Language Models as Implicit Tree Search

Ziliang Chen [1] **Zhao-Rong Lai** [2] **Yufeng Yang** [3] **Liangda Fang** [2] **Zhanfu Yang** [4] **Liang Lin** [1 3]

Abstract

Despite advancing language model (LM) alignment, direct preference optimization (DPO) falls short in LM reasoning with the free lunch from reinforcement learning (RL). As the breakthrough, this work proposes a new RL-free preference optimization method aiming to achieve DPO along with learning another LM, whose response generation policy holds the asymptotic equivalence with AlphaZero-like search, the apex of algorithms for complex reasoning missions like chess Go. While circumventing explicit value and reward modeling, the neural implicit tree search executed by the extra LM remains seeking to equip DPO with reasoning procedure technically akin to AlphaZero. Our experiments demonstrate that our methodology outperforms both regular DPO variants in human preference alignment, and MCTS-based LMs in mathematical reasoning and planning tasks.

1. Introduction

Preference optimization paradigms, notably Reinforcement Learning from Human Feedback (RLHF) (Bai et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2024b), are foundational to modern Language Model (LM) alignment, aiming to imbue these models with human's behavioral characteristics, inclination, and value. In particular, DPO circumvent reward modeling so as to directly optimize LMs with preferential response pairs, thereby reinforcement learning (RL)'s complexities and instability avoided to yield more efficiently and stably aligned models.

Distinct from preference alignment, sophisticated reasoning capabilities underlying human brains are not typically replicated by RLHF and DPO alone: the emulation often necessitates synergistic integration with other techniques, *i.e.*, meticulous prompt engineering (Wei et al., 2022), process-based reward modeling (Lightman et al., 2023), and Monte Carlo Tree Search (MCTS) (Xie et al., 2024a). Among these complementary approaches, MCTS stands out as essentially promising, due to its exploration of complex decision spaces in achieving high-level reasoning tasks such as game Go (Silver et al., 2017b). Massive leading research were proposed to combine MCTS with preference optimization, whereas their MCTS procedures inevitably require the value function learned by RL for executing the search principles of Upper Confidence Bound (UCB). It is noteworthy that the critical advantage of DPO is to skip the value learning to achieve faster and more stable preference optimization than RLHF. The standard MCTS's reliance on an RL-learned value function directly conflicts with DPO's core strength of bypassing RL entirely, showing a significant hurdle for their seamless integration. It implies the necessity to reconcile MCTS methodologies with the RL-free nature behind DPO.

Rather than the regular MCTS, this paper started from its theoretical variant derived from (Grill et al., 2020b), where AlphaZero-like MCTS can be treated as a stochastic policy solved by the state-specific local optimization regularized by the reverse KL-divergence. With this regard, each state-specific local tree search decision asymtotically converges to AlphaZero-like MCTS decision with the gap bounded by the empirical visit counts, thus, lifting the expressiveness of MCTS limited by the integer-count probability and sparse tree width. Regretfully, the local solution of this stochastic policy involves the parameter only solvable by dichotomic search per state. In terms of the exponentially complex state space in token-level language generation, this *implicit tree search* (ITS) is hardly applied in MCTS-based LM research.

Contributions. We propose a new RL-free preference optimization paradigm aiming to approximate ITS by LMs. Specifically,

- Beyond the original token-selection policy in DPO, we propose another LM to learn the global policy of ITS with regards to neural universal approximation. The preference optimization built on top of DPO, without dichotomic search and any other RL elements, inherits the critical advantages of the DPO family.

[1]Research Institute of Multiple Agents and Embodied Intelligence, Peng Cheng Laboratory; [2]Jinan University, Guangzhou, China; [3]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China; [4]Department of Computer Science, Rutgers University. Correspondence to: Liang Lin <linlng@mail.sysu.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

- In terms of the gradient analysis to the preference optimization with ITS, learning its global policy yields the response generation with diverse and better aligned preference, therefore, simultaneously benefits AI alignment and reasoning tasks.
- Self-improved preference policy augmentation and decoding strategy are proposed for our ITS-based preference optimization approaches, in order to align with the MCTS-based LLM decoding. The connection between the decoding strategies and group-relative policy optimization (Shao et al., 2024) has also been presented.

Our experiments included the evaluation across human preference alignment, mathematical reasoning, and mathematical planning, where our approach concurrently reaped the optima against DPO variants and MCTS-derived baselines.

2. Preliminaries

For language generation, language models (LM) θ serve as a token-level Markov decision process (MDP) $(\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\mathcal{T})$, where the state $\boldsymbol{s}_t = (\boldsymbol{x}, \boldsymbol{y}_{<t})$ in $\mathcal{S}$ consists of a prompt $\boldsymbol{x}$ followed by a sequence of response tokens $\boldsymbol{x}_{<t}$ generated at the previous t-1 steps, and the action $\boldsymbol{a}_t = y_t$ in $\mathcal{A}$ denotes the token selected at the current t step. Given this, the transition kernel $\mathcal{P}$: $(\mathcal{S},\mathcal{A})\rightarrow \mathcal{S}$ holds the deterministic mapping from $(\boldsymbol{s}_t, \boldsymbol{a}_t)$ to the next state $\boldsymbol{s}_{t+1} = (\boldsymbol{x}, \boldsymbol{y}_{<t+1})$ with $\boldsymbol{y}_{<t+1} = (\boldsymbol{y}_{<t}, y_t)$. The reward $\mathcal{R}$: $(\mathcal{S},\mathcal{A}) \rightarrow \mathbb{R}$ quantifies the preference of selecting tokens, and $T\in\mathcal{T}$ denotes the step to cease the generation when $\boldsymbol{a}_T$ is the end-of-sequence (**EoS**) token. The goal is to learn the sequential token-selection policy π_θ that maximizes the accumulated reward $R(\boldsymbol{x}, \boldsymbol{y})$=$\sum_{t=1}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t)$, $\forall r \in \mathcal{R}$.

RLHF and DPO. Token-level MDP needs $R(\boldsymbol{x}, \boldsymbol{y})$ to reflects human preferences. RLHF (Bai et al., 2022) captures this from prompt $\boldsymbol{x}$ with its response pairs $\boldsymbol{y}^{(\mathrm{w})} \succ \boldsymbol{y}^{(\mathrm{l})}$, where $\succ$ identifies $\boldsymbol{y}^{(\mathrm{w})}$ more preferable than $\boldsymbol{y}^{(\mathrm{l})}$. The comparison is made by Bradley-Terry (BT) model (Bradley & Terry, 1952) in the contextual bandit setting

$$P_{\mathrm{BT}}(\boldsymbol{y}^{(w)} \succ \boldsymbol{y}^{(l)}|\boldsymbol{x}) = \frac{\exp(R(\boldsymbol{x}, \boldsymbol{y}^{(w)}))}{\exp(R(\boldsymbol{x}, \boldsymbol{y}^{(w)})) + \exp(R(\boldsymbol{x}, \boldsymbol{y}^{(l)}))}. \tag{1}$$

where $R(\boldsymbol{x}, \boldsymbol{y})$ can be learned through

$$\min_R -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}^{(w)},\boldsymbol{y}^{(l)})\sim D}[\log \sigma(R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) - R(\boldsymbol{x}, \boldsymbol{y}^{(l)}))] \tag{2}$$

with the human preference dataset D. It serves RLHF as a KL-constraint RL objective

$$\max_{\pi_\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, y\sim\pi_\theta(\cdot|\boldsymbol{x})}[R(\boldsymbol{x}, \boldsymbol{y}) - \beta D_{KL}(\pi_\theta(\cdot|\boldsymbol{x})\|\pi_{\mathrm{ref}}(\cdot|\boldsymbol{x}))], \tag{3}$$

where the reference policy π_{ref} is initialized by supervised fine-tuning (SFT) LM to learn the generation policy π_θ via PPO (Schulman et al., 2017), and $\beta > 0$ controls their trade off. DPO (Rafailov et al., 2024b) reconsiders the optimization from a interpretation of $R(\boldsymbol{x}, \boldsymbol{y})$ represented as $R(\boldsymbol{x}, \boldsymbol{y}) = \beta \log \frac{\pi_\theta(\boldsymbol{y}|\boldsymbol{x})}{\pi_{\mathrm{ref}}(\boldsymbol{y}|\boldsymbol{x})} + \beta \log Z(x)$. With such regards, Eq.2 is equivalent with

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}^{(w)},\boldsymbol{y}^{(l)})\sim\mathcal{D}} \left[\log \sigma\Big(\beta \log \frac{\pi_\theta(\boldsymbol{y}_w|\boldsymbol{x})}{\pi_{\mathrm{ref}}(\boldsymbol{y}_{\boldsymbol{w}}|\boldsymbol{x})} - \beta \log \frac{\pi_\theta(\boldsymbol{y}_l|\boldsymbol{x})}{\pi_{\mathrm{ref}}(\boldsymbol{y}_l|\boldsymbol{x})}\Big)\right], \tag{4}$$

therefore the reward modeling phase is skipped to directly optimize the token-selection policy π_θ.

Token-level Interpretation of LM Alignments. The aforementioned LM alignments focus on a complete response $\boldsymbol{y}$, yet π_θ is executed in each MDP step using token-level rewards $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ with $\beta \log \frac{\pi_\theta(y_t|[\boldsymbol{x},\boldsymbol{y}_{<t}])}{\pi_{\mathrm{ref}}(y_t|[\boldsymbol{x},,\boldsymbol{y}_{<t}])}$=$\beta \log \frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\mathrm{ref}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}$. (Rafailov et al., 2024a) re-formulates Eq.3 via the token-level interpretation

$$\begin{aligned}&\max_{\pi_\theta} \mathbb{E}_{\boldsymbol{s}_0=\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{a}_t\sim\pi_\theta(\cdot|\boldsymbol{s}_t)} \sum_{t=1}^{T_{\boldsymbol{y}}} \Big[r(\boldsymbol{s}_t, \boldsymbol{a}_t) - \beta \log \frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\mathrm{ref}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}\Big] \\ \leftrightarrows &\max_{\pi_\theta} \mathbb{E}_{\boldsymbol{s}_0=\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{a}_t\sim\pi_\theta(\cdot|\boldsymbol{s}_t)} \sum_{t=1}^{T_{\boldsymbol{y}}} \Big[r(\boldsymbol{s}_t, \boldsymbol{a}_t) + \beta \log \pi_{\mathrm{ref}}(\boldsymbol{a}_t|\boldsymbol{s}_t) \\ &\qquad + \beta\mathcal{H}(\pi)\Big].\end{aligned} \tag{5}$$

So given $r_{\mathrm{ME}}(\boldsymbol{s}_t, \boldsymbol{a}_t)$=$r(\boldsymbol{s}_t, \boldsymbol{a}_t)$+$\beta \log \pi_{\mathrm{ref}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$, RLHF with the reframed reward is maximum-entropy RL (Ziebart, 2010) whose fixed point solution leads to the optimal policy π^*_θ that satisfies $\pi^*_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)$=$\exp\big(\frac{Q^*(\boldsymbol{s}_t,\boldsymbol{a}_t)-V^*(\boldsymbol{s}_t)}{\beta}\big)$ with respect to $V^*(\boldsymbol{s}_t) = \beta \log \sum_{a\in\mathcal{A}} \exp\big(Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t)/\beta\big)$. It is noteworthy that the optimal state-action value function holds the initial value $Q^*(\boldsymbol{s}_0, \boldsymbol{a}_0) = R(\boldsymbol{x}, \boldsymbol{y}) = V^*(\boldsymbol{s}_0) + \beta \sum_{t=1}^{T_{\boldsymbol{y}}} \log \frac{\pi^*_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\mathrm{ref}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}$ with $V^*(\boldsymbol{s}_0) = V^*(\boldsymbol{x})$, henceforth we have the objective similarly derived from DPO

$$\mathcal{L}_{\text{T-DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}^{(\mathrm{w})},\boldsymbol{y}^{(\mathrm{l})})\sim\mathcal{D}} \left[\log \sigma\Big(\beta \sum_{t=1}^{T_w} \log \frac{\pi_\theta(\boldsymbol{a}^w_t|\boldsymbol{s}^w_t)}{\pi_{\mathrm{ref}}(\boldsymbol{a}^w_t|\boldsymbol{s}^w_t)} - \beta \sum_{t=1}^{T_l} \log \frac{\pi_\theta(\boldsymbol{a}^l_t|\boldsymbol{s}^l_t)}{\pi_{\mathrm{ref}}(\boldsymbol{a}^l_t|\boldsymbol{s}^l_t)}\Big)\right], \tag{6}$$

where $\boldsymbol{a}^w_t$, $\boldsymbol{a}^l_t$ indicate the t-th token of the responses $\boldsymbol{y}^w$, $\boldsymbol{y}^l$; and $\boldsymbol{s}^w_t = [\boldsymbol{x}, \boldsymbol{y}^w_{<t}]$, $\boldsymbol{s}^l_t = [\boldsymbol{x}, \boldsymbol{y}^l_{<t}]$, respectively.

MCTS and AlphaZero-like Tree Search. Given the state-action value treated as feedback from a contextual bandit, MCTS (Browne et al., 2012) builds an online search tree incrementally for decision making. (More details refers to (Silver et al., 2017b)).MCTS algorithms are famous in massive researches, among which the most successful case is Alpha Zero (Silver et al., 2017b) and its variants (Silver et al., 2017b; Schrittwieser et al., 2020). The MCTS variant underlying these methods remarked as *AlphaZero-like tree search*, compute a policy at the root of the search tree improved from the distribution predicted by the prior policy

π, which is updated by distilling back the tree-search policy to update the prior. The significant difference from other MCTS is AlphaZero-like tree incorporating π as the prior during the search procedure. It leads to the heuristic search principle inspired from UCB:

$$\boldsymbol{a}^\star \triangleq \arg\max_{a\in\mathcal{A}_{\mathsf{MCTS}}}\left[Q^\pi(\boldsymbol{s},\boldsymbol{a})+c\pi(\boldsymbol{a}|\boldsymbol{s})\frac{\sqrt{\sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}} n(\boldsymbol{s},\boldsymbol{a}')}}{1+n(\boldsymbol{s},\boldsymbol{a})}\right]. \tag{7}$$

where $c>0$ is a balance factor between exploration and exploitation in the MCTS search space $\mathcal{A}_{\mathsf{MCTS}}$. The four-stage procedure of Eq.7 is shown in Appendix.A.

3. Implicit Tree Search: Warm-Up

AlphaZero-like MCTS variants have been intensively explored in leading researches for LM training (Feng et al., 2023; Zhang et al., 2024a) and alignment (Xie et al., 2024b; Chen et al., 2024a), in order to mitigate their downsides of reasoning and long-form generation ability captured in post-training. In spite of different algorithm designs, the methodologies are consistent with MCTS in the data-generation manner: $\hat{\pi}$ merges with the decoding strategy to generate responses for training data augmentation, then the token-selection policy π_θ and visit counts updated along with these data, further renew the decoding by Eq.7. Despite significantly advancing LM alignments, the decision-making pipeline relies on the online state-action visit count $n(\boldsymbol{s},\boldsymbol{a})$ per state-action pair. In terms of the integer-count probability and the sparse tree width in the initial training iters, it inevitably suffers from two critical problems.

Cold-start exploration expressiveness. $\frac{\sqrt{\sum_{a'\in\mathcal{A}} n(\boldsymbol{s},\boldsymbol{a}')}}{1+n(\boldsymbol{s},\boldsymbol{a})}$ is the key to differentiate $\hat{\pi}$ and π, so AlphaZero-like tree search typically employs its *empirical visit distribution* $\frac{1+n(\boldsymbol{s},\boldsymbol{a})}{|\mathcal{A}_{\mathsf{MCTS}}|+\sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}} n(\boldsymbol{s},\boldsymbol{a}')}$ or its exponential generalization $\frac{1+\left(n(\boldsymbol{s},\boldsymbol{a}')\right)^r}{|\mathcal{A}_{\mathsf{MCTS}}|+\sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}}\left(n(\boldsymbol{s},\boldsymbol{a}')\right)^r}$ to explore the tree structure. Whereas the discrete nature of visit counts limit the expressiveness of exploration strategy.

Proposition 3.1. *Given a fixed $r\in\mathbb{R}/\{0\}$, the action selection probability by the exponential empirical visit distribution holds its value with the equal cardinality of $\mathbb{Z}$.*

Proposition 3.2. *Given a learnable $r\in\mathbb{R}$, the action selection probability by the exponential empirical visit distribution holds the value in the range $(\frac{1}{|\mathcal{A}_{\mathsf{MCTS}}|},1)$.*

The problem is cold-start because early decisions heavily influences the tree growth yet the ratios of integers are more unstable when the visit counts are small. It may drive the language generation into bias or sub-optima.

Sparse-action improvement. $\hat{\pi}$ solely improved for actions searched at least once by the previous simulation whereas those with $n(\boldsymbol{s},\boldsymbol{a})$=0 would be dominated by θ. In terms of the large action space in language and the deterministic policy in Eq.7, it may cause a large simulation budget to improve actions never visited.

Recognizing that these limitations are intrinsic to the use of online visit counts in conventional MCTS algorithms, we supersedes the online tree-search strategy with stochastic policy optimization established in (Grill et al., 2020b) (depicted in Figure.1.b), named *Implicit Tree Search* since the optimized policy is provably equivalent with the AlphaZero-like tree search exploration defined in Eq.7.

3.1. MCTS as Regularized Policy Optimization

More specifically, for any state, the action selection formula in Eq.7 holds the identical interpretation

$$\boldsymbol{a}^\star(\boldsymbol{s}) \triangleq \arg\max_{a\in\mathcal{A}_{\mathsf{MCTS}}}\left[Q^\pi(\boldsymbol{s},\boldsymbol{a})+\lambda_N(\boldsymbol{s})\cdot\frac{\pi(\boldsymbol{a}|\boldsymbol{s})}{\hat{\pi}(\boldsymbol{a}|\boldsymbol{s})}\right] \tag{8}$$

where $\hat{\pi}(\boldsymbol{a}|\boldsymbol{s}) \triangleq \frac{1+n(\boldsymbol{s},\boldsymbol{a})}{|\mathcal{A}_{\mathsf{MCTS}}|+\sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}} n(\boldsymbol{s},\boldsymbol{a}')}$ represents the empirical visit distribution and $\lambda_N(\boldsymbol{s}) \triangleq c\cdot\frac{\sqrt{\sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}} n(\boldsymbol{s},\boldsymbol{a}')}}{|\mathcal{A}_{\mathsf{MCTS}}|+\sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}} n(\boldsymbol{s},\boldsymbol{a}')}$ denotes the *state-specific multiplier*. Notice that for any $\boldsymbol{s}\in\mathcal{S}$, the empirical visit distribution $\hat{\pi}(\cdot|\boldsymbol{s})$ is the only way that search algorithm influences the optimal action of tree search. In language generation context, we set π=π_θ, so for any $\boldsymbol{s}\in\mathcal{S}$, $\hat{\pi}(\cdot|\boldsymbol{s})$ holds a corresponding predominant policy $\overline{\pi}(\cdot|\boldsymbol{s})$ as

Theorem 3.3. ***(Asymptotic equivalence between ITS and MCTS policies)*** *$\forall\boldsymbol{s}\in\mathcal{S}$, let $\overline{\pi}(\cdot|\boldsymbol{s})$ be the solution of*

$$\overline{\pi}(\cdot|\boldsymbol{s})\triangleq\arg\max_{\mathbf{y}(\boldsymbol{s})\in\mathbb{S}}\Big[\mathbf{Q}^{\pi_\theta}(\boldsymbol{s})^\top\mathbf{y}(\boldsymbol{s}) - \lambda_N(\boldsymbol{s})D_{KL}[\pi_\theta(\cdot|\boldsymbol{s}),\mathbf{y}(\boldsymbol{s})\Big] \tag{9}$$

where $\mathbb{S}$ denotes the $|\mathcal{A}_{\mathsf{MCTS}}|$-dimensional simplex, and $\mathbf{Q}^{\pi_\theta}(\boldsymbol{s})=\big(Q^{\pi_\theta}(\boldsymbol{s},\boldsymbol{a}_1),Q^{\pi_\theta}(\boldsymbol{s},\boldsymbol{a}_2),\cdots,Q^{\pi_\theta}(\boldsymbol{s},\boldsymbol{a}_{|\mathcal{A}_{\mathsf{MCTS}}|})\big)$ is the Q-value vector with respect to the state $\boldsymbol{s}$. Then as the visit counts increase, the empirical visit distribution $\hat{\pi}(\cdot|\boldsymbol{s})$ in Eq.8 converges to $\overline{\pi}(\cdot|\boldsymbol{s})$ with the upper bound

$$\Big|\Big|\hat{\pi}(\cdot|\boldsymbol{s})-\overline{\pi}(\cdot|\boldsymbol{s})\Big|\Big| \le \frac{|\mathcal{A}_{\mathsf{MCTS}}|+1}{|\mathcal{A}_{\mathsf{MCTS}}|+N} \tag{10}$$

where N indicates the total rounds of simulation.

Theorem.3.3 is concluded from the definition.1 and proposition.1,5 in (Grill et al., 2020b). The theoretical result verifies that, given sufficient simulation steps, there exists an asymptotic equivalence between the exploration of AlphaZero-like tree search $\hat{\pi}(\cdot|\boldsymbol{s})$ and the stochastic-sampling policy $\overline{\pi}(\cdot|\boldsymbol{s})$. Note that the policy optimization in a simplex allows the

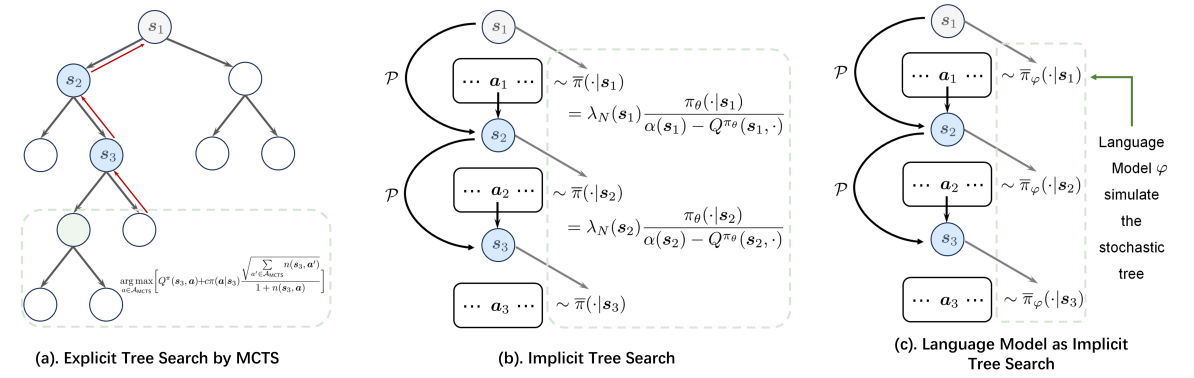


Figure 1. Comparison between the diagrams of (a).MCTS (regular MCTS and AlphaZero-like tree search), (b). Implicit Tree Search (ITS) (Grill et al., 2020b), and our IT-PO algorithm.

continuous value change in $\overline{\pi}(\cdot|\boldsymbol{s})$ that prevents the risk of cold-start expressiveness. Besides, the reversed KL divergence is smooth on $\overline{\pi}(\cdot|\boldsymbol{s})$, ensuring the sparse-action improvement resolved.

Accordingly, the ***expand*** and ***backup*** stages along with the MCTS procedure require the state-specific stochastic policy optimization formulated as follows

Lemma 3.4. ***(Solution of ITS policy** (Grill et al., 2020b))* *$\forall \boldsymbol{s} \in \mathcal{S}$, the solution $\overline{\pi}(\cdot|\boldsymbol{s})$ of Eq.9 holds*

$$\forall \boldsymbol{a}_t \in \mathcal{A}_{\text{MCTS}},\ \ \overline{\pi}(\boldsymbol{a}_t|\boldsymbol{s}_t) = \lambda_N(\boldsymbol{s}_t)\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\alpha(\boldsymbol{s}_t) - Q^{\pi_\theta}(\boldsymbol{s}_t, \boldsymbol{a}_t)} \tag{11}$$

where $\alpha(\boldsymbol{s}_t)$ is defined as

$$\begin{aligned}
&1).\alpha(\boldsymbol{s}_t) \triangleq \max\left\{\alpha \in \mathbb{R},\ \text{s.t.} \sum_{\boldsymbol{a}\in\mathcal{A}_{\text{MCTS}}} \pi(\boldsymbol{a}|\boldsymbol{s}_t) = 1\right\} \\
&2).\alpha(\boldsymbol{s}_t) \geq \alpha(\boldsymbol{s}_t)_{\min} \triangleq \max_{\boldsymbol{a}\in\mathcal{A}_{\text{MCTS}}} \Big(Q(\boldsymbol{s}_t, \boldsymbol{a}) + \lambda_N \pi_\theta(\boldsymbol{a}|\boldsymbol{s}_t)\Big) \\
&3).\alpha(\boldsymbol{s}_t) \leq \alpha(\boldsymbol{s}_t)_{\max} \triangleq \max_{\boldsymbol{a}\in\mathcal{A}_{\text{MCTS}}} Q(\boldsymbol{s}_t, \boldsymbol{a}) + \lambda_N
\end{aligned} \tag{12}$$

Lemma.3.4 is derived from Appendix.B.3 in the paper.

3.2. Challenges of Combing Implicit Tree and LMs

Observed that $\sum_{\boldsymbol{a}\in\mathcal{A}_{\text{MCTS}}}\pi(\boldsymbol{a}|\boldsymbol{s}_t)$ monotonically decreases on $(\alpha(\boldsymbol{s}_t)_{\min}, \alpha(\boldsymbol{s}_t)_{\max})$, the theoretical result guarantees the state-specific hyper-parameter $\alpha(\boldsymbol{s}_t)$ uniquely identified using dichotomic search over $(\alpha(\boldsymbol{s}_t)_{\min}, \alpha(\boldsymbol{s}_t)_{\max})$. In other words, $\overline{\pi}$ **can not be universally optimized by gradient descent and even for each state, $\overline{\pi}(\cdot|\boldsymbol{s}_t)$'s solution in Eq.11 is not exactly closed-form**. To this end, $\overline{\pi}$ can not be flexibly used as explicit search like regular MCTS algorithms and instead, only applicable for policy distillation to update the Q function in AlphaZero-like MCTS. However, it is hardly applied for LLM-based preference optimization.

The vital problem roots in the exponentially increase size of the valid state space. As demonstrated in token-level MDP, texts are generated by sequential token selection with the state $\boldsymbol{s}_t = (\boldsymbol{x}, \boldsymbol{y}_{<t})$ deterministically transmitted as $\mathcal{P}(\boldsymbol{s}_t, \boldsymbol{a}_t) \rightarrow \boldsymbol{s}_{t+1} = (\boldsymbol{x}, \boldsymbol{y}_{<t+1})$. It implies that for each prompt $\boldsymbol{x}$, the valid state space holds the size $|V|^t$ ($|V|$ indicates the size of token vocabulary V) exponentially increase with t. Given such challenge, Lemma.3.4 solely promises the local policy so that the actor-critic modeling is required to obtain π_θ and Q^{π_θ} simultaneously, as leading LLM-based MCTS approaches do. What's worse, the local policy also relies on $\alpha(\boldsymbol{s}_t)$ inferred by dichotomic search per state, implying the computation and buffer also exponentially increasing. This problem even more severe when the action space $\mathcal{A}_{\text{MCTS}}$ become sentence-level.

4. Preference Optimization with Implicit Tree Search

In the previous section, we demonstrated the advantages of $\overline{\pi}$ over the empirical visit exploration employed by regular MCTS methods, while its solution of Lemma.3.4 is hardly implemented in the LM realm due to the state-specific non-differentiable hyper-parameters $\alpha(\boldsymbol{s}_t)$. In this section, we derive the new RL-free preference optimization paradigm to approximate $\overline{\pi}$ without either Q function or $\alpha(\boldsymbol{s}_t)$, yielding the response generation as MCTS without value modeling.

4.1. Language Models as ITS Policy

Specifically, we chase for the universal approximator of $\overline{\pi}$ based on the extra policy network $\overline{\pi}_\varphi$ parameterized by the other language model φ (Figure.1.c). It supports the instant inference of $\overline{\pi}(\cdot|\boldsymbol{s}_t)$ (*i.e.*, no dichotomic search required to obtain $\alpha(\boldsymbol{s}_t)$), no matter $\boldsymbol{s}_t$ was visited or not.

Thought consistent with π_θ in the reversed KL constraint, $\overline{\pi}$ is essentially a stochastic strategy generalized from deterministic policy in the AlphaZero-like tree search (Theo-

rem.3.3). It motivates us to initialize its universal approximator $\overline{\pi}_\varphi$ by supervised fine-tuning. Specifically, we fine-tune a pre-trained LM to obtain π_{sft}, whose parameters initialize the reference policy π_{ref}, the token selection policy π_θ, and our ITS policy approximator $\overline{\pi}_\varphi$. Then we applied the DPO variant algorithm, *i.e.*, conservative DPO (cDPO) (Mitchell, 2023b) to update π_θ:

$$\mathcal{L}_{\mathsf{cDPO}}(\theta) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}^{(w)},\boldsymbol{y}^{(l)})\sim D} \\ -[(1-\epsilon)\log[\sigma(u_\theta(\boldsymbol{x},\boldsymbol{y}^{(w)},\boldsymbol{y}^{(l)}))]+\epsilon\log[\sigma(u_\theta(\boldsymbol{x},\boldsymbol{y}^{(l)},\boldsymbol{y}^{(w)}))] \tag{13}$$

where $u(\cdot)$ indicates the preference logit derived from BT model, $\epsilon \in [0, 0.2)$ denotes the margin then when $\epsilon = 0$, cDPO degenerates into DPO.

After training with $\min_\varphi \mathcal{L}_{\mathsf{cDPO}}(\varphi;\theta)$, the parameters of π_θ, π_{ref} are frozen to update the parameter of $\overline{\pi}_\varphi$. This manner provides the stable π_θ to derive the optimization of φ, *i.e.*, Implicit-Tree Preference Optimization (IT-PO).

4.2. Implicit Tree Preference Optimization (IT-PO)

IT-PO seeks for learning $\overline{\pi}_\varphi$ to substitute $\overline{\pi}$ without either reward or value functions. It is noteworthy that $\overline{\pi}(\cdot|\boldsymbol{s}_t)$ inherently contains $Q^{\pi_\theta}(\cdot,\boldsymbol{s}_t)$ from Lemma.3.4, which serves as the implicit policy evaluation for $\overline{\pi}$ with the lower values implying increased exploration and vice versa. Therefore if $\overline{\pi}_\varphi$ approximate $\overline{\pi}$ well enough, ITS can implicitly derive the value function from $\overline{\pi}_\varphi$ without explicit modeling Q^{π_θ}.

For simplicity, we first consider step-synchronous ITS with the policy π_θ, wherein the MCTS action node is token-level, *i.e.*, $\mathcal{A}_{\mathsf{MCTS}} = \mathcal{A}$. Specifically, given fixed θ and rounds of simulation (*i.e.*, $\lambda_N(\boldsymbol{s}_t)$ is also fixed), the ITS policy $\overline{\pi}(\cdot|\boldsymbol{s}_t)$, $\forall \boldsymbol{s}_t \in \mathcal{S}$ can be uniquely determined according to Lemma.3.4. Suppose that φ achieve the optimal parameter such that $\overline{\pi}_\varphi(\boldsymbol{a}_t|\boldsymbol{s}_t) = \overline{\pi}(\boldsymbol{a}_t|\boldsymbol{s}_t)$, it holds the observation

Lemma 4.1. *Suppose $(\boldsymbol{x},\boldsymbol{y}^{(w)})$ denotes a pair of prompt and its preferred response. For each state $\boldsymbol{s}_t^{(w)} = (\boldsymbol{x},\boldsymbol{y}_{<t}^{(w)})$, the action selects either the preferred token $\boldsymbol{a}_t^{(w)} = y_t^{(w)}$ or a random token $\hat{\boldsymbol{a}}_t$ sampled from $\overline{\pi}_{\varphi^*}(\cdot|\boldsymbol{s}_t^{(w)})$; and each state $\boldsymbol{s}_t^{(l)} = (\boldsymbol{x},\boldsymbol{y}_{<t}^{(l)})$, the action selects either the dispreferred token $\boldsymbol{a}_t^{(l)} = y_t^{(l)}$ or a random token $\check{\boldsymbol{a}}_t$ sampled from $\overline{\pi}_{\varphi^*}(\cdot|\boldsymbol{s}_t^{(l)})$. It holds*

$$R(\boldsymbol{x},\boldsymbol{y}^{(w)}) = \\ \sum_{t=1}^{T_w}\left(r(\boldsymbol{s}_t^{(w)},\hat{\boldsymbol{a}}_t)-\lambda_N(\boldsymbol{s}_t^{(w)})\left(\frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}-\frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}\right)\right); \\ R(\boldsymbol{x},\boldsymbol{y}^{(l)}) = \\ \sum_{t=1}^{T_l}\left(r(\boldsymbol{s}_t^{(l)},\check{\boldsymbol{a}}_t)-\lambda_N(\boldsymbol{s}_t^{(l)})\left(\frac{\pi_\theta(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}-\frac{\pi_\theta(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}\right)\right). \tag{14}$$

The lemma demonstrates that we may construct the accumulated reward of preference prompt-response pair by the ITS exploration starts from each state of the preferred and dispreferred sequences (*i.e.*, root or intermediate node). As demonstrated in (Rafailov et al., 2024a), DPO-like methods hold the dense reward re-parameterized as $r(\boldsymbol{s}_t,\boldsymbol{a}_t) = \beta\log\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t) - \beta\log\pi_{\text{ref}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$. Combine this with Lemma.4.1 leading to

Theorem 4.2. ***(Step-Synchronous IT-PO)*** *A prompt $\boldsymbol{x}$ drawn from D has a response pairs $\boldsymbol{y}^{(w)} \succ \boldsymbol{y}^{(l)}$. Given $\hat{\boldsymbol{A}}=\{\hat{\boldsymbol{a}}_t \sim \overline{\pi}_\varphi(\cdot|\boldsymbol{s}_t^{(w)})\}_{t=1}^{T_w}$ and $\check{\boldsymbol{A}}=\{\check{\boldsymbol{a}}_t \sim \overline{\pi}_\varphi(\cdot|\boldsymbol{s}_t^{(l)})\}_{t=1}^{T_l}$, it holds*

$$U_{\mathsf{ss}}(\boldsymbol{x},\boldsymbol{y}^{(w)},\boldsymbol{y}^{(l)}) = R(\boldsymbol{x},\boldsymbol{y}^{(w)}) - R(\boldsymbol{x},\boldsymbol{y}^{(l)}) \\ =\mu_w(\varphi,\theta) - \mu_l(\varphi,\theta) + \delta(\theta),$$

$$\text{s.t. } \mu_w(\varphi,\theta)=-\sum_{t=1}^{T_w}\lambda_N(\boldsymbol{s}_t^{(w)})\left(\frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}-\frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}\right)$$

$$\mu_l(\varphi,\theta)=-\sum_{t=1}^{T_l}\lambda_N(\boldsymbol{s}_t^{(l)})\left(\frac{\pi_\theta(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}-\frac{\pi_\theta(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}\right)$$

$$\delta(\theta)=\beta\left(\sum_{t=1}^{T_w}\log\frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\pi_{\text{ref}}(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}-\sum_{t=1}^{T_l}\log\frac{\pi_\theta(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}{\pi_{\text{ref}}(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}\right) \tag{15}$$

then the step-synchronous IT-PO is proposed by

$$\mathcal{L}_{\mathsf{ss\text{-}IT\text{-}PO}}(\varphi;\theta) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}^{(w)},\boldsymbol{y}^{(l)})\sim D,\hat{\boldsymbol{A}},\check{\boldsymbol{A}}\sim\overline{\pi}_\varphi} \\ -[(1-\epsilon)\log\sigma(U_{\mathsf{ss}}(\boldsymbol{x},\boldsymbol{y}^{(w)},\boldsymbol{y}^{(l)}))+\epsilon\log\sigma(U_{\mathsf{ss}}(\boldsymbol{x},\boldsymbol{y}^{(l)},\boldsymbol{y}^{(w)}))] \tag{16}$$

Theorem 4.2 establishes that **optimal approximation between $\overline{\pi}$ and $\overline{\pi}_\varphi$ must preserve their stochastic search strategies' conformity to the BT preference model**. The minimization of Eq.13 is also derived from cDPO that incorporates the margin to tolerate the preference noises possibly introduced by $\hat{\boldsymbol{A}}$, $\check{\boldsymbol{A}}$ drawn from $\overline{\pi}_\varphi$. It benefits $\overline{\pi}_\varphi$ to generate the high-quality responses that exceed the old data.

Step-Asynchronous ITS. MCTS algorithms significantly improve the LM's reasoning ability where the action space of each node focuses on sentence. It holds the asynchronous step between $\overline{\pi}$ and π_θ, *i.e.*, $\mathcal{A}_{\mathsf{MCTS}} = \bigcup_{n=1}^{N}\mathcal{A}^n$, where $\mathcal{A}^n$ is the sentence-level action space with n-length tokens and N indicates the maximum length of sentences. Observe that the token-level MDP could be also identified into the sentence level, where $\overline{\pi}$, π_θ, π_{ref} take their sentence-level actions by generating token sequences ended by "$[\backslash n]''$ token (Definition B.3 in Appendix). We re-frame their sentence-level policy as $\overline{\pi}_\varphi^S$, π_θ^S, π_{ref}^S so that given the state $\boldsymbol{s}_t \in \mathcal{S}_S$ and the sentence-level action $A_t = (a_t^{(i)})_{i=1}^{N_t} \in \mathcal{A}_S$ (N_t indicates the token number in the t-th sentence action and $a_t^{(i)}$ indicates the i-th token in the t-th sentence), and the sentence-level reward $r_S(s_t,A_t) = \sum_{i=1}^{N_t} r(s_t,a_t^{(i)})$.

Given this, we derive the sentence-level ITS optimized by step-asynchronous variant of Eq.4.2:

Theorem 4.3. *(**Step-Asynchronous-IT-PO**) A prompt $\boldsymbol{x}$ drawn from D has a response pairs $\boldsymbol{y}^{(\mathrm{w})} \succ \boldsymbol{y}^{(\mathrm{l})}$ composed of T_w^S, T_l^S sentences, respectively. $A_t^{(w)}$, $A_t^{(l)}$ denotes the t-th sentence in $\boldsymbol{y}^{(w)}$, $\boldsymbol{y}^{(l)}$, respectively. Suppose the state $\boldsymbol{s}_t^{(w)}$,$\boldsymbol{s}_t^{(l)}$ transmits along sentence-level MDP (Definition.B.3), $\boldsymbol{a}_t^{(w,i)}$/$\boldsymbol{a}_t^{(l,i)}$ indicates the i-th token in the $A_t^{(w)}$/$A_t^{(l)}$, $\boldsymbol{s}_t^{(w,i)}$/$\boldsymbol{s}_t^{(l,i)}$ denotes the sequential context ahead of $\boldsymbol{a}_t^{(w,i)}$/$\boldsymbol{a}_t^{(l,i)}$ in $\boldsymbol{y}^{(w)}$/$\boldsymbol{y}^{(l)}$. Suppose that the sentence-level policies π_{ref}^S, π_θ^S, $\overline{\pi}_\varphi^S$ are identified by LM-based token-selection policies π_{ref}, π_θ, $\overline{\pi}_\varphi$, respectively; $\hat{A}_t = (\hat{\boldsymbol{a}}_t^{(1)}, \cdots, \hat{\boldsymbol{a}}_t^{(\hat{N}_t)}) \sim \overline{\pi}_\varphi^S(\cdot|\boldsymbol{s}_t^{(w)}), \hat{\boldsymbol{s}}_t^{(0)} = \boldsymbol{s}_t^{(w)}, \hat{\boldsymbol{s}}_t^{(i+1)} = \mathcal{P}(\hat{\boldsymbol{s}}_t^{(i)}, \hat{\boldsymbol{a}}_t^{(i)}); \check{A}_t = (\check{\boldsymbol{a}}_t^{(1)}, \cdots, \check{\boldsymbol{a}}_t^{(\check{N}_t)}) \sim \overline{\pi}_\varphi^S(\cdot|\boldsymbol{s}_t^{(l)}), \check{\boldsymbol{s}}_t^{(0)} = \boldsymbol{s}_t^{(l)}, \check{\boldsymbol{s}}_t^{(i+1)} = \mathcal{P}(\check{\boldsymbol{s}}_t^{(i)}, \check{\boldsymbol{a}}_t^{(i)})$. It holds*

$$
\begin{aligned}
&U_{\mathsf{sa}}(\boldsymbol{x}, \boldsymbol{y}^{(w)}, \boldsymbol{y}^{(l)}) = R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) - R(\boldsymbol{x}, \boldsymbol{y}^{(l)}) \\
&= \mu_w^S(\varphi, \theta) - \mu_l^S(\varphi, \theta) + \delta^S(\theta), \\
\text{s.t. } &\mu_w^S(\varphi, \theta) = -\sum_{t=1}^{T_w^S} \lambda_N(\boldsymbol{s}_t^{(w)}) \Big(\prod_{i=1}^{|A_t^{(w)}|} \frac{\pi_\theta(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})} - \prod_{i=1}^{\hat{N}_t} \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} \Big) \\
&\mu_l^S(\varphi, \theta) = -\sum_{t=1}^{T_l^S} \lambda_N(\boldsymbol{s}_t^{(l)}) \Big(\prod_{i=1}^{|A_t^{(l)}|} \frac{\pi_\theta(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})} - \prod_{i=1}^{\check{N}_t} \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})} \Big) \\
&\delta^S(\theta) = \beta \Big(\sum_{t=1}^{T_w^S} \sum_{i=1}^{\hat{N}_t} \log \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\pi_{\mathrm{ref}}(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} - \sum_{t=1}^{T_l^S} \sum_{i=1}^{\check{N}_t} \log \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\pi_{\mathrm{ref}}(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})} \Big)
\end{aligned} \tag{17}
$$

then the step-synchronous IT-PO is proposed by

$$
\begin{aligned}
\mathcal{L}_{\text{sa-IT-PO}}(\varphi; \theta) = \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}^{(w)}, \boldsymbol{y}^{(l)}) \sim D, \{\hat{A}_t\}_{t=1}^{T_l^S}, \{\check{A}_t\}_{t=1}^{T_w^S} \sim \overline{\pi}_\varphi} \\
-[(1\text{–}\epsilon) \log \sigma(U_{\mathsf{ss}}(\boldsymbol{x}, \boldsymbol{y}^{(w)}, \boldsymbol{y}^{(l)})) + \epsilon \log \sigma(U_{\mathsf{ss}}(\boldsymbol{x}, \boldsymbol{y}^{(l)}, \boldsymbol{y}^{(w)}))]
\end{aligned} \tag{18}
$$

4.3. Self-Improved Training and Decoding

In Sec.4.1, 4.2, LM φ serves as a universal approximator of the ITS policy $\overline{\pi}$ across the state space. Since $\overline{\pi}$ behaves as the stochastic variant of AlphaZero-like tree search, we turn to discuss their common purpose, *i.e.*, distilling $\overline{\pi}_\varphi$ back into the policy π_θ to enhance both their performances.

Preference Policy Augmentation. Regular MCTS algorithms used to sample decision-making trajectory to refine the original policy π_θ, which is recently treated as a type of generalized policy improvement. Instead, (Grill et al., 2020b) used to distill their ITS policy by $-D_{KL}(\overline{\pi}, \pi_\theta)$. It eases the sampling process but also sacrifices the benefits derived from the high-quality response trajectories.

With this regards, we propose *preference policy augmentation* by resampling M responses $\{\boldsymbol{y}_k\}_{k=1}^K$ generated from each prompt $\boldsymbol{x} \sim \mathcal{D}$ by $\overline{\pi}_\varphi$. $\{\boldsymbol{y}_k\}_{k=1}^K$ are then constructed into pairs with their preference relations evaluated by U_{ss} or U_{as}, thus, $(\boldsymbol{y}_i \succ \boldsymbol{y}_j)$ holds if either $U_{\mathsf{ss}}(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j) > 0$ at the token level or $U_{\mathsf{as}}(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j) > 0$ at the sentence level. After ranking their value by U_{ss} or U_{as}, the top-M preference pairs are selected for each prompt $\boldsymbol{x}$ in $\mathcal{D}$. Then we collect them across all prompts in $\mathcal{D}$ to construct the ITS improved dataset $\mathcal{D}^+$. They join with $\mathcal{D}$ to refine the token-selection policy via learning θ with cDPO. We elaborate the IT-PO algorithm pipelines of its step-synchronous and step-asynchronous cases in our implementation in Appendix.C.

Self-Improvement in Gradient Analysis. Through analyzing the TI-PO loss function's gradients with respect to φ, we demonstrate how $\overline{\pi}_\varphi$ facilitates **the generated responses both diverse and better aligned** with ground-truth preferences. For clarity, our analysis is derived from the step-synchronous ITS using U_{ss} for preference alignment and focuses on the positive preference logit in $-\log[\sigma(U_{\mathsf{ss}})]$:

$$
\begin{aligned}
&\nabla_\varphi \Big(-\log[\sigma(U_{\mathsf{ss}}(\boldsymbol{x}, \boldsymbol{y}^{(w)})), \boldsymbol{y}^{(l)}] \Big) \\
&= (\nabla_\varphi \mu_w - \nabla_\varphi \mu_l) \cdot \underbrace{\nabla_{\triangle R}\big(-\log \sigma(\triangle R)\big)}_{\text{higher when reward estimate is wrong by } \varphi} \\
\text{s.t. } &\nabla_\varphi \mu_w = \sum_{t=1}^{T_w} \lambda_N(\boldsymbol{s}_t^{(w)}) \Big(\frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)}) \nabla_\varphi \log[\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})]}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})} \\
&\qquad - \frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)}) \nabla_\varphi \log[\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})]}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})} \Big) \\
&-\nabla_\varphi \mu_l = \sum_{t=1}^{T_l} \lambda_N(\boldsymbol{s}_t^{(l)}) \Big(-\frac{\pi_\theta(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)}) \nabla_\varphi \log[\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})]}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})} \\
&\qquad + \frac{\pi_\theta(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)}) \nabla_\varphi \log[\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})]}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})} \Big).
\end{aligned} \tag{19}
$$

Compared with DPO, the second multiplier observed from the decomposed formula achieves the identical gradient effects. $\nabla_\varphi \log[\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})]$ and $-\nabla_\varphi \log[\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})]$ similarly exist in the first term yet they take the likelihood increase and decrease effects with coefficients $\frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}$ and $\frac{\pi_\theta(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}$. It implies that when $\overline{\pi}_\varphi < \pi_\theta$ to the ground-truth preference pairs, IT-PO takes the *aggressive* likelihood influence to align $\overline{\pi}_\varphi$ and π_θ while the *conservative* likelihood influence if $\overline{\pi}_\varphi > \pi_\theta$. More importantly, the implicit tree sampled from the ground-truth preference nodes, *i.e.*, $\hat{\boldsymbol{a}}_t \sim \overline{\pi}_\varphi(\cdot|\boldsymbol{s}_t^{(w)})$, $\check{\boldsymbol{a}}_t \sim \overline{\pi}_\varphi(\cdot|\boldsymbol{s}_t^{(l)})$ also influence the likelihood: $-\nabla_\varphi \log[\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})]$ and $\nabla_\varphi \log[\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})]$ demonstrates that on account of the ground-truth preference nodes, $\overline{\pi}_\varphi$ learns to search lower preference child nodes from $\boldsymbol{s}_t^{(w)}$ yet higher preference child nodes from $\boldsymbol{s}_t^{(l)}$ with the same conservative-aggressive strategy. Hence $\overline{\pi}_\varphi$ generates certain responses from the preferred context yet explore more diverse responses when the context is not preferred.

ITS-guided Decoding. Beyond self-enhanced training, recent study on LLM-based MCTS (Feng et al., 2023) demonstrates promising results in improving the decoding process by MCTS. Motivated from this, we propose the stochastic-tree variant decoding from the spirits of their MCTS-α and MCTS-rollout strategies.

1). **ITS-α**. For each initial state $\boldsymbol{x}_{\mathsf{root}}$, the original MCTS-$\alpha$

decoding strategy applied Alpha-like tree search for policy evaluation then backup the visit count $n(\boldsymbol{s}, \boldsymbol{a})$ of the exponential visit distribution $\frac{n(\boldsymbol{s}_t,\boldsymbol{a})^{1/\gamma}}{\sum_{\boldsymbol{a}'} n(\boldsymbol{s}_t,\boldsymbol{a}')^{1/\gamma}}$ to guide decoding. Since $\overline{\pi}_\varphi$ is the universal approximator of $\overline{\pi}$, the stochastic variant of Alpha-like tree search. It is straightfoward to use the exponential version of $\overline{\pi}_\varphi$, $\frac{\exp(\log[\overline{\pi}_\varphi(\boldsymbol{a}|\boldsymbol{s}_t)]/\gamma)}{\sum_{\boldsymbol{a}'}\exp(\log[\overline{\pi}_\varphi(\boldsymbol{a}'|\boldsymbol{s}_t)]/\gamma)}$ (step-synchronous IT-PO) or $\frac{\exp(\log[\overline{\pi}^S_\varphi(\boldsymbol{a}|\boldsymbol{s}_t)]/\gamma)}{\sum_{\boldsymbol{a}'}\exp(\log[\overline{\pi}^S_\varphi(\boldsymbol{a}'|\boldsymbol{s}_t)]/\gamma)}$ (step-asynchronous IT-PO) in our decoding strategy.

2). **ITS-rollout**. When the root $\boldsymbol{x}_{\text{root}}$ and the intermediate state nodes sufficiently differ from the states visited in training, the simulation rounds N behaves more closely as zero to fail the approximation between $\hat{\pi}$ and $\overline{\pi}$ in Theorem.3.3. In this case, $\overline{\pi}_\varphi$ need to be updated to adapt the state $\boldsymbol{x}_{\text{root}}$ as the backup process in MCTS. Due to no value function available, we employ π_θ to implicitly evaluate arbitrary pairs of responses $\boldsymbol{y}_1$,$\boldsymbol{y}_2$ generated by $\overline{\pi}_\varphi(\cdot|\boldsymbol{x})$, then the sampled responses with preferences evaluated by θ would join the meta-update of $\overline{\pi}_\varphi$ to the state $\boldsymbol{x}$

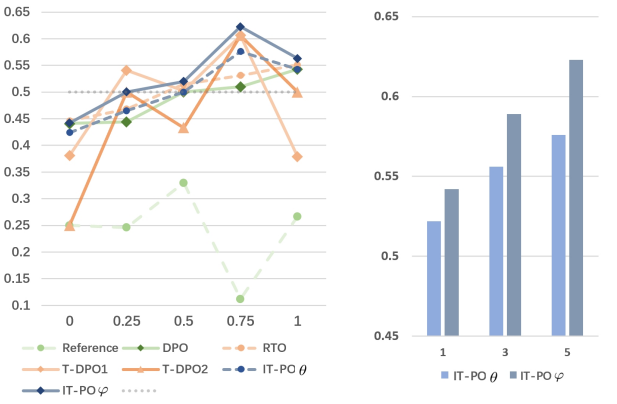
$$\begin{aligned}\varphi' \leftarrow & \varphi - \nabla_\varphi^{(\boldsymbol{x},\boldsymbol{y}_1\succ\boldsymbol{y}_2)}\mathcal{L}_{\text{sa-IT-PO}} \\ & \text{s.t. } \boldsymbol{y}_1 \succ \boldsymbol{y}_2 \ \text{ if } u_\theta(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2) > 0,\end{aligned} \tag{20}$$

which refreshes $\frac{\exp(\log[\overline{\pi}_{\varphi'}(\boldsymbol{a}|\boldsymbol{s}_t)]/\gamma)}{\sum_{\boldsymbol{a}'}\exp(\log[\overline{\pi}_{\varphi'}(\boldsymbol{a}'|\boldsymbol{s}_t)]/\gamma)}$ to facilitate our decoding strategy. ITS-rollout can be treated as test-time-training version of ITS-α. Its implementation first updates φ' by (20), then use ITS-α with φ' to facilitate the decoding process. In our experiment, 8 responses for each test prompt were generated to achieve self-training, resulting the extra $0.5\sim1$ hour as the decoding warm-up stage.

Provided the decoding probabilities obtained by ITS-rollout, the BF-S strategy (Breath-first Search) can be applied based on the *implicit advantage function* solely approximated by $\overline{\pi}_\varphi$ and π_θ. More specifically,

$$\begin{aligned}A^{\pi_\theta}(\boldsymbol{a},\boldsymbol{s}) =& Q^{\pi_\theta}(\boldsymbol{a},\boldsymbol{s}) - \mathbb{E}[Q^{\pi_\theta}(\boldsymbol{a}',\boldsymbol{s})|\boldsymbol{a}'\sim\pi_\theta(\cdot|\boldsymbol{s})] \\ \approx& Q^{\pi_\theta}(\boldsymbol{a},\boldsymbol{s}) - \frac{1}{N}\sum_{\boldsymbol{a}'\sim\pi_\theta(\cdot|\boldsymbol{s})} Q^{\pi_\theta}(\boldsymbol{a}',\boldsymbol{s}) \\ =& \Big(\alpha(\boldsymbol{s}) - \lambda_N(\boldsymbol{s})\frac{\pi_\theta(\boldsymbol{a}|\boldsymbol{s})}{\overline{\pi}(\boldsymbol{a}|\boldsymbol{s})}\Big) \\ & - \Big(\frac{1}{N}\sum_{\boldsymbol{a}'\sim\pi_\theta(\cdot|\boldsymbol{s})}\big(\alpha(\boldsymbol{s}) - \lambda_N(\boldsymbol{s})\frac{\pi_\theta(\boldsymbol{a}'|\boldsymbol{s})}{\overline{\pi}(\boldsymbol{a}'|\boldsymbol{s})}\big)\Big) \\ =& -\lambda_N(\boldsymbol{s})\Big(\frac{\pi_\theta(\boldsymbol{a}|\boldsymbol{s})}{\overline{\pi}(\boldsymbol{a}|\boldsymbol{s})} - \frac{1}{N}\sum_{\boldsymbol{a}'}\frac{\pi_\theta(\boldsymbol{a}'|\boldsymbol{s})}{\overline{\pi}(\boldsymbol{a}'|\boldsymbol{s})}\Big) \\ \approx& -\lambda_N(\boldsymbol{s})\Big(\frac{\pi_\theta(\boldsymbol{a}|\boldsymbol{s})}{\overline{\pi}_\varphi(\boldsymbol{a}|\boldsymbol{s})} - \frac{1}{N}\sum_{\boldsymbol{a}'}\frac{\pi_\theta(\boldsymbol{a}'|\boldsymbol{s})}{\overline{\pi}_\varphi(\boldsymbol{a}'|\boldsymbol{s})}\Big),\end{aligned} \tag{21}$$

where N indicates how many responses drawn from π_θ to approximate the advantage function. Such decoding strategy is closely related with wisdom of group-relative policy optimization (GRPO), a well-known value-free RL algorithm employed to train Deepseek (Shao et al., 2024).



(a) ***Win rate*** across temperature scaling

(b) ***Win rate*** across alternative policy improvement

Figure 2. **Win rate** measured by GPT-4 via the consistent prompts in previous studies: (a) **Win rate** of baselines decoding with different temperatures; (b) **Win rate** of π_θ and $\overline{\pi}_\varphi$ across the alternative phases of their preference policy distillation.

5. Experiments

In this section, we demonstrate the superiority of IT-PO from the step-synchronous (Theorem.4.2) and step-asynchronous (Theorem.4.3) perspectives. In the step-synchronous cases, IT-PO trains $\overline{\pi}_\varphi$ to perform token-level ITS in order to provide the fine-grained human preference alignment; in the step-asynchronous scenarios, we evaluate $\overline{\pi}_\varphi$ trained by IT-PO to perform sentence-level search in step-asynchronous scenarios, *i.e.*, mathematical reasoning and planning tasks where LLM-based MCTS algorithms are extensively used.

5.1. Experiments on Token-level Preference Alignment

Anthropic HH dataset (Bai et al., 2022)[1] consists of 170k dialogues between a human and an automated assistant, each of which presents as a history with alternative responses with respect to different preferences annotated by humans. We conduct the conventional evaluation setup using Pythia 2.8 (Biderman et al., 2023) as the base model, then each dialogue with its preferred completion in Anthropic HH training set is incorporated for supervised fine-tuning to derive three LMs: the parameter-frozen π_{ref} and parameter-initialized π_θ, $\overline{\pi}_\varphi$. In terms of preference alignment on token-level MDP, SS-IT-PO with respect to the LMs θ, φ are alternatively trained, then the IT-PO variants both compared with token-level DPO-family baselines, *i.e.*, DPO (Rafailov et al., 2024a), RTO (Zhong et al., 2024a), TDPO_1 and TDPO_2 (Zeng et al., 2024a).

The experiment primarily aims for three evaluation metrics: 1). **Accuracy**: we adopt the evaluation split in (Zeng et al., 2024a) to train all models then evaluate their performance

[1]https://huggingface.co/datasets/Anthropic/hh-rlhf

Table 1. Comparison in terms of the trade-off between Alignment (accuracy) and Diversity (entropy) on the Anthropic HH dataset. The ↑ indicates higher values are preferable.

Method	Alignment	Diversity
	Acc (%) ↑	Ent ↑
DPO	59.43	3.196
RTO	61.43	3.314
TDPO_1	60.08	4.727
TDPO_2	**67.33**	**4.915**
IT-PO θ (ours)	67.75	4.564
IT-PO φ (ours)	**69.12**	**5.315**

in terms of the accuracy on the generated responses relative to chosen completions in the test dataset; 2). **Diversity**: Nucleus sampling with $p = 0.95$ to generate 25 responses then the predictive entropy across the responses indicates the generation diversity. 3).**Win rate**: all baseline approaches are evaluated through GPT-4 against the chosen responses in the test set, so that $> 50\%$ implies the human preference alignment achieved in their performances.

Both LMs θ and φ utilize **ITS**-α to enhance the decoding strategy. The exponential rate γ =1 consistently across the evaluation of three metrics. This choice aligns with MCTS-α, which employs token-level tree search in alignment tasks. **Accuracy** and **Diversity** across the baselines are presented in Table 5.1. TDPO_2 demonstrates competitive performance, achieving alignment accuracy comparable to IT-PO θ, even exhibiting superior diversity. On the other hand, IT-PO φ, trained as the stochastic tree policy, outperforms TDPO_2 by a significant margin in both alignment and diversity metrics. It is because that LM θ in our framework performs more likely as a policy evaluator trained to boost the evolution of stochastic tree policy $\overline{\pi}_\varphi$.

To verify our assumption, we further observe their **Win rate** illustrated in Figure.2(a). Through scaling the temperature during inference, T-DPOs are found quite sensitive to the temperature, with lower performance than most baselines when the temperature value are extreme. Instead, both IT-PO θ and IT-PO φ consistently performs with robust value over 50% in 7 out of 10 cases where the other baselines underperform or even fail in the qualified **Win rate** level. In Figure.2 (b), we further observe the **Win rate** progress when IT-PO θ and IT-PO φ by through their alternative policy distillation. When no alternative strategy used (iter = 1), the win rate of IT-PO θ and IT-PO φ are solely on par with the win rates of DPO and RTO, respectively, which largely underperform T-DPO variants. While after five iterations of alternative preference policy distillation, **Win rate**s in IT-PO θ and IT-PO φ are both incredibly improved to exceed their early, even their teachers' performances. These observations are consistent with our analysis to φ's gradients.

Table 2. Performance comparison of different methods on GSM8k and Game24 datasets

Setting	Method	Performance(%) / # Tokens			
		GSM8k		Game24	
Path@1	CoT-greedy	41.4	98	12.7	76
	BFS-V	**52.5**	485	64.8	369
	MCTS-α	51.9	561	63.3	412
	MCTS-Rollout	47.8	3.4k	**71.3**	670
	ITS-α (**ours**)	**53.2**	561	67.6	380
	ITS-Rollout (**ours**)	51.6	3.8k	**73.2**	646
Equal-Token	$\text{CoT-SC}_{\text{MAJ}}$	46.8	500	14.6	684
	$\text{CoT-SC}_{\text{ORM}}$	**52.3**	500	50.6	684
	$\text{BFS-V}_{\text{ORM}}$	-	-	**70.90**	1.6k
	MCTS_{ORM}	-	-	69.34	649
	ITS-α (**ours**)	-	-	70.64	1.6k
	ITS-Rollout (**ours**)	-	-	**71.42**	698

5.2. Experiments on Mathematical Tasks

Our second experimental suite includes the tasks of mathematical reasoning on GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021) and mathematical planning on Game24 (Cobbe et al., 2021). Although the benchmarks are famous in evaluating LLM's reasoning capability by PRM (process reward model) (Lightman et al., 2023) and MCTS-based methods (Feng et al., 2023), they are nontrivial for DPO family since explicit value function can not be skipped to execute MCTS reasoning strategies.

Baselines and Evaluation. Since most DPO-based approaches can not adapt to LLM reasoning without explicit value modeling, we are more interested in the comparison between TI-PO and state-of-the-art tree-search (TS) LLM and Chain-of-Thought (CoT) (Wei et al., 2022) baselines in (Feng et al., 2023). More specifically, the evaluated baselines for GSM8K and Game24 are derived from LLAMA-7b [2] base model, specifically include CoT-greedy (greedy value search by CoT), BFV (Breath-first search with their learned value function), MCTS-α (AlphaZero-like tree search with their learned value function), MCTS-rollout (MCTS-α variant that allows the backup process happen in the intermediate step). As a counterpart, we incorporate the policies π_θ and $\overline{\pi}_\varphi$ with their LMs fine-tuned by cDPO (Eq.13) and step-asynchronous IT-PO objective (Eq.17), respectively, then achieve their alternative post-training via preference policy augmentation. After five iterations, the policy models jointly facilitate the decoding processes via **ITS-**α and **ITS-rollout** strategies. All baselines are evaluated in the single path setup via Path@1 and Equal-Token, the latter try to

[2]https://huggingface.co/meta-llama/Llama-2-7b

Table 3. Path@1 metric on Game24 with different node size.

Method	Performance(%) / # Tokens					
	width=6		width=20		width=50	
MCTS-α	41.6	243	63.3	412	74.5	573
MCTS-Rollout	43.8	401	71.3	670	80.7	833
BFS-V	43.2	206	64.8	370	74.6	528
ITS-α (**ours**)	46.1	267	67.6	380	75.0	647
ITS-Rollout (**ours**)	**48.9**	489	**73.2**	646	**81.8**	954

Table 4. The experimental results in MATH. All baselines employed Qwen1.5-32B as their base models.

Baselines/Decoding	greedy CoT	-α	-rollout
Qwen1.5-32B	36.1	-	-
MCTS- (Qwen1.5-32B)	-	36.0	36.7
ITS- (φ=Qwen1.5-32B)	-	39.8	40.2
ITS- (φ=Qwen1.5-32B)	-	37.9	38.2

compare the results with the similar scale computation consumption. For the evaluation on MATH, we construct the training set integrated with the training splits of GSM8K and MATH, then consider Qwen1.5-32B (Team, 2024) as our base model. Beyond this, we also introduced LLAMA3.1-8B as an alternative of φ to verify whether φ can be replaced by a smaller LLM. To this, we have the evaluated LLMs (θ,φ) defined by (Qwen1.5-32B,Qwen1.5-32B) and (Qwen1.5-32B,LLAMA3-8B), respectively. We also employed greedy CoT (3 shots), MCTS-α, and MCTS-rollout for their comparison.

Implementation. Distinct from human preference data with pairwise responses, reasoning tasks only consist of question prompts and its correct solution responses. It motivates us to reconfigure their data to adapt our preference optimization regime. Beyond this, since our "tree-search" strategy is indeed a stochastic policy, the search depth and breadth for training LLM φ were not limited in all datasets, while their decoding procedures *i.e.*, ITS-α and ITS-rollout, inherited the pruning setup derived from the MCTS baselines to address the heavy computation while the tree search runs for reasoning-oriented inference. Details refer to Appendix.C.

Results.According to the GSM8k results (Table 2), when examining Path@1 performance across all baseline methods, tree-search algorithms (excluding our proposed methods) don't show significant advantages over standard CoT approaches. MCTS variants actually perform worse than BFV, despite their higher computational requirements. However, the Game24 results tell a different story. In this task, CoT approaches perform poorly, largely because Game24's structure allows for wider and deeper search trees, which better showcases the strengths of tree-search algorithms (Table B.5). But interestingly, **ITS-α** and **ITS-rollout** consistently outperform other approaches, regardless of the default tree width and depth limits during inference.

This superior performance can be attributed to our IT-PO training regime. Unlike traditional approaches that slowly build sparse trees based on visit counts, our approximated $\overline{\pi}$ encourages broader exploration across the entire action space. This approach is computationally efficient since IT-PO only needs to sample the next sentence (rather than the complete reasoning path) from preferred/dispreferred contexts (as defined by $\hat{A}_t$ and $\check{A}_t$ in Eq. 4.3). This enables faster search and backup operations compared to standard MCTS. As a result, even with bounded search width and depth during inference, the stochastic policy demonstrates robust performance due to effective training simulation. In order to support our claim, we further ablate the maximum tree width and node size during inference to observe the performance variation across different baselines. As shown in Table.3,4, expanding the node size and tree width significantly boosts regular TS-LLM performances. While ITS-α and ITS-Rollout have already achieved impressive results with the small node size and and tree width. Beyond this, they also enjoy the performance growth along with the increasing exploration space.

In Table.5, we presented the evaluation in MATH that contains more difficult mathematical reasoning tasks. In terms of the greedy CoT results in LLAMA3-8B (20.5), some observations can be found in the table. First, we note that the MCTS strategy derived from GSM8K is unreliable in MATH, probably due to the conflict between the problem complexity and limited search width and depth during training, while ITS-α and ITS-rollout did not suffer from this problem. Second, with a weaker model (LLAMA3-8B) to implement LLM φ, IT-PO still enabled the improvement of LLM-based reasoning. It implies the feasibility of IT-PO. Specifically, despite avoiding value modeling, IT-PO introduces the extra LM φ instead of the single LM θ in standard DPO variants, leading to the increased computational and memory requirement. While the evidences in Table.5 demonstrated that using LM φ with the significantly smaller and weaker base model than LM θ, ITS-(θ =Qwen1.5-32B, φ = LLAMA3-8B) still yields the results very competitive with ITS-(θ = Qwen1.5-32B, φ = Qwen1.5-32B).

6. Concluding Remark

This paper presents a new RL-free methodology to equip DPO with MCTS interpreted as stochastic policy to better align LMs with human preferences, and simultaneously outperforms MCTS-based LLM reasoning baseline methods in both mathematical reasoning and planning tasks.

Acknowledgement

The research was supported in part by Guangdong S&T Programme (Grant No. 2024B0101010003); in part by The Major Key Project of PCL (No. PCL2024A04, PCL2025A02); in part by National Natural Science Foundation of China (NSFC) under Grant No.62206110, 62176103, 62377208, and 62276114; in part by the Science and Technology Planning Project of Guangzhou under grants 2024A04J9896, 2025A03J3565.

In particular, we thank all the reviewers for their constructive suggestions that help to improve this work.

Impact Statement

This work advances preference-based learning for large language models through improved exploration mechanism, which has implications for AI alignment and safer deployment of language technologies. By enhancing LLMs' reasoning and long-form generation capabilities through more principled exploration and preference learning, our approach could lead to more reliable and controllable language models. However, improved reasoning capabilities could also enable more sophisticated text generation that may be misused. The integration of MCTS with preference learning represents a step toward more transparent optimization of language model behavior, though careful consideration should be given to the selection of preference data to avoid encoding harmful biases. We believe the technical advances presented here can contribute positively to the development of more capable and aligned language models when deployed thoughtfully with appropriate safeguards and oversight.

References

Abdulhai, M., White, I., Snell, C., Sun, C., Hong, J., Zhai, Y., Xu, K., and Levine, S. Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models. *arXiv preprint arXiv:2311.18232*, 2023.

Amini, A., Vieira, T., and Cotterell, R. Direct preference optimization with an offset, 2024. URL https://arxiv.org/abs/2402.10571.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

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

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

Chen, G., Liao, M., Li, C., and Fan, K. Step-level value preference optimization for mathematical reasoning. *arXiv preprint arXiv:2406.10858*, 2024a.

Chen, G., Liao, M., Li, C., and Fan, K. Step-level value preference optimization for mathematical reasoning, 2024b. URL https://arxiv.org/abs/2406.10858.

Chen, Z., Wang, K., Wang, X., Peng, P., Izquierdo, E., and Lin, L. Deep co-space: Sample mining across feature transformation for semi-supervised learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 28 (10):2667–2678, 2017.

Chen, Z., Huang, X., Guan, Q., Lin, L., and Luo, W. A retrospect to multi-prompt learning across vision and language. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22190–22201, 2023.

Chen, Z., Zheng, Y., Lai, Z.-R., Guan, Q., and Lin, L. Diagnosing and rectifying fake ood invariance: A restructured causal approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11471–11479, 2024c.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 2021.

Coulom, R. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pp. 72–83. Springer, 2006.

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., Shum, K., and Zhang, T. Raft: Reward ranked finetuning for generative foundation model alignment, 2023. URL https://arxiv.org/abs/2304.06767.

Dong, H., Xiong, W., Pang, B., Wang, H., Zhao, H., Zhou, Y., Jiang, N., Sahoo, D., Xiong, C., and Zhang, T. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. Implementation matters in deep policy gradients: A case study on ppo and trpo, 2020. URL https://arxiv.org/abs/2005.12729.

Feng, X., Wan, Z., Wen, M., McAleer, S. M., Wen, Y., Zhang, W., and Wang, J. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.

Feng, X., Wan, Z., Wen, M., McAleer, S. M., Wen, Y., Zhang, W., and Wang, J. Alphazero-like tree-search can guide large language model decoding and training, 2024. URL https://arxiv.org/abs/2309.17179.

Grill, J.-B., Altché, F., Tang, Y., Hubert, T., Valko, M., Antonoglou, I., and Munos, R. Monte-carlo tree search as regularized policy optimization. pp. 3769–3778, 2020a.

Grill, J.-B., Altché, F., Tang, Y., Hubert, T., Valko, M., Antonoglou, I., and Munos, R. Monte-carlo tree search as regularized policy optimization. In *International Conference on Machine Learning*, pp. 3769–3778. PMLR, 2020b.

Hao, S., Gu, Y., Ma, H., et al. Reasoning with language models is planning with a world model. *arXiv preprint arXiv:2305.14992*, 2023.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Kirk, R., Mediratta, I., Nalmpantis, C., Luketina, J., Hambro, E., Grefenstette, E., and Raileanu, R. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.

Kocsis, L. and Szepesvári, C. Bandit based monte-carlo planning. In *European conference on machine learning*, pp. 282–293. Springer, 2006.

Liao, W., Chu, X., and Wang, Y. Tpo: Aligning large language models with multi-branch & multi-step preference trees. *arXiv preprint arXiv:2410.12854*, 2024.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Liu, A., Bai, H., Lu, Z., Sun, Y., Kong, X., Wang, S., Shan, J., Jose, A. M., Liu, X., Wen, L., Yu, P. S., and Cao, M. Tis-dpo: Token-level importance sampling for direct preference optimization with estimated weights, 2024a. URL https://arxiv.org/abs/2410.04350.

Liu, Z., Lu, M., Zhang, S., Liu, B., Guo, H., Yang, Y., Blanchet, J., and Wang, Z. Provably mitigating overoptimization in rlhf: Your sft loss is implicitly an adversarial regularizer, 2024b. URL https://arxiv.org/abs/2405.16436.

Mitchell, E. A note on dpo with noisy preferences & relationship to ipo. *arXiv preprint arXiv:2304.12345*, 2023a.

Mitchell, E. A note on dpo with noisy preferences & relationship to ipo, 2023b.

Ouyang, Y., Wang, L., Yang, F., Zhao, P., Huang, C., Liu, J., Pang, B., Yang, Y., Zhan, Y., Sun, H., Lin, Q., Rajmohan, S., Deng, W., Zhang, D., Sun, F., and Zhang, Q. Token-level proximal policy optimization for query generation, 2024. URL https://arxiv.org/abs/2411.00722.

Pan, L., Albalak, A., Wang, X., and Wang, W. Y. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Rafailov, R., Hejna, J., Park, R., and Finn, C. Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*, 2024a.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024b.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024c.

Rosin, C. D. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3): 203–230, 2011.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017a.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017b.

Tafjord, O., Mishra, B. D., and Clark, P. Proofwriter: Generating implications, proofs, and abductive statements over natural language. *arXiv preprint arXiv:2012.13048*, 2020.

Team, Q. Introducing qwen1.5, February 2024. URL https://qwenlm.github.io/blog/qwen1.5/.

Uesato, J., Kushman, N., Kumar, R., et al. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Wang, X., Song, L., Tian, Y., Yu, D., Peng, B., Mi, H., Huang, F., and Yu, D. Towards self-improvement of llms via mcts: Leveraging stepwise knowledge with curriculum preference learning, 2024a. URL https://arxiv.org/abs/2410.06508.

Wang, X., Song, L., Tian, Y., Yu, D., Peng, B., Mi, H., Huang, F., and Yu, D. Towards self-improvement of llms via mcts: Leveraging stepwise knowledge with curriculum preference learning, 2024b. URL https://arxiv.org/abs/2410.06508.

Wang, Y., Liu, Q., and Jin, C. Is rlhf more difficult than standard rl? a theoretical perspective. *Advances in Neural Information Processing Systems*, 36:76006–76032, 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Xie, Y., Goyal, A., Zheng, W., Kan, M.-Y., Lillicrap, T. P., Kawaguchi, K., and Shieh, M. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024a.

Xie, Y., Goyal, A., Zheng, W., Kan, M.-Y., Lillicrap, T. P., Kawaguchi, K., and Shieh, M. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024b.

Xie, Y., Goyal, A., Zheng, W., Kan, M.-Y., Lillicrap, T. P., Kawaguchi, K., and Shieh, M. Monte carlo tree search boosts reasoning via iterative preference learning, 2024c. URL https://arxiv.org/abs/2405.00451.

Yao, S., Yu, D., Zhao, J., et al. Tree of thoughts: Deliberate problem-solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

Yuan, W., Pang, R. Y., Cho, K., Li, X., Sukhbaatar, S., Xu, J., and Weston, J. Self-rewarding language models, 2024. URL https://arxiv.org/abs/2401.10020.

Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., and Huang, F. Rrhf: Rank responses to align language models with human feedback without tears, 2023. URL https://arxiv.org/abs/2304.05302.

Zeng, Y., Liu, G., Ma, W., Yang, N., Zhang, H., and Wang, J. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*, 2024a.

Zeng, Y., Liu, G., Ma, W., Yang, N., Zhang, H., and Wang, J. Token-level direct preference optimization, 2024b. URL https://arxiv.org/abs/2404.11999.

Zhang, D., Zhoubian, S., Hu, Z., Yue, Y., Dong, Y., and Tang, J. Rest-mcts*: Llm self-training via process reward guided tree search. *arXiv preprint arXiv:2406.03816*, 2024a.

Zhang, X., Du, C., Pang, T., Liu, Q., Gao, W., and Lin, M. Chain of preference optimization: Improving chain-of-thought reasoning in llms, 2024b. URL https://arxiv.org/abs/2406.09136.

Zhong, H., Feng, G., Xiong, W., Cheng, X., Zhao, L., He, D., Bian, J., and Wang, L. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*, 2024a.

Zhong, H., Feng, G., Xiong, W., Cheng, X., Zhao, L., He, D., Bian, J., and Wang, L. Dpo meets ppo: Reinforced token optimization for rlhf, 2024b. URL https://arxiv.org/abs/2404.18922.

Zhou, W., Zhang, S., Zhao, L., and Meng, T. T-reg: Preference optimization with token-level reward regularization, 2024. URL https://arxiv.org/abs/2412.02685.

Ziebart, B. D. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

A. Related Work

The integration of search-based optimization techniques in language model alignment has significantly advanced AI alignment, with Reinforcement Learning with Human Feedback (RLHF)(Bai et al., 2022; Wang et al., 2023; Kirk et al., 2023; Dong et al., 2024) being a widely used approach. RLHF employs reward models trained from human feedback to optimize model behavior through reinforcement learning, typically using Proximal Policy Optimization (PPO) (Zhong et al., 2024b). However, RLHF has been criticized for instability, sample inefficiency, and over-optimization issues (Engstrom et al., 2020; Liu et al., 2024b; Chen et al., 2023; 2017; 2024c). To address these challenges, methods like Reward Ranked FineTuning (RAFT) (Dong et al., 2023) and Rank Responses to align Human Feedback (RRHF) (Yuan et al., 2023) have been proposed to refine ranking-based optimization without explicit reinforcement learning. More recently, Direct Preference Optimization (DPO) (Rafailov et al., 2024c; Amini et al., 2024) has emerged as an alternative, allowing language models to be aligned directly from human preference data without requiring a reward model.

Unlike PPO, which operates within a reinforcement learning framework by optimizing policies through reward feedback, DPO reformulates the alignment problem as a supervised learning task, making policy updates more stable while maintaining alignment with human preferences. Further refinements, such as processing the paragraph at token level with methods such as TDPO (Zeng et al., 2024b), T-REG (Zhou et al., 2024), TPPO (Ouyang et al., 2024), and TIS-DPO (Liu et al., 2024a), improve efficiency by incorporating token-wise adjustments, improving preference alignment in a manner that contrasts with RLHF's reliance on policy gradient updates.

Monte Carlo Tree Search (MCTS) has been extensively applied in decision-making tasks, particularly in game-playing AI, as demonstrated by its success in AlphaGo (Silver et al., 2017b), AlphaZero (Silver et al., 2017a), and MuZero (Schrittwieser et al., 2020). Recent advancements have expanded its application to large language models (LLMs) for structured text generation, such as in AlphaZero-like Tree-Search for LLMs (TS-LLM) (Feng et al., 2023). Similarly, Xie et al. (Xie et al., 2024c) propose an iterative preference learning approach, using MCTS to refine step-wise reasoning capabilities in LLMs. Further developments by Wang et al. (Wang et al., 2024b) introduce self-improvement techniques where LLMs leverage MCTS for preference-guided reinforcement learning, while Chen et al. (Chen et al., 2024b) focus on step-level value preference optimization, allowing fine-grained preference learning at intermediate steps. Zhang et al. (Zhang et al., 2024b) extend this concept with chain preference optimization, improving long-range decision-making for complex reasoning tasks. Lastly, Liao et al. (Liao et al., 2024) introduce Tree-based Preference Optimization (TPO), which integrates MCTS with preference alignment techniques to refine LLM outputs progressively. Despite these advancements, a key challenge remains: these approaches still require learning an explicit reward value function, which is critical for optimizing the search process and improving the efficiency of LLM training and inference.

Our work proposes *Implicit Tree Search (ITS)*, which integrates stochastic policy optimization with AlphaZero-like search principles while eliminating explicit tree structures. ITS leverages reversed KL-divergence constraints and a stochastic sampling policy to enhance exploration expressiveness, addressing cold-start issues common in MCTS-based methods. Similar ideas have been explored in Monte Carlo-based regularized policy optimization (Grill et al., 2020a), (Wang et al., 2024a) they use pairwise training framework that enables LLMs to self-improve through MCTS behavior but ITS extends these principles to preference learning in language models.

Furthermore, our approach aligns with broader research on AI safety and preference-based learning (Yuan et al., 2024; Xie et al., 2024a; Chen et al., 2024a; Mitchell, 2023a). ITS represents a novel *search-and-learn* paradigm that improves structured reasoning in language models without relying on explicit tree expansion, bridging the gap between MCTS and modern alignment techniques.

C.1. Fundamentals of Monte Carlo Tree-Search Methods

Monte Carlo Tree Search (MCTS) has been widely adopted as an effective strategy for solving problems requiring sequential decision-making and planning. Traditional MCTS operations, as introduced by (Kocsis & Szepesvári, 2006) and (Coulom, 2006), include four key steps: selection, expansion, simulation, and backpropagation. However, these operations can be adapted for more advanced frameworks like AlphaZero (Silver et al., 2017b), which incorporates a learned value function and policy network to guide the search process.

To address challenges in balancing exploration and exploitation, we utilize a variant of MCTS with a modified Predictor Upper Confidence Tree (PUCT) algorithm (Rosin, 2011). The algorithm selects actions a_t at each node s_t as follows:

$$a_t = \arg\max_a \big(Q(s_t, a) + U(s_t, a)\big),$$

where $U(s_t, a)$ is calculated using the formula:

$$U(s,a) = c_{\text{puct}} \cdot \pi_\theta(s,a) \cdot \frac{\sqrt{\sum_b N(s,b)}}{1+N(s,a)}.$$

Here, $N(s,a)$ represents the visit count of action a at node s, and c_{puct} is a constant controlling exploration, as defined by:

$$c_{\text{puct}} = \log\left(\frac{\sum_b N(s,b) + c_{\text{base}} + 1}{c_{\text{base}}}\right) + c_{\text{init}}.$$

Node Expansion and Assessment: Upon reaching a leaf node s_L, if it is not terminal, the tree is expanded by generating possible successor nodes. The value of the leaf node is then estimated using a neural network. For terminal nodes, a reward function $R(s_L)$ is used, or an Outcome Reward Model (ORM) serves as an approximation (Uesato et al., 2022).

Value Propagation: Once a leaf node is assessed, the computed values are propagated back through the path $s_0, s_1, \ldots, s_L$. For each node, the visit count is updated as:

$$N(s_t, a) = N(s_t, a) + 1,$$

and the cumulative action value is updated as:

$$W(s_t, a) = W(s_t, a) + v(s_L).$$

The mean action value is then computed as:

$$Q(s_t, a) = \frac{W(s_t, a)}{N(s_t, a)}.$$

This combination of learned value functions and search-based methods enables efficient exploration of large decision spaces, as demonstrated in AlphaZero (Silver et al., 2017b) and its applications to tree-search-guided language models (Yao et al., 2023; Hao et al., 2023).

B. Proofs.

B.1. Proof of Proposition 3.1

Proof. Given $r \in \mathbb{R}$ and $\forall \boldsymbol{a}' \in \mathcal{A}_{\text{MCTS}}/\{\boldsymbol{a}\}$, $n(\boldsymbol{s}, \boldsymbol{a}')$ are fixed, the probability of empirical visit distribution to the action $\boldsymbol{a}$ denotes as $\frac{1+n(\boldsymbol{s},\boldsymbol{a})^r}{|\mathcal{A}_{\mathsf{MCTS}}|+\sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}} n(\boldsymbol{s},\boldsymbol{a}')^r}$. It holds

$$\frac{1+n(\boldsymbol{s},\boldsymbol{a})^r}{|\mathcal{A}_{\mathsf{MCTS}}| + \sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}} n(\boldsymbol{s},\boldsymbol{a}')^r} = 1 - \frac{1 - (|\mathcal{A}_{\mathsf{MCTS}}| + \sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}/\{a\}} n(\boldsymbol{s},\boldsymbol{a}')^r)}{|\mathcal{A}_{\mathsf{MCTS}}| + \sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}/\{a\}} n(\boldsymbol{s},\boldsymbol{a}')^r + n(\boldsymbol{s},\boldsymbol{a})^r} \tag{22}$$

Since $|\mathcal{A}_{\mathsf{MCTS}}| + \sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}/\{a\}} n(\boldsymbol{s},\boldsymbol{a}')^r$ is constant, the exponential empirical visit distribution only changes along with $n(\boldsymbol{s},\boldsymbol{a}')^r$ changes. Due to $|\mathcal{A}_{\mathsf{MCTS}}| > 1$ as "tree" definition, $|\mathcal{A}_{\mathsf{MCTS}}| + \sum_{a'\in\mathcal{A}_{\mathsf{MCTS}}/\{a\}} n(\boldsymbol{s},\boldsymbol{a}')^r > 1$. Note that $f(x) = 1 - \frac{1-c}{c+x}$ is bijective with respect to $c > 1$ and when $r \neq 0$, $\{n^r : n \in \mathbb{Z} \cup \{0\}\}$ holds the consistent cardinality with $\mathbb{Z}$. The proposition has been proved. □

B.2. Proof of Proposition 3.2

Proof. To prove the proposition, we only need to prove the following lemma:

Lemma B.1. *Given positive integers a, b, and $\{c_i\}_{i=1}^a$, define*

$$f(r) \;=\; \frac{1+b^r}{a+1+\sum_{i=1}^a c_i^r + b^r}.$$

We claim that for all real r, the value of $f(r)$ lies strictly within the open interval $\left(\frac{1}{a+1}, 1\right)$

Proof. 1. **As $r \to -\infty$:** Since $b^r \to 0$ and $c_i^r \to 0$ for each i, we have

$$\lim_{r\to-\infty} f(r) \;=\; \lim_{r\to-\infty} \frac{1+b^r}{a+1+\sum_{i=1}^{a} c_i^r + b^r} \;=\; \frac{1+0}{a+1+0+0} \;=\; \frac{1}{a+1}.$$

Thus $f(r)$ never goes below $1/(a+1)$.

2. **As $r \to +\infty$:** Let $M = \max\{b, c_1, c_2, \ldots, c_a\}$. For sufficiently large r, M^r dominates b^r and each c_i^r, so the numerator and denominator in $f(r)$ are asymptotically proportional to M^r, giving

$$\lim_{r\to+\infty} f(r) \;=\; 1.$$

Since b^r and c_i^r increase with r, one can show $f(r)$ itself is strictly increasing in r. Consequently, its image over $r \in \mathbb{R}$ is precisely

$$\left(\tfrac{1}{a+1},\, 1\right).$$

Counterexample. Because $f(r)$ is bounded below by $\frac{1}{a+1}$, any real x with

$$0 < x < \frac{1}{a+1}$$

cannot be realized by $f(r)$. For instance, choose

$$x_0 \;=\; \frac{1}{2(a+1)}.$$

We observe

$$0 < \frac{1}{2(a+1)} < \frac{1}{a+1},$$

but there is no real r for which $f(r) = x_0$. Hence, values in $\left(0, \frac{1}{a+1}\right)$ are not attainable. □

Set $a = |\mathcal{A}_{\mathsf{MCTS}}| - 1$, $b = n(\boldsymbol{s}, \boldsymbol{a})$, and $c_i = n(\boldsymbol{s}, \boldsymbol{a}_i)$, $\forall \boldsymbol{a}_i \in \mathcal{A}_{\mathsf{MCTS}}/\{\boldsymbol{a}\}$, the proposition has been proved. □

B.3. Proof of Lemma 4.1

Proof. As discussed in Lemma.3.4,

$$\overline{\pi}_\varphi(\boldsymbol{a}_t|\boldsymbol{s}_t) = \lambda_N(\boldsymbol{s}_t)\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\alpha(\boldsymbol{s}_t) - Q^{\pi_\theta}(\boldsymbol{s}_t, \boldsymbol{a}_t)} \leftrightarrows r(\boldsymbol{s}_t, \boldsymbol{a}_t) = \alpha(\boldsymbol{s}_t) - V^{\pi_\theta}(\boldsymbol{s}_t) - \lambda_N(\boldsymbol{s}_t)\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\overline{\pi}_\varphi(\boldsymbol{a}_t|\boldsymbol{s}_t)}, \tag{23}$$

where $Q^{\pi_\theta}(\boldsymbol{s}_t, \boldsymbol{a}_t) = r(\boldsymbol{s}_t, \boldsymbol{a}_t) + V^{\pi_\theta}(\boldsymbol{s}_t)$. Provided $(\boldsymbol{x}, \boldsymbol{y}^{(w)})$ denoting a pair of prompt and its preferred response, we construct the preferred state-action trajectory along with the token-level MDP, *i.e.*, $\{(\boldsymbol{s}_t^{(w)}, \boldsymbol{a}_t^{(w)})\}_{t=1}^{T_w}$ *w.r.t.* $\boldsymbol{s}_t^{(w)} = (\boldsymbol{x}, \boldsymbol{y}_{<t}^{(w)})$ and $\boldsymbol{a}_t^{(w)} = y_t^{(w)}$. Then based on the definition, for the t-th state in the preferred state-action trajectory, we sample another action $\hat{\boldsymbol{a}}_i$ from the parameterized ITS exploration policy $\overline{\pi}_\varphi$ *i.e.*, $\hat{\boldsymbol{a}}_t \sim \overline{\pi}_\varphi(\cdot|\boldsymbol{s}_t^{(w)})$. For the preferred state-action trajectory, Eq.23 holds at each step so that

$$R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) = \sum_{t=1}^{T_w} r(\boldsymbol{s}_t^{(w)}, \boldsymbol{a}_t^{(w)}) = \sum_{t=1}^{T_w} \left(\alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)})\frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_{\varphi^*}(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}\right) \tag{24}$$

and $\forall t \in T_w$,

$$r(\boldsymbol{s}_t^{(w)}, \hat{\boldsymbol{a}}_t) = \alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)})\frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_{\varphi^*}(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}. \tag{25}$$

Therefore

$$
\begin{aligned}
R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) - \sum_{t=1}^{T_w} r(\boldsymbol{s}_t^{(w)}, \hat{\boldsymbol{a}}_t) &= \sum_{t=1}^{T_w} \left(\alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_{\varphi^*}(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})} \right) \\
&\quad - \sum_{t=1}^{T_w} \left(\alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_{\varphi^*}(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})} \right) \\
&= -\lambda_N(\boldsymbol{s}_t^{(w)}) \left(\frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_{\varphi^*}(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})} - \frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_{\varphi^*}(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})} \right).
\end{aligned} \tag{26}
$$

Therefore $R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) = \sum_{t=1}^{T_w} \left(r(\boldsymbol{s}_t^{(w)}, \hat{\boldsymbol{a}}_t) - \lambda_N(\boldsymbol{s}_t^{(w)}) \left(\frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})} - \frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})} \right) \right)$.

The dispreferred state-action trajectory from $(\boldsymbol{x}, \boldsymbol{y}^l)$ can be also constructed in the same routine, *i.e.*, $\{(\boldsymbol{s}_t^{(l)}, \boldsymbol{a}_t^{(l)})\}_{t=1}^{T_l}$ *w.r.t.* $\boldsymbol{s}_t^{(l)} = (\boldsymbol{x}, \boldsymbol{y}_{<t}^{(l)})$ and $\boldsymbol{a}_t^{(l)} = y_t^{(l)}$. For the i-th state in the dispreferred state-action trajectory, we sample another action $\check{\boldsymbol{a}}_i$ from $\overline{\pi}_\varphi$ *i.e.*, $\check{\boldsymbol{a}}_i \sim \overline{\pi}_\varphi(\cdot|\boldsymbol{s}_t^{(l)})$. For the dispreferred state-action trajectory, it holds the mirror formulations of Eq.24-26 , then from the same deduction, $R(\boldsymbol{x}, \boldsymbol{y}^{(l)}) = \sum_{t=1}^{T_l} \left(r(\boldsymbol{s}_t^{(l)}, \check{\boldsymbol{a}}_t) - \lambda_N(\boldsymbol{s}_t^{(l)}) \left(\frac{\pi_\theta(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})} - \frac{\pi_\theta(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})} \right) \right)$ is hold. □

B.4. Proof of Theorem 4.2

Proof. Let's introduce the dense reward reparameterzation theory in (Rafailov et al., 2024a):

Lemma B.2. *(Reparameterzed dense reward) (Rafailov et al., 2024a) Given a reference policy π_{ref} and a parameter $\beta > 0$ all reward classes consistent with the Plackett-Luce (and Bradley-Terry) models, the step-wise reward $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ can be represented with the a re-parameterization*

$$
r(\boldsymbol{s}_t, \boldsymbol{a}_t) = \beta \log \pi(\boldsymbol{a}_t|\boldsymbol{s}_t) - \beta \log \pi_{\text{ref}}(\boldsymbol{a}_t|\boldsymbol{s}_t) \tag{27}
$$

within the token MDP where $V^*(\boldsymbol{s}_t) = 0$ for all terminal state.

Here we combine Lemma.4.1 and Lemma.B.2 to certify our theorem. Specifically, $V^{\pi_\theta}(\boldsymbol{s}_T) = V^*(\boldsymbol{s}_T) = 0$ for all terminal state given any sequence, since θ is optimized from DPO-like algorithms. By Lemma.B.2, we have the dense reward decomposition

$$
r(\boldsymbol{s}_t^{(w)}, \hat{\boldsymbol{a}}_t) = \beta \log \pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)}) - \beta \log \pi_{\text{ref}}(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)}) = \beta \log \frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}{\pi_{\text{ref}}(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}, \forall t \in [T_w]. \tag{28}
$$

The decomposition holds in the dispreferred states such that

$$
r(\boldsymbol{s}_t^{(l)}, \check{\boldsymbol{a}}_t) = \beta \log \pi_\theta(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)}) - \beta \log \pi_{\text{ref}}(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)}) = \beta \log \frac{\pi_\theta(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}{\pi_{\text{ref}}(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}, \forall t \in [T_l]. \tag{29}
$$

where the dispreferred state-action trajectory along with the token-level MDP, *i.e.*, $\{(\boldsymbol{s}_t^{(l)}, \boldsymbol{a}_t^{(l)})\}_{t=1}^{T_l}$ is constructed by

$\boldsymbol{s}_t^{(l)} = (\boldsymbol{x}, \boldsymbol{y}_{<t}^{(l)})$ and $\boldsymbol{a}_t^{(l)} = y_t^{(l)}, \forall t \in [T_l]$. Given this, we have

$$
\begin{aligned}
U_{\mathsf{ss}}(\boldsymbol{x}, \boldsymbol{y}^{(w)}, \boldsymbol{y}^{(l)}) =& R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) - R(\boldsymbol{x}, \boldsymbol{y}^{(l)}) \\
=& \left(R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) - \sum_{t=1}^{T_w} r(\boldsymbol{s}_t^{(w)}, \hat{\boldsymbol{a}}_t) \right) + \left(\sum_{t=1}^{T_w} r(\boldsymbol{s}_t^{(w)}, \hat{\boldsymbol{a}}_t) - \sum_{t=1}^{T_l} r(\boldsymbol{s}_t^{(l)}, \check{\boldsymbol{a}}_t) \right) + \left(\sum_{t=1}^{T_l} r(\boldsymbol{s}_t^{(l)}, \check{\boldsymbol{a}}_t) - R(\boldsymbol{x}, \boldsymbol{y}^{(l)}) \right) \\
=& \underbrace{\sum_{t=1}^{T_w} -\lambda_N(\boldsymbol{s}_t^{(w)}) \left(\frac{\pi_\theta(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w)}|\boldsymbol{s}_t^{(w)})} - \frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})} \right)}_{\mu_w(\varphi;\theta)} + \underbrace{\sum_{t=1}^{T_l} \lambda_N(\boldsymbol{s}_t^{(l)}) \left(\frac{\pi_\theta(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l)}|\boldsymbol{s}_t^{(l)})} - \frac{\pi_\theta(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})} \right)}_{\mu_l(\varphi;\theta)} \\
& + \underbrace{\sum_{t=1}^{T_w} \beta \log \frac{\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})}{\pi_{\text{ref}}(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(w)})} - \sum_{t=1}^{T_l} \beta \log \frac{\pi_\theta(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}{\pi_{\text{ref}}(\check{\boldsymbol{a}}_t|\boldsymbol{s}_t^{(l)})}}_{\delta_\theta}
\end{aligned}
\tag{30}
$$

The theorem has been proved. □

B.5. Proof of Theorem.4.3

Here we provide the formal proofs to the sentence-level version of Lemma.4.1 and Theorem.3.3. They are based on the definition of Sentence-level MDP:

Definition B.3. (Sentence-level MDP) Suppose that $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{T})$ denotes the token-level MDP. It identifies the sentence-level MDP $\mathcal{M}_S = (\mathcal{S}_S, \mathcal{A}_S, \mathcal{P}_S, \mathcal{R}_S, \mathcal{T}_S)$ as

$$
\mathcal{S}_S = \bigcup_{n=1}^{+\infty} \mathcal{S}^n; \mathcal{A}_S = \mathcal{A}_{\mathsf{MCTS}} = \bigcup_{n=1}^{N} \mathcal{A}^n; \mathcal{P}_S : \mathcal{S}_S \times \mathcal{A}_S \to \mathcal{S}_S; \mathcal{R}_S : \mathcal{S}_S \times \mathcal{A}_S \to \mathbb{R}; \mathcal{T}_S = \mathcal{T}. \tag{31}
$$

where $\mathcal{S}^n$ indicates the sequence composed of n sentences ended by "$\backslash n''$ in order and obviously, $\mathcal{S}^n \subset \mathcal{S}$. $\mathcal{A}^n$ is the sentence-level action space with n-length tokens and N indicates the maximum length of sentences. The transition kernel $\mathcal{P}_S$ and reward space $\mathcal{R}_S$ holds $\mathcal{P}_S \subset \mathcal{S} \times \mathcal{A}_S \to \mathcal{S}$ and $\mathcal{R}_S \subset \mathcal{S} \times \mathcal{A}_S \to \mathbb{R}$. To this, if we consider the policies hold the decomposition $\pi^S(A_t|\boldsymbol{s}_t) = \prod_{i=1}^{N_t} \pi(a_t^{(i)}|\boldsymbol{s}_t^{(i)})$ ($A_t = (a_t^{(i)})_{i=1}^{N_t} \in \mathcal{A}_S$ denotes the sentence-level action, N_t indicates the token number in the selected sentence action, $a_t^{(i)}$ indicates the i-th token in the t-th sentence, and $\boldsymbol{s}^{(i)} = (\boldsymbol{s}, a_t^{(1)}), \cdots, a_t^{(i-1)}).$) $\mathcal{R}_S \subset \mathcal{S} \times \mathcal{A}_S \to \mathbb{R}$, to this, we define $r^S(A_i, \boldsymbol{s}) \in \mathcal{R}_S$ such that $r^S(\boldsymbol{s}_t, A_i) = \sum_i^{N_t} r(\boldsymbol{s}_t^{(i)}, a_t^{(i)})$.

Lemma B.4. *Given the prompt $\boldsymbol{x}$ with its preference response pair $\boldsymbol{y}^{(w)}$, $\boldsymbol{y}^{(l)}$ composed of T_w^S, T_l^S sentences, $A_t^{(w)}$, $A_t^{(l)}$ denotes the t-th sentence in $\boldsymbol{y}^{(w)}$, $\boldsymbol{y}^{(l)}$, respectively. Suppose the state $\boldsymbol{s}_t^{(w)}$,$\boldsymbol{s}_t^{(l)}$ transmits along sentence-level MDP, $\boldsymbol{a}_t^{(w,i)}$/$\boldsymbol{a}_t^{(l,i)}$ indicates the i-th token in the $A_t^{(w)}$/$A_t^{(l)}$, $\boldsymbol{s}_t^{(w,i)}$/$\boldsymbol{s}_t^{(l,i)}$ denotes the sequential context ahead of $\boldsymbol{a}_t^{(w,i)}$/$\boldsymbol{a}_t^{(l,i)}$ in $\boldsymbol{y}^{(w)}$/$\boldsymbol{y}^{(l)}$. Suppose that the sentence-level policies π_{ref}^S, π_θ^S, $\overline{\pi}_\varphi^S$ are identified by LM-based token-selection policies π_{ref}, π_θ, $\overline{\pi}_\varphi$, respectively; $\hat{A}_t = (\hat{\boldsymbol{a}}_t^{(1)}, \cdots, \hat{\boldsymbol{a}}_t^{(\hat{N}_t)}) \sim \overline{\pi}_\varphi^S(\cdot|\boldsymbol{s}_t^{(w)})$,$\hat{\boldsymbol{s}}_t^{(0)} = \boldsymbol{s}_t^{(w)}$,$\hat{\boldsymbol{s}}_t^{(i+1)} = \mathcal{P}(\hat{\boldsymbol{s}}_t^{(i)}, \hat{\boldsymbol{a}}_t^{(i)})$;$\check{A}_t = (\check{\boldsymbol{a}}_t^{(1)}, \cdots, \check{\boldsymbol{a}}_t^{(\check{N}_t)}) \sim \overline{\pi}_\varphi^S(\cdot|\boldsymbol{s}_t^{(l)})$,$\check{\boldsymbol{s}}_t^{(0)} = \boldsymbol{s}_t^{(l)}$,$\check{\boldsymbol{s}}_t^{(i+1)} = \mathcal{P}(\check{\boldsymbol{s}}_t^{(i)}, \check{\boldsymbol{a}}_t^{(i)})$. It holds*

$$
\begin{aligned}
R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) &= \sum_{t=1}^{T_w^S} \left(\sum_{i=1}^{\hat{N}_t} r(\hat{\boldsymbol{s}}_t^{(i)}, \hat{\boldsymbol{a}}_t^{(i)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \left(\prod_{i=1}^{|A_t^{(w)}|} \frac{\pi_\theta(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})} - \prod_{i=1}^{\hat{N}_t} \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} \right) \right) \\
R(\boldsymbol{x}, \boldsymbol{y}^{(l)}) &= \sum_{t=1}^{T_l^S} \left(\sum_{i=1}^{\check{N}_t} r(\check{\boldsymbol{s}}_t^{(i)}, \check{\boldsymbol{a}}_t^{(i)}) - \lambda_N(\boldsymbol{s}_t^{(l)}) \left(\prod_{i=1}^{|A_t^{(l)}|} \frac{\pi_\theta(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})} - \prod_{i=1}^{\check{N}_t} \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})} \right) \right)
\end{aligned}
\tag{32}
$$

Proof. Suppose that $Q^{\pi_\theta^S}(\boldsymbol{s}_t, \cdot)$ denotes the state-action value function on the sentence-level policy π_θ^S with respect to LM θ. Due to π_θ^S defined with the action space $\mathcal{A}_S = \mathcal{A}_{\mathsf{MCTS}}$, it results in

$$
\overline{\pi}_\varphi^S(A_t|\boldsymbol{s}_t) = \lambda_N(\boldsymbol{s}_t) \frac{\pi_\theta^S(A_t|\boldsymbol{s}_t)}{\alpha(\boldsymbol{s}_t) - Q^{\pi_\theta^S}(\boldsymbol{s}_t, A_t)} \leftrightarrows r^S(\boldsymbol{s}_t, A_t) = \alpha(\boldsymbol{s}_t) - V^{\pi_\theta^S}(\boldsymbol{s}_t) - \lambda_N(\boldsymbol{s}_t) \frac{\pi_\theta^S(A_t|\boldsymbol{s}_t)}{\overline{\pi}_\varphi^S(A_t|\boldsymbol{s}_t)}, \tag{33}
$$

where $A_t = (a_t^{(i)})_{i=1}^{N_t} \in \mathcal{A}_S$ denotes the sentence-level action, N_t indicates the token number in the selected sentence action, and $a_t^{(i)}$ indicates the i-th token in the t-th sentence. It holds $Q^{\pi_\theta^S}(\boldsymbol{s}_t, A_t) = r^S(\boldsymbol{s}_t, A_t) + V^{\pi_\theta}(\boldsymbol{s}_t) = \sum_i^{N_t} r(\boldsymbol{s}_t^{(i)}, a_t^{(i)}) + V^{\pi_\theta^S}(\boldsymbol{s}_t)$.

Provided $(\boldsymbol{x}, \boldsymbol{y}^{(w)})$ denoting a pair of prompt and its preferred response, we construct the preferred state-action trajectory along with the sentence-level MDP. In particular, suppose T_w^S denotes the number of sentences included in the preferred response $\boldsymbol{y}^{(w)}$, which can be uniquely identified through the location of "$\backslash n''$ tokens. We denote them as a sequence of sentence-level actions $\{A_t^{(w)}\}_{t=1}^{T_w^S}$. $\forall t \in T_w^S$, and make the decomposition such that $\{(\boldsymbol{s}_t^{(w)}, A_t^{(w)})\}_{t=1}^{T_w^S}$ with respect to

$$\begin{aligned} A_t^{(w)} =& (y^{(w)}_{\sum_{j=1}^{t-1} N_j^{(w)}+1}, \cdots, y^{(w)}_{\sum_{j=1}^{t-1} N_j^{(w)}+N_t^{(w)}}) \\ =& (\boldsymbol{a}_t^{(w,1)}, \cdots, \boldsymbol{a}_t^{(w,N_t^{(w)})}) \\ \boldsymbol{s}_t^{(w)} =& (\boldsymbol{x}, A_1^{(w)}, \cdots, A_{t-1}^{(w)}), \end{aligned} \tag{34}$$

Then for each $(\boldsymbol{s}_t^{(w)}, A_t^{(w)})$

$$\boldsymbol{s}_t^{(w,i)} = (\boldsymbol{s}_t^{(w)}, \boldsymbol{a}_t^{(w,1)}, \cdots, \boldsymbol{a}_t^{(w,i-1)}), \forall i \in [N_t^{(w)}]. \tag{35}$$

where $\boldsymbol{a}_t^{(w,i)}$ denotes the response token $y^{(w)}_{\sum_{j=1}^{t-1} N_j^{(w)}+i}$ in $\boldsymbol{y}^{(w)}$.

Then we provide the derivation

$$\begin{aligned} R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) =& \sum_{t=1}^{T_w^S} \sum_{i}^{N_t^{(w)}} r(\boldsymbol{s}_t^{(i)}, a_t^{(i)}) = \sum_{t=1}^{T_w^S} r^S(\boldsymbol{s}_t^{(w)}, A_t) \\ =& \sum_{t=1}^{T_w^S} \left(\alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta^S}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \frac{\pi_\theta^S(A_t^{(w)}|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi^S(A_t^{(w)}|\boldsymbol{s}_t^{(w)})} \right) \\ =& \sum_{t=1}^{T_w^S} \left(\alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \frac{\prod_{i=1}^{N_t^{(w)}} \pi_\theta(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})}{\prod_{i=1}^{N_t^{(w)}} \overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})} \right) \\ =& \sum_{t=1}^{T_w^S} \left(\alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \prod_{i=1}^{N_t^{(w)}} \frac{\pi_\theta(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})} \right) \end{aligned} \tag{36}$$

Beyond this, for the t-th state in the preferred state-action trajectory, we sample another action $\hat{A}_t$ from the parameterized ITS exploration policy $\overline{\pi}_\varphi^S$ *i.e.*, $\hat{A}_t = (\hat{\boldsymbol{a}}_t^1, \cdots, \hat{\boldsymbol{a}}_t^{(\hat{N}_t)}) \sim \overline{\pi}_\varphi^S(\cdot|\boldsymbol{s}_t^{(w)})$. It can be telescoped into $\{\hat{\boldsymbol{a}}_t^{(i)} \sim \overline{\pi}_\varphi(\cdot|\hat{\boldsymbol{s}}_t^{(i)})\}_{i=1}^{\hat{N}_t}$ with respect to $\hat{\boldsymbol{s}}_t^{(i)} = (\boldsymbol{x}, A_1^{(w)}, \cdots, A_{t-1}^{(w)}, \hat{\boldsymbol{a}}_t^{(1)}, \cdots, \hat{\boldsymbol{a}}_t^{(i-1)})$. Therefore $\forall t \in [T_w^S]$, it holds

$$\begin{aligned} r^S(\boldsymbol{s}_t^{(w)}, \hat{A}_t) = \sum_{i=1}^{N_t} r(\hat{\boldsymbol{a}}_t^{(i)}, \hat{\boldsymbol{s}}_t^{(i)}) =& \alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta^S}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \frac{\pi_\theta^S(\hat{A}_t|\boldsymbol{s}_t^{(w)})}{\overline{\pi}_\varphi^S(\hat{A}_t|\boldsymbol{s}_t^{(w)})} \\ =& \alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta^S}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \frac{\prod_{i=1}^{\hat{N}_t} \pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\prod_{i=1}^{\hat{N}_t} \overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} \\ =& \alpha(\boldsymbol{s}_t^{(w)}) - V^{\pi_\theta^S}(\boldsymbol{s}_t^{(w)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \prod_{i=1}^{\hat{N}_t} \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} \end{aligned} \tag{37}$$

Derived from the similar routine in Eq.26, we combine Eq.36, 37 to obtain

$$R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) = \sum_{t=1}^{T_w^S} \left(\sum_{i=1}^{\hat{N}_t} r(\hat{\boldsymbol{s}}_t^{(i)}, \hat{\boldsymbol{a}}_t^{(i)}) - \lambda_N(\boldsymbol{s}_t^{(w)}) \left(\prod_{i=1}^{N_t^{(w)}} \frac{\pi_\theta(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})} - \prod_{i=1}^{\hat{N}_t} \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} \right) \right) \tag{38}$$

For the dispreferred response $\boldsymbol{y}^{(l)}$, it refers to the mirror notation of $(w) \to (l)$ to construct the dispreferred state-action trajectory $\{(\boldsymbol{s}_t^{(l)}, A_t^{(l)})\}_{t=1}^{T_l^S}$ and for each $(\boldsymbol{s}_t^{(l)}, A_t^{(l)})$, $\boldsymbol{a}_t^{(l,i)} = y^{(w)}_{\sum_{j=1}^{t-1} N_j^{(l)}+i}$ and $\boldsymbol{s}_t^{(l,i)} = (\boldsymbol{s}_t^{(l)}, \boldsymbol{a}_t^{(l,1)}, \cdots, \boldsymbol{a}_t^{(l,i-1)}), \forall i \in [N_t^{(l)}]$ in $\boldsymbol{y}^{(l)}$. Given this, for the t-th state in the dispreferred state-action trajectory, instead of $A_t^{(l)}$, we sample another sentence action $\check{A}_t$ from the parameterized ITS exploration policy $\overline{\pi}_\varphi^S$ *i.e.*, $\check{A}_t = (\hat{\boldsymbol{a}}_t^1, \cdots, \check{\boldsymbol{a}}_t^{(\check{N}_t)}) \sim \overline{\pi}_\varphi^S(\cdot|\boldsymbol{s}_t^{(l)})$. It can be telescoped into $\{\check{\boldsymbol{a}}_t^{(i)} \sim \overline{\pi}_\varphi(\cdot|\check{\boldsymbol{s}}_t^{(i)})\}_{i=1}^{\check{N}_t}$ with respect to $\check{\boldsymbol{s}}_t^{(i)} = (\boldsymbol{x}, A_1^{(l)}, \cdots, A_{t-1}^{(l)}, \check{\boldsymbol{a}}_t^{(1)}, \cdots, \check{\boldsymbol{a}}_t^{(i-1)})$. From the same deduction routine, it holds

$$r^S(\boldsymbol{s}_t^{(l)}, \check{A}_t) = \alpha(\boldsymbol{s}_t^{(l)}) - V^{\pi_\theta^S}(\boldsymbol{s}_t^{(l)}) - \lambda_N(\boldsymbol{s}_t^{(l)}) \prod_{i=1}^{\check{N}_t} \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}, \ \forall t \in [T_l^S] \tag{39}$$

and

$$R(\boldsymbol{x}, \boldsymbol{y}^{(l)}) = \sum_{t=1}^{T_l^S} \left(\sum_{i=1}^{\check{N}_t} r(\check{\boldsymbol{s}}_t^{(i)}, \check{\boldsymbol{a}}_t^{(i)}) - \lambda_N(\boldsymbol{s}_t^{(l)}) \left(\prod_{i=1}^{N_t^{(l)}} \frac{\pi_\theta(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})} - \prod_{i=1}^{\check{N}_t} \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})} \right) \right) \tag{40}$$

□

On top of Lemma.B.4, we prove Theorem 4.3 as follows

Proof. According to Lemm.B.2, we have

$$\begin{aligned} r(\hat{\boldsymbol{s}}_t^{(i)}, \hat{\boldsymbol{a}}_t^{(i)}) &= \beta \log \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\pi_{\text{ref}}(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}, \mathsf{s.t.} \forall t \in T_w^S, \forall i \in \hat{N}_t; \\ r(\check{\boldsymbol{s}}_t^{(i)}, \check{\boldsymbol{a}}_t^{(i)}) &= \beta \log \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\pi_{\text{ref}}(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}, \mathsf{s.t.} \forall t \in T_l^S, \forall i \in \check{N}_t. \end{aligned} \tag{41}$$

Therefore

$$\begin{aligned} &U_{\mathsf{sa}}(\boldsymbol{x}, \boldsymbol{y}^{(w)}, \boldsymbol{y}^{(l)}) = R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) - R(\boldsymbol{x}, \boldsymbol{y}^{(l)}) \\ =& \left(R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) - \sum_{t=1}^{T_w^S} \sum_{i=1}^{\hat{N}_t} r(\hat{\boldsymbol{s}}_t^{(i)}, \hat{\boldsymbol{a}}_t^{(i)}) \right) + \left(\sum_{t=1}^{T_w^S} \sum_{i=1}^{\hat{N}_t} r(\hat{\boldsymbol{s}}_t^{(i)}, \hat{\boldsymbol{a}}_t^{(i)}) - \sum_{t=1}^{T_l^S} \sum_{i=1}^{\check{N}_t} r(\check{\boldsymbol{s}}_t^{(i)}, \check{\boldsymbol{a}}_t^{(i)}) \right) + \left(\sum_{t=1}^{T_l^S} \sum_{i=1}^{\check{N}_t} r(\check{\boldsymbol{s}}_t^{(i)}, \check{\boldsymbol{a}}_t^{(i)}) - R(\boldsymbol{x}, \boldsymbol{y}^{(l)}) \right) \\ =& \left(R(\boldsymbol{x}, \boldsymbol{y}^{(w)}) - \sum_{t=1}^{T_w^S} \sum_{i=1}^{\hat{N}_t} r(\hat{\boldsymbol{s}}_t^{(i)}, \hat{\boldsymbol{a}}_t^{(i)}) \right) + \left(R(\boldsymbol{x}, \boldsymbol{y}^{(l)}) - \sum_{t=1}^{T_l^S} \sum_{i=1}^{\check{N}_t} r(\check{\boldsymbol{s}}_t^{(i)}, \check{\boldsymbol{a}}_t^{(i)}) \right) \\ &\qquad + \beta \left(\sum_{t=1}^{T_w^S} \sum_{i=1}^{\hat{N}_t} \log \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\pi_{\text{ref}}(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} - \sum_{t=1}^{T_l^S} \sum_{i=1}^{\check{N}_t} \log \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\pi_{\text{ref}}(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})} \right) \\ =& \underbrace{\sum_{t=1}^{T_w^S} -\lambda_N(\boldsymbol{s}_t^{(w)}) \left(\prod_{i=1}^{N_t^{(w)}} \frac{\pi_\theta(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})} - \prod_{i=1}^{\hat{N}_t} \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} \right)}_{\mu_w^S(\varphi,\theta)} + \underbrace{\sum_{t=1}^{T_l^S} \lambda_N(\boldsymbol{s}_t^{(l)}) \left(\prod_{i=1}^{N_t^{(l)}} \frac{\pi_\theta(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})} - \prod_{i=1}^{\check{N}_t} \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})} \right)}_{-\mu_l^S(\varphi,\theta)} \\ &\qquad + \underbrace{\beta \left(\sum_{t=1}^{T_w^S} \sum_{i=1}^{\hat{N}_t} \log \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\pi_{\text{ref}}(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} - \sum_{t=1}^{T_l^S} \sum_{i=1}^{\check{N}_t} \log \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\pi_{\text{ref}}(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})} \right)}_{\delta^S(\theta)} \end{aligned} \tag{42}$$

Set up $N_i^{(w)} = |A_t^{(w)}|$ and $N_i^{(l)} = |A_t^{(l)}|$, then the theorem has been proved. □

Algorithm 1 The algorithm pipeline of IT-PO

Input: preference dataset $\mathcal{D}$; pre-trained LLMs θ, φ.
Hyper-parameters: ϵ; K; the batch size M to construct the batch of cDPO; the number of alternative training n_{alter}.
Output: LLMs θ^*, φ^*.
Initialize LLMs θ, φ by SF with the prompt-response pairs drawn from $\mathcal{D}$, $N_{\mathsf{alter}} = 0$;
Minimize $\mathcal{L}_{\mathsf{cDPO}}(\theta)$ to post-train LLM θ with the prompt and its pairwise responses drawn from the dataset $\mathcal{D}$;
repeat
  Implicit-Tree Preference Optimization:
  repeat
    Construct the training batch by M triplets drawn from $\mathcal{D}$;
    For each triplet $(\boldsymbol{x}, \boldsymbol{y}^{(w)}, \boldsymbol{y}^{(l)})$, construct $\{(\boldsymbol{a}_t^{(w)}, \boldsymbol{s}_t^{(w)})\}_{t=1}^{T_w}$, $\{(\boldsymbol{a}_t^{(l)}, \boldsymbol{s}_t^{(l)})\}_{t=1}^{T_l}$ based on the rule of token-level MDP (*w.r.t.* $\mathcal{L}_{\mathsf{ss\text{-}IT\text{-}PO}}(\varphi;\theta)$ for LM alignment) or sentence-level MDP (*w.r.t.* $\mathcal{L}_{\mathsf{as\text{-}IT\text{-}PO}}(\varphi;\theta)$ for LM Reasoning);
    Set $K_w = \frac{K}{T_w}, \forall t \in T_w$ (**parallel**), draw K_w tokens from $\overline{\pi}_\varphi(\cdot|\boldsymbol{s}_t^{(w)})$ to construct the samples of token-level action random variable $\hat{\boldsymbol{a}}_t$, or draw K_w sentences from $\overline{\pi}_\varphi^S(\cdot|\boldsymbol{s}_t^{(w)})$ to construct the samples of the sentence-level action random variable $\hat{A}_t$;
    Set $K_l = \frac{K}{T_l}, \forall t \in T_l$ (**parallel**), draw K_l tokens from $\overline{\pi}_\varphi(\cdot|\boldsymbol{s}_t^{(l)})$ to construct the samples of token-level action random variable $\check{\boldsymbol{a}}_t$, or draw K_l sentences from $\overline{\pi}_\varphi^S(\cdot|\boldsymbol{s}_t^{(l)})$ to construct the samples of the sentence-level action random variable $\check{A}_t$;
    Fix θ, minimize $\mathcal{L}_{\mathsf{ss\text{-}IT\text{-}PO}}(\varphi;\theta)$ to update φ for LM alignment or minimize $\mathcal{L}_{\mathsf{as\text{-}IT\text{-}PO}}(\varphi;\theta)$ to update φ for LM reasoning;
  until The number of training epoches reach the same of $\mathcal{L}_{\mathsf{cDPO}}(\theta)$.
  Preference Policy Augmentation:
  For each prompt $\boldsymbol{x}$ in $\mathcal{D}$, generate K responses $\{\boldsymbol{y}_k\}_{i=1}^K$ using $\overline{\pi}_\varphi$;
  For each prompt $\boldsymbol{x}$ in $\mathcal{D}$ and $\forall i \neq j \in [K]$, calculate $U_{\mathsf{ss}}(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j)$ or $U_{\mathsf{as}}(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j)$;
  Build the token-level policy augmentation pool $\mathcal{D}_{\mathsf{ss}}^+ = \{(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j) | \forall i \neq j \in [K], U_{\mathsf{ss}}(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j) > 0, (\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j) \in \mathsf{top}_K(\{U_{\mathsf{ss}}(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j)\}_{i\neq j\in[K]})$, or build the sentence-level policy augmentation pool $\mathcal{D}_{\mathsf{as}}^+ = \{(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j) | \forall i \neq j \in [K], U_{\mathsf{as}}(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j) > 0, (\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j) \in \mathsf{top}_K(\{U_{\mathsf{as}}(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j)\}_{i\neq j\in[K]})$
  Minimize $\mathcal{L}_{\mathsf{cDPO}}(\theta)$ to post-train LLM θ with the prompt and its pairwise responses drawn from $\mathcal{D} \cup \mathcal{D}_{\mathsf{ss}}^+$ or $\mathcal{D} \cup \mathcal{D}_{\mathsf{as}}^+$;
  $N_{\mathsf{alter}} = N_{\mathsf{alter}} + 1$;
until $N_{\mathsf{alter}} = n_{\mathsf{alter}}$
$\theta^* \leftarrow \theta$, $\varphi^* \leftarrow \varphi$.

C. Implementation

The algorithm pipeline of IT-PO are presented in Algorithm.1. The algorithm implementation almost holds the consistency with its theoretical foundation except for two details:(1) the specification of $\lambda_N(\boldsymbol{s}_t)$; (2) the φ's gradient implementation in $\mathcal{L}_{\mathsf{ss\text{-}IT\text{-}PO}}(\varphi;\theta)$ and $\mathcal{L}_{\mathsf{as\text{-}IT\text{-}PO}}(\varphi;\theta)$.

Suppose that $\lambda_N(\boldsymbol{s}_t^{(w)}), \lambda_N(\boldsymbol{s}_t^{(l)})$ denote the number of reasoning paths generated from the state $\boldsymbol{s}_t$, which consists of a prompt $\boldsymbol{x}$ and its responses $\boldsymbol{y}^{(w)}, \boldsymbol{y}^{(l)}$ with their contents in the previous $t-1$ steps. For each preference pair, we set $\forall t \in \{1, \ldots, T_w\}, \lambda_N(\boldsymbol{s}_t^{(w)}) = K_w$ and $\forall t \in \{1, \ldots, T_l\}, \lambda_N(\boldsymbol{s}_t^{(l)}) = K_l$, where K_w, K_l denote how many search start from the t-th leaf nodes $\boldsymbol{s}_t^{(w)}, \boldsymbol{s}_t^{(l)}$, respectively. Notice that T_w, T_l change with respect to the preference pair. So IT-PO adaptively configure $K_w = \frac{K}{T_w}, K_l = \frac{K}{T_l}$ to balance the optimization with different response lengths in $\boldsymbol{y}^{(w)}, \boldsymbol{y}^{(l)}$ for each pair. We set $K = 8$, inspired from the number of sampled responses for each prompt in many RLHF implementations.

In φ's gradient analysis in $\mathcal{L}_{\mathsf{ss\text{-}IT\text{-}PO}}(\varphi;\theta)$, the gradient of φ consists of terms in the form $\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\overline{\pi}_\varphi(\boldsymbol{a}_t|\boldsymbol{s}_t)} \nabla \log[\bar{\pi}(\boldsymbol{a}_t|\boldsymbol{s}_t)]$. Due to $\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\overline{\pi}_\varphi(\boldsymbol{a}_t|\boldsymbol{s}_t)} \in (0, +\infty)$, updating the models with $\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\overline{\pi}_\varphi(\boldsymbol{a}_t|\boldsymbol{s}_t)} \nabla \log[\bar{\pi}(\boldsymbol{a}_t|\boldsymbol{s}_t)]$ may suffer from exploding/vanishing gradients. To this, we used their logarithmic scaling $\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\overline{\pi}_\varphi(a_t|s_t)} \rightarrow \exp(\log(\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)) - \log(\overline{\pi}_\varphi(\boldsymbol{a}_t|\boldsymbol{s}_t)))$ to ensure the less sensitive

Table 5. Task setups. The node, tree max width, and tree max depth are search space parameters. The max tree-width and tree-depth follow the empirical experience in (Feng et al., 2023).

Task	Task Category	Train/test size	Node	Tree Max width	Tree Max depth
GSM8k	Mathematical Reasoning	7.5k / 1.3k	Sentence	6	8
Game24	Mathematical Planning	1.0k / 0.3k	Sentence	20	4
MATH	Mathematical Reasoning	- / -	Sentence	6	8

Table 6. Performance Comparison on ProofWriter and Chess Endgame Tasks

Setting	Baselines	ProofWriter (Acc %)	Chess Endgame (Win rate %)
Path@1	CoT-greedy	37.72	58.14
	BFS-V	48.94	67.75
	MCTS-α	66.71	96.90
	MCTS-rollout	69.23	98.76
	ITS-α (ours)	71.77	99.21
	ITS-rollout (ours)	75.31	99.83
Equal-Token	CoT-SC-MAJ	36.50	9.84
	CoT-SC-MAJ	36.58	73.80
	BFS-V-ORM	63.42	93.18
	MCTS-ORM	60.86	94.26
	ITS-α (ours)	74.26	96.48
	ITS-rollout (ours)	78.15	98.57

update ratio. In φ's gradient analysis in $\mathcal{L}_{\text{as-IT-PO}}(\varphi;\theta)$, it can derive the similar gradient as

$$\begin{aligned}\nabla_\varphi \mu_w^S(\varphi,\theta) = \sum_{t=1}^{T_w^S} \lambda_N(\boldsymbol{s}_t^{(w)}) \Bigg(\prod_{i=1}^{|A_t^{(w)}|} \frac{\pi_\theta(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})} \Big(\sum_{j=1}^{|A_t^{(w)}|} \nabla \log[\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,j)}|\boldsymbol{s}_t^{(w,j)})] \Big) \\ - \prod_{i=1}^{\hat{N}_t} \frac{\pi_\theta(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t^{(i)}|\hat{\boldsymbol{s}}_t^{(i)})} \Big(\sum_{j=1}^{\hat{N}_t} \nabla \log[\overline{\pi}_\varphi(\hat{\boldsymbol{a}}_t^{(j)}|\hat{\boldsymbol{s}}_t^{(j)})] \Big) \Bigg); \\ -\nabla_\varphi \mu_l^S(\varphi,\theta) = -\sum_{t=1}^{T_l^S} \lambda_N(\boldsymbol{s}_t^{(l)}) \Bigg(\prod_{i=1}^{|A_t^{(l)}|} \frac{\pi_\theta(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l,i)}|\boldsymbol{s}_t^{(l,i)})} \Big(\sum_{j=1}^{|A_t^{(l)}|} \nabla \log[\overline{\pi}_\varphi(\boldsymbol{a}_t^{(l,j)}|\boldsymbol{s}_t^{(l,j)})] \Big) \\ + \prod_{i=1}^{\check{N}_t} \frac{\pi_\theta(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})}{\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t^{(i)}|\check{\boldsymbol{s}}_t^{(i)})} \Big(\sum_{j=1}^{\check{N}_t} \nabla \log[\overline{\pi}_\varphi(\check{\boldsymbol{a}}_t^{(j)}|\check{\boldsymbol{s}}_t^{(j)})] \Big) \Bigg),\end{aligned} \tag{43}$$

where we also took the logarithmic scaling for the gradients in their implementation, *e.g.*,

$$\prod_{i=1}^{|A_t^{(w)}|} \frac{\pi_\theta(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})}{\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})} \to \prod_{i=1}^{|A_t^{(w)}|} \exp\big(\log(\pi_\theta(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)})) - \log(\overline{\pi}_\varphi(\boldsymbol{a}_t^{(w,i)}|\boldsymbol{s}_t^{(w,i)}))\big) \tag{44}$$

and so do others.

To train our policy networks in GSM8K, Game24, and MATH, we propose the preference data reconfigured from their training set. For each question $\boldsymbol{x}$ with correct response $\boldsymbol{y}$, we compute $-\log \pi_{\text{ref}}(\boldsymbol{y}'|\boldsymbol{x})$ across all responses y' from another

question to generate the hard negative preference pairs, then take them to initialize π_θ via cDPO. We set ϵ=0 to human alignment task, yet set ϵ=0.25 to mathematical reasoning and planning tasks. As for tree-based decoding, we employ the same tree-width pruning strategy in (Feng et al., 2024), difference only rises from the deterministic decision making or stochastic decision making. In GSM8K, we also provided the component analysis by varying ϵ in the range $\{100\%, 75\%, 50\%, 25\%, 0\%\}$, along with the results ranged from 53.2, 52.9, 53.4, 51.8, 50.1, accordingly. The performance drastically drops when the ratio less than 50%.

D. LLM-based Reasoning Experiments Beyond Mathematics

We offer evaluation based on ProofWriter (Tafjord et al., 2020) for deductive logical reasoning, and Chess Endgame (Abdulhai et al., 2023) for long-term multi-turn decision making. For ProofWriter, we follow (Pan et al.) to generate the test set, then the rest are merged to 41,433 training instances. All training and test instances employed the prompt template in (Pan et al.) that initiated the start of CoT, then we employ LLAMA2-7B as the base model and all fine-tuning methods only run for a single epoch. For Chess Endgame, we follow the experimental setup in (Feng et al., 2023). With regards to each prompt-response pair $(\boldsymbol{x}, \boldsymbol{y}^{(w)})$, in ProofWriter and Chess Endgame, we find the dispreferred response $\boldsymbol{y}^{(l)}$ using the same strategy in our mathematical reasoning tasks. We ensure the evaluation in the fair comparison with the CoT and LLM-based tree-search baselines : CoT-greedy, BFS-V, MCTS- tr, MCTS-rollout, CoT-SC-MAJ, CoT-SC-ORM, BFS-V-ORM, MCTS-ORM, whose implementations are consistent in the paper.

For simplicity, we skip the average token number metric to highlight Acc in ProofWriter and Win rate in Chess Endgame. While their results remain based on Path@1 to promise the computation efficiency, and Equal-Token to encourage the comparison in the similar scale of computation consumption cross baselines. In the table, we found that CoT variants almost fail in ProofWriter due to their performances close to random guess (33.33%). MCTS variants obtain significantly better results yet basically under-perform ITS variants with substantial gap in ACC, probably due to the cold-start effect in MCTS learned with one epoch. As for Chess Endgame, ITS variants almost solve the problems with Win rates 99.83% in Path@1 and 98.57% in Equal-Token. It proves ITS also competitive in long-horizon reasoning.