.

Table 6: Performance (percentage) on WebShop under different outcome reward formulations.

	Success as Reward		Score as Reward	
Method	Success Rate	Score	Success Rate	Score
OR	75.5 ± 0.6	84.4 ± 0.7	73.3 ± 0.2	85.2 ± 0.3
PRIME	80.5 ± 0.4	88.1 ± 0.5	78.4 ± 0.3	89.7 ± 0.4
iStar	79.6 ± 0.6	89.4 ± 0.3	79.5 ± 0.4	90.2 ± 0.3
BARS	$\mathbf{83.2 \pm 0.5}$	$\mathbf{92.7 \pm 0.4}$	$\mathbf{82.5 \pm 0.4}$	$\mathbf{93.8 \pm 0.3}$

E LIMITATIONS

BARS depends on the quality of the outcome reward signal used in the upper-level objective. In our experiments, we focus on binary success rewards, which are reliable since the ground truth answers are given. In more complex real-world settings, outcome rewards may be noisy, delayed, or partially specified, which could affect the quality of the learned shaping reward.

F POTENTIAL SOCIETAL IMPACT

This work contributes to improving reinforcement learning for agentic AI systems by addressing sparse supervision and unstable optimization in multi-turn decision-making. By enabling more stable and efficient learning from outcome-based signals, BARS can improve the reliability of AI agents in applications such as information retrieval, decision support, and tool-augmented reasoning. More broadly, our framework aligns with efforts in human-centered AI by improving credit assignment while preserving the original task objective, which may help reduce unintended behaviors introduced by heuristic or biased reward design. However, the ability to learn dense shaping rewards also introduces potential risks. If the outcome reward is misaligned with human values or imperfectly specified, the learned shaping reward may amplify these biases and lead to undesirable behaviors.

G USAGE OF LLMS

In this paper, we use LLMs to polish the writing.