Spectral Policy Optimization: Coloring your Incorrect Reasoning in GRPO

Peter Chen [1] **Xiaopeng Li** [2] **Ziniu Li** [3] **Xi Chen** [4] **Tianyi Lin** [2]

Abstract

Reinforcement learning (RL) has demonstrated significant success in enhancing reasoning capabilities in large language models (LLMs). One of the most widely used RL methods is Group Relative Policy Optimization (GRPO) (Shao et al., 2024), known for its memory efficiency and success in training DeepSeek-R1 (Guo et al., 2025a). However, GRPO stalls when all sampled responses in a group are incorrect – referred to as an *all-negative-sample* group – as it fails to update the policy, hindering learning progress. The contributions of this paper are two-fold. First, we propose a simple yet effective framework that introduces response diversity within all-negative-sample groups in GRPO using AI feedback. We also provide a theoretical analysis, via a stylized model, showing how this diversification improves learning dynamics. Second, we empirically validate our approach, showing the improved performance across various model sizes (7B, 14B, 32B) in both offline and online learning settings with 9 benchmarks, including base and distilled variants. Our findings highlight that learning from all-negative-sample groups is not only feasible but beneficial, advancing recent insights from Xiong et al. (2025).

1. Introduction

The rapid rise of OpenAI-o1 (Jaech et al., 2024), DeepSeek-R1 (Guo et al., 2025a) and Kimi-1.5 (Team et al., 2025) has highlighted the emergence of *large AI reasoning models*. In contrast to instruction-tuned models (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023; Achiam et al., 2023) that generate quick responses based on a statistical inference of what the next token should be, these reasoning models take time to break a complex prompt (e.g., a mathematical problem) down into individual steps and work through chain of thought (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024; Xiang et al., 2025) to offer slower yet more accurate responses. As a consequence, reasoning models are more human-like and can perform complex tasks (Yang et al., 2018; Shi et al., 2024; Jain et al., 2025). As generative AI applications have evolved beyond basic conversational interfaces, these reasoning models are expected to grow in power and adoption, thereby positioning them as a key development to monitor in practice[1].

At the heart of the revolution is the post-training with outcome and verifiable rewards (Cobbe et al., 2021; Uesato et al., 2022; Zelikman et al., 2022; Singh et al., 2023; Hosseini et al., 2024; Lightman et al., 2024; Wang et al., 2024; Setlur et al., 2025; Zhang et al., 2025b), and reinforcement learning (RL) methods (Schulman et al., 2015; 2017; Li et al., 2024b; Ahmadian et al., 2024; Shao et al., 2024; Xiong et al., 2025) being simple, intuitive and practical. A prominent RL method is proximal policy optimization (PPO) (Schulman et al., 2017), which requires a critic (or value) model for estimating the advantage. This model is important for general RL tasks but can be unnecessary for RL in large language models (LLMs) due to their deterministic transition nature (Li et al., 2024b). This insight has motivated the development of group relative policy optimization (GRPO) (Shao et al., 2024) and its variants (Yu et al., 2025b; Liu et al., 2025b; Chu et al., 2025; Zhang et al., 2025a), which directly estimate the advantage in a group-relative manner.

However, all these methods encounter a significant limitation when all sampled responses in a group are incorrect – referred to as an *all-negative-sample* group – as this prevents the policy from receiving any learning signal, stalling progress (Xiong et al., 2025). Specifically, given a prompt

[1]Department of Mathematics, Columbia University, NY, USA [2]Department of IEOR, Columbia University, NY, USA [3]School of Data Science, Chinese University of Hong Kong, Shenzhen, China [4]Stern School of Business, New York University, NY, USA. Correspondence to: Tianyi Lin <tl3335@columbia.edu>, Xi Chen <xc13@stern.nyu.edu>.

42^{nd} International Conference on Machine Learning, The second AI for MATH Workshop at the 42nd International Conference on Machine Learning Vancouver, Canada. 2025. Copyright 2025 by the author(s).

[1]https://platform.openai.com/docs/guides/reasoning-best-practices

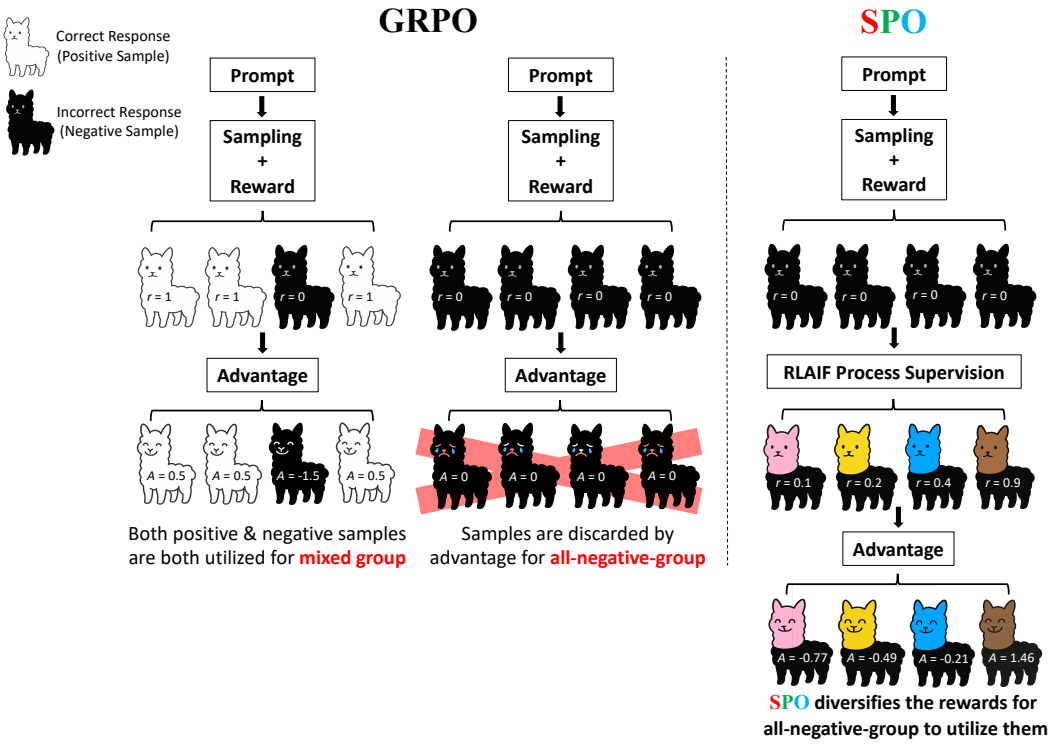

Figure 1: Main pipeline for Spectral Policy Optimization. On each llama, r indicates the reward of the sampled response and A indicates response's advantage through group relative computation.

$\mathbf{x}$, GRPO samples a group of responses $\{\mathbf{y}_i\}_{i=1}^{G}$ from the old policy π_{old} and assigns *binary* rewards $\{r_i\}_{i=1}^{G}$ using a rule-based reward model, where $r_i = 1$ if $\mathbf{y}_i$ is correct and 0 otherwise. GRPO then computes the advantage by normalizing r_i using the group's mean and standard deviation. In an all-negative-sample group with $r_i = 0$ for all i, the advantage for each response is 0, demonstrating no learning signal. Such all-negative-sample groups are common during the early phases of GRPO training, when the model's reasoning abilities remain weak[2]. This limitation underscores a gap between artificial and human intelligence: humans effectively learn from mistakes, which act as essential learning signals during cognitive development (Chialvo & Bak, 1999). In mathematical reasoning, for example, the all-negative-sample groups encourage a child to revise or refine their rules and helps enhance the reasoning capabilities.

Recent work (Xiong et al., 2025) has suggested that the utility of negative samples in RL-based large reasoning model training can be more nuanced than previously assumed. Rather than replying on raw negative feedback (i.e., $r_i = 0$ if $\mathbf{y}_i$ is incorrect), they also advocate for more principled mechanisms to differentiate negative samples. A common approach is using process reward models (PRMs) (Lightman et al., 2024; Wang et al., 2024; Setlur et al., 2025; Zhang et al., 2025b) which are trained to assign intermediate rewards to reasoning steps. PRMs can provide fine-grained supervision to incorporate the sample quality, but they are vulnerable to reward hacking (Skalse et al., 2022) and fail to capture logical dependencies in multi-step reasoning tasks.

Inspired by PRMs, we show that all-negative-sample groups in GRPO can be effectively leveraged by process supervision using AI feedback (Bai et al., 2022; Lee et al., 2024). Unlike traditional PRMs, our approach employs advanced reasoning capabilities of LLMs (e.g., o4-mini[3] and Claude 3.7[4]), enabling a holistic evaluation of multi-step reasoning chains to color negative samples from "black-and-white" outcome reward to diversified spectral reward. For example, a negative sample consists of 5 reasoning steps: $(a_1, a_2, a_3, a_4, a_5)$. Our approach checks each step sequentially to identify the first mistake. If the first mistake is found in a_3, then a_1 and a_2 are correct and the proportion of correct reasoning steps in this negative sample is $\frac{2}{5}$. Then, we use this score to output a reward in $[0, 1)$ (see Equation (2)). Notably, our method avoids the memory overhead of traditional PRM architectures and removes the need for expensive step-level human annotations, simplifying and

[2]To reduce computational cost, it is also common to employ a small group sampling and with a small number of token generation length, which further increases the likelihood of all-negative-sample groups

[3]https://platform.openai.com/docs/models/o4-mini

[4]https://www.anthropic.com/claude/sonnet

accelerating the training pipeline.

Contribution. In this paper, we propose a simple and efficient framework that introduces response diversity within all-negative-sample groups in GRPO using AI feedback. It is designed to effectively utilize all-negative-sample groups, offering a more robust solution to reasoning. Our contributions can be summarized as follows:

1. We identify that the difficulty of GRPO in enhancing reasoning capabilities is exacerbated by the ineffective handling of all-negative-sample groups in GRPO. We propose to mitigate this issue by proposing a *Spectral Policy Optimization* (SPO) framework that leverages AI feedback to *color* negative samples in GRPO from "black-and-white" outcome reward to diversified "spectral" reward. We also provide a theoretical analysis, via a stylized model, explaining why SPO can potentially outperform GRPO.

2. We conduct the experiments demonstrating the effectiveness of our approach in improving LLM reasoning. Evaluations are undertaken across various model sizes (7B, 14B, 32B) in both offline and online learning settings with 10 state-of-the-art benchmarks, including base and distilled variants. Experimental results validate our approach's effectiveness, which addresses limitations in current GRPO pipelines.

Related works. Our work mainly relates to the literature on reinforcement learning from AI feedback (RLAIF) and reinforcement learning for reasoning. Due to space constraints, we defer discussion of additional topics to Appendix B. Prior to our study, the framework of reinforcement learning from human feedback (RLHF) uses human-preference-aligned reward models to evaluate response quality (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022). A key barrier to scale RLHF is the need for high-quality human labels. Previous studies (Gilardi et al., 2023; Ding et al., 2023) have shown that modern LLMs exhibit strong alignment with human judgments, suggesting that AI-generated labels can serve as a viable alternative. In this context, (Bai et al., 2022) was the first to explore RLAIF, jointly optimizing helpfulness and harmlessness using both human and AI-generated labels, and (Roit et al., 2023; Kwon et al., 2023; Lee et al., 2024) showed that LLMs can produce informative reward signals for RL post-training. Our work leverages AI feedback to introduce response diversity within all-negative-sample groups by assigning intermediate binary rewards to reasoning steps. Indeed, one identifies the proportion of correct steps in the reasoning trajectory and use it to compute a reward $r_i \in [0, 1)$. Our approach is more robust for complex reasoning tasks, as advanced LLMs are capable of reliably detecting the mistake in individual reasoning step.

Reinforcement learning has gained prominence as an effective method for improving the reasoning capabilities of LLMs, especially in mathematics and programming. The recent findings (Xiong et al., 2025) have shown that the REINFORCE-type methods (including GRPO (Shao et al., 2024)) can not effectively learn from all-negative-sample groups. Our work alleviates this issue by leveraging AI feedback to differentiate negative samples. We also provide a theoretical analysis through a stylized model, explaining why such diversification improves GRPO's learning dynamics.

2. Preliminaries and Technical Background

We provide an overview of the setup of reasoning in LLMs, the definitions for outcome reward model (ORM), and the scheme of GRPO.

Reasoning and reward models. Modern LLMs are designed based on the Transformer architecture (Vaswani et al., 2017) and follow user prompts $\mathbf{x}$ to generate a response $\mathbf{y} = (a_1, \ldots, a_H)$, i.e., a sequence of tokens with $a_h \in \mathcal{V}^\star$, where $\mathcal{V}$ is a vocabulary of tokens and $\mathcal{V}^\star$ is the set of all possible token sequences that can be formed using elements from $\mathcal{V}$. We consider an LLM as a policy $\pi_\theta(\mathbf{y}|\mathbf{x})$ which corresponds to probabilities to $\mathbf{y}$ given $\mathbf{x}$. For assigning probabilities to each steps of $\mathbf{y}$, the policy π_θ operates in an auto-regressive manner as follows,

$$\pi_\theta(\mathbf{y}|\mathbf{x}) = \Pi_{h=1}^{H} \pi_\theta(a_h|\mathbf{x}, a_1, a_2, \ldots, a_{h-1}).$$

where θ stands for the model's parameter. For the prompt $\mathbf{x}$ with a ground-truth response $\mathbf{y}_\mathbf{x}^\star$, we evaluate π_θ by running a regular expression match on the final answer: $r(\mathbf{x}, \mathbf{y}) = 1$ if $\mathbf{y}$ matches $\mathbf{y}_\mathbf{x}^\star$ on the *final answer* and $r(\mathbf{x}, \mathbf{y}) = 0$ otherwise (Hendrycks et al., 2021). We consider the reasoning task that depends on a dataset $\mathcal{D}$ containing samples $(\mathbf{x}, \mathbf{y}_\mathbf{x}^\star)$, where $\mathbf{x}$ is a problem and $\mathbf{y}_\mathbf{x}^\star$ refers to a ground-truth answer.

GRPO. The policy gradient methods (Williams, 1992; Sutton & Barto, 1998) are introduced to maximize the objective function $J(\theta) = \mathbb{E}_{\mathbf{x}\sim\rho, \mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x}, \mathbf{y})]$ where ρ is a prompt distribution and π_θ is an LLM. Indeed, we perform $\theta \leftarrow \theta + \eta \nabla_\theta J(\theta)$ at each iteration. In practice, we collect the sampled trajectories from $\pi_{\theta_{\text{old}}}$ which is different from π_θ and hope to use them to compute the policy gradient estimator. This motivates us to adopt the importance sampling to rewrite the objective function as follows,

$$J(\theta) = \mathbb{E}_{\mathbf{x}\sim\rho, \mathbf{y}\sim\pi_{\theta_{\text{old}}}(\cdot|\mathbf{x})} \left[\frac{\pi_\theta(\mathbf{y}|\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}|\mathbf{x})} r(\mathbf{x}, \mathbf{y}) \right].$$

However, importance sampling can lead to high variance when π_θ deviates from $\pi_{\theta_{\text{old}}}$. To stabilize the training, one can maximize the following clipped surrogate function as

follows,

$$J(\theta) = \mathbb{E}_{\mathbf{x}\sim\rho,\mathbf{y}\sim\pi_{\theta_{\text{old}}}(\cdot|\mathbf{x})}\Big[\min\Big\{\frac{\pi_\theta(\mathbf{y}|\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}|\mathbf{x})}r(\mathbf{x},\mathbf{y}), \texttt{clip}\Big(\frac{\pi_\theta(\mathbf{y}|\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}|\mathbf{x})},\,1-\epsilon,\,1+\epsilon\Big)r(\mathbf{x},\mathbf{y})\Big\}\Big].$$

In this context, GRPO and its variants (Yu et al., 2025b; Liu et al., 2025b; Chu et al., 2025; Zhang et al., 2025a) operate within the aforementioned framework and estimate a policy gradient from groups of samples. For each prompt $\mathbf{x}$, GRPO samples a group of responses $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$ and maximizes the objective function in the following form of

$$J(\theta) = \frac{1}{G}\sum_{i=1}^{G}\Big[\min\Big\{\frac{\pi_\theta(\mathbf{y}_i|\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}_i|\mathbf{x})}A_i, \texttt{clip}\Big(\frac{\pi_\theta(\mathbf{y}_i|\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}_i|\mathbf{x})},\,1-\epsilon,\,1+\epsilon\Big)A_i\Big\}\Big].$$

where $\epsilon \in (0,1)$ is a hyper-parameter and the advantage A_i is computed using a group of rewards corresponding to $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_G\}$ within each group as follows,

$$A_i = \frac{r(\mathbf{x},\mathbf{y}_i) - \texttt{mean}(\{r(\mathbf{x},\mathbf{y}_1),\ldots,r(\mathbf{x},\mathbf{y}_G)\})}{\texttt{std}(\{r(\mathbf{x},\mathbf{y}_1),\ldots,r(\mathbf{x},\mathbf{y}_G)\})}, \tag{1}$$

where $r(\mathbf{x},\mathbf{y}_i) = 1$ if $\mathbf{y}_i$ matches the ground-truth final answer and $r(\mathbf{x},\mathbf{y}) = 0$ otherwise.

Remark 2.1. When the rewards are identical across all of samples within one group, the advantage $A_i = 0$, resulting in no parameter updates. This makes sense for all-positive-sample group where $r(\mathbf{x},\mathbf{y}_i) = 1$ for all i but turns out to be a **significant** limitation for all-negative-sample group where $r(\mathbf{x},\mathbf{y}_i) = 0$ for all i, demonstrating that GRPO can **not** effectively learn from mistakes.

3. Main Results

We introduce SPO that leverages AI feedback to better utilize all-negative-sample groups in GRPO. We also provide a theoretical analysis via a stylized model, explaining why SPO can potentially lead to provably faster learning than GRPO.

3.1. Process supervision via RLAIF

An incorrect final answer does not necessarily indicate that the entire reasoning process is invalid. For instance, a model may follow a logically sound sequence of steps but commit a minor error—such as an arithmetic mistake—that leads to an incorrect conclusion. It would be inappropriate to treat such responses equivalently to those that exhibit fundamentally flawed or incoherent reasoning. To address this issue, we propose a more principled reward mechanism for negative samples, wherein AI feedback is employed to differentiate between structurally sound but partially incorrect reasoning and wholly erroneous responses. This refinement remains applicable even under constraints such as reduced output length, where the model may be unable to complete the full answer but still demonstrates a valid reasoning trajectory.

Reward framework. Concretely, we utilize an advanced reasoning model to evaluate each agent response on a sentence-by-sentence basis, identifying the first sentence that contains a substantive error—be it a computational mistake or a reasoning fallacy—that causes the trajectory to deviate from the correct solution path. To quantify this, we define a *Reasoning Trajectory Score* for wrong response $\mathbf{y}_{\text{wrong}}$, denoted by $\mathcal{RTS}(\mathbf{y}_{\text{wrong}}) \in [0,1]$, in which we prompt a stronger model to reason the logical correctness of each sentence in the sampled response one by one (see Appendix D) and identify the first clear mistake that causes the solution to deviate from the correct answer. We consider all portions of the response before this mistake as the correct reasoning segment, and $\mathcal{RTS}(\mathbf{y}_{\text{wrong}})$ is calculated as the ratio of the length of this correct reasoning segment to the total length of the response. Suppose a model generates $\mathbf{y}_{\text{wrong}}$ that consists of 5-step reasoning trajectory $(a_1, a_2, a_3, a_4, a_5)$. RLAIF checks each step sequentially to find the first mistake. If the first mistake is in a_4, then $\mathcal{RTS}(\mathbf{y}_{\text{wrong}}) = \frac{3}{5}$, showing that three steps of reasoning are correct in this trajectory.

RLAIF prompting details. We empirically observe that different prompting strategies can cause even strong judge models to produce varying judgments. To ensure robustness and make the reward signal as deterministic as possible, we recommend the following prompting method. First, in addition to the sampled response for RLAIF evaluation, a reference reasoning solution should also be provided. This typically requires a training dataset suitable for supervised fine-tuning—namely, one that contains both correct answers and correct reasoning trajectories. Providing such a reference enables the judge model to better understand the intended solution path and more accurately identify errors in the sampled response for evaluation. Second, instead of prompting the judge model to assess the answer holistically, we suggest prompting it to evaluate the solution step by step, sentence by sentence. For each sentence, the judge should explain why it is correct or, if incorrect, why it constitutes an error. After identifying the first clear mistake, the judge should continue reading the remaining sentences and reason through how this error leads to the final incorrect conclusion.

Based on this process supervision, we introduce a new outcome reward function:

$$r_{\texttt{AIF}}(\mathbf{y}) = \begin{cases} 1, & \text{if } \mathbf{y} \text{ is correct,} \\ \frac{1}{1+\exp(\beta(\mathcal{RTS}(\mathbf{y})-\gamma))}, & \text{otherwise.} \end{cases} \tag{2}$$

where $\gamma > 0$ and $\beta > 0$ are two constant parameters to decide scale threshold and scale intensity, respectively. This design ensures that the model receives a more informative gradient signal during training, thereby encouraging refinement of partially correct reasoning rather than indiscriminate penalization of all incorrect outputs. This specification of $r_{\texttt{AIF}}$ can be directly incorporated into the advantage calculation in Equation (1). As a consequence, we refer to SPO as GRPO using Equation (2).

Remark 3.1. The reward function $r_{\texttt{AIF}}$ in Equation (2) yields the same optimal policy as the original binary outcome reward r, while providing more nuanced training signals.

3.2. Theory on a stylized model

We present a stylized analysis to explain why SPO can potentially lead to provably faster learning than GRPO. We consider the setting introduced in Section 2, and focus on a simplified case where the number of reasoning steps is $H = 2$, and each step has only two possible choices, i.e., $a_h \in \{1, 2\}$ for $h = 1, 2$. This minimal configuration follows precedent in the literature (Dayan, 1991; Li et al., 2024b), where similar simplified examples have been used to validate theoretical insights.

For simplicity, we assume a unique ground-truth response $\mathbf{y}^\star_{\mathbf{x}} = (2, 2)$ for the prompt $\mathbf{x}$. Our goal is to iteratively learn the optimal policy using samples generated by the current policy π_θ. In our stylized model, we restrict the sample space to $\mathbf{y} \in \{(1, 1), (2, 1), (2, 2)\}$, excluding the sample $(1, 2)$. This is based on an observation from human reasoning: a correct reasoning step is unlikely to, and should not, follow an incorrect precursor.

Reward mechanisms. To illustrate the effect of SPO, we compare the learning dynamics of SPO and GRPO this stylized setting. In particular, GRPO uses classical ORM to assign $r((2, 2)) = 1$ and $r((2, 1)) = r((1, 1)) = 0$. Thus, only the complete selection of the "good" action 2 in both steps yields the optimal reward. In contrast, SPO assigns $r_{\texttt{AIF}}((2, 2)) = 1$, $r_{\texttt{AIF}}((2, 1)) = \frac{1}{2}$, and $r_{\texttt{AIF}}((1, 1)) = 0$. The key difference is that partial progress – selecting the "good" action 2 even in just the first step – receives no credit in GRPO yet proportional credit in SPO. Note that $\frac{1}{2}$ is only chosen to reflect the qualitative behavior of the reward mechanism for simplicity, while the exact values used in our experiments are computed by Equation (2).

In our analysis, we consider the population-level learning dynamics of GRPO with the group size $G = 2$ and no clipping or importance sampling for simplicity. We denote $p^{(k)}_{\texttt{GRPO}}$ as the probability of selecting a "good" action in the first step at iteration k under GRPO (with classical ORM), and $q^{(k)}_{\texttt{GRPO}}$ as the probability of selecting a "good" action in the second step, given that a "good" action is chosen in the first step. We also denote $p^{(k)}_{\texttt{SPO}}$ and $q^{(k)}_{\texttt{SPO}}$ as the corresponding probabilities under SPO. The following theorem summarizes our main theoretical results.

Theorem 3.2. *Suppose that we initialize* $p^{(0)}_{GRPO} = q^{(0)}_{GRPO} = p^{(0)}_{SPO} = q^{(0)}_{SPO} = \frac{1}{2}$ *and use the stepsize* $\eta = 1$[5] *for GRPO (with ORM) and SPO. Then, we have (i) both SPO and GRPO achieve successful learning:* $p^{(k)}_{GRPO}, q^{(k)}_{GRPO}, p^{(k)}_{SPO}, q^{(k)}_{SPO} \to 1$ *as* $k \to +\infty$*; (ii) SPO outperforms GRPO in learning the "good" action in the first step consistently:* $p^{(k)}_{SPO} > p^{(k)}_{GRPO}$ *for all* $k \geq 1$*; (iii) SPO outperforms GRPO in learning the optimal policy eventually:* $p^{(k)}_{SPO} q^{(k)}_{SPO} > p^{(k)}_{GRPO} q^{(k)}_{GRPO}$ *for all* k *satisfying* $q^{(k)}_{SPO} > \frac{31}{32}$.

Remark 3.3 (Theoretical insights). The first result confirms that SPO successfully learns the optimal policy. The second and third results show that SPO offers both rapid acquisition of partially correct reasoning steps and retention of partial reasoning capabilities despite incorrect final answers.

Remark 3.4 (Technical novelty). To our knowledge, Theorem 3.2 is among the very first results for GRPO-type methods with multiple samples and steps in the context of LLM reasoning. Notably, it provides a **per-iteration** comparison of learning under different reward mechanisms, an aspect rarely explored in prior works. However, the provably improved learning of the optimal policy is only local, despite consistent numerical support in a global sense; see Figure 2 in Appendix C.4.

4. Experiment

We demonstrate the benefits of coloring negative samples in GRPO through experiments conducted in both offline and online settings. In the offline setting, both SPO and GRPO update the policy model using a fixed dataset. In the online setting, the policy is updated using data generated by the policy in the previous iteration. Offline RL requires fewer computational resources and generally results in more stable training, whereas online RL offers greater flexibility and learning capacity. The latter approach has been widely adopted in large AI reasoning models, including DeepSeek-R1 (Guo et al., 2025a).

4.1. Experiment setups

Offline models. For baseline models, we first focus on stronger ones without supervised fine-tuning (SFT), specifically Qwen2.5-14B-Instruct and Qwen2.5-32B-Instruct.

Offline benchmarks. Recent studies have shown that a small set of carefully curated prompts can significantly enhance the reasoning capabilities of LLMs. In our experiments, we use the `GAIR/LIMO` dataset (Ye et al., 2025),

[5] We use the unit stepsize for simplicity. Our results are valid for any sufficiently small step size.

which has demonstrated strong potential for improving the reasoning performance of large-scale (32B) models in offline SFT settings. For evaluation, we present offline RL results across four standard mathematical reasoning benchmarks: the American Invitational Mathematics Examination (AIME24), the American Mathematics Competitions (AMC23), MATH500 (Hendrycks et al., 2021), and OlympiadBench (He et al., 2024). Our goal is to highlight the rich information contained in all-negative-sample-groups – specifically, that learning exclusively from them have already driven model improvement. For benchmarks with fewer than 100 questions (AMC23, AIME24), we report the results in terms of both pass@16 and avg@16 using a decoding temperature of 0.6 and Top_P=0.95. We report only avg@16 if pass@16 does not show significant gains. For benchmarks with more than 100 questions, we report the results in terms of pass@1 using greedy decoding. Across all benchmarks, the maximum output length is fixed at 32K tokens.

Offline training. We adopt the standard GRPO response sampling and generation framework, but perform all response generation and model updates under an offline RL framework (Peters & Schaal, 2007). Specifically, we update the model with the advantage estimated with offline dataset (see e.g., (Peng et al., 2019; Li et al., 2024b)). For each prompt, we sample 6 responses within one group and identify all-negative-sample groups where all responses yield incorrect answers. We apply RLAIF to assign specific rewards to differentiate negative samples within one group (see Equation (2)), which are then used for offline RL training. The model is trained for 3 epochs with a learning rate of 2×10^{-6}. As a contrastive baseline, we also perform offline RL using only positive responses that output correct answers, applying an ORM reward to guide learning. This parallel setup enables a direct comparison of the effectiveness of learning from purely negative versus purely positive reasoning processes.

Online models. Proceeding to the online setting, we expand the range of baseline models to examine whether negative samples can consistently improve the performance of GRPO. Our baseline models include **(i)** Qwen-Instruct models: Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct, and Qwen2.5-32B-Instruct; and **(ii)** R1-Distillation models: DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B. Online GRPO training is implemented using the verl framework (Sheng et al., 2025). For RLAIF, we use o4-mini from OpenAI for Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct, Qwen2.5-32B-Instruct, and DeepSeek-R1-Distill-Qwen-7B. We also try Claude3.7 from Anthropic as the reward model for DeepSeek-R1-Distill-Llama-8B.

Online benchmarks. Compared to offline RL, online RL typically leads to greater enhancement in model's reasoning abilities. Since we have included strong distillation models as baselines, some benchmarks used in the offline evaluation are becoming saturated. For example, baseline models can achieve over 90% accuracy even before applying GRPO. To address this, we expand our evaluation benchmarks beyond AMC23, AIME24, MATH500, and OlympiadBench, by including additional benchmarks: AIME25, GradeSchool (Ye et al., 2025), CHMath24, Kaoyan, and Gaokao. Specifically, CHMath24 is from the 2024 Chinese High School Mathematics League Competition, Gaokao is from China's 2024 National College Entrance Examination, Kaoyan is from the Chinese Graduate School Entrance Examinations, and GradeSchool is for elementary-level mathematical reasoning. Among these benchmarks, CHMath24 and Gaokao each contain fewer than 100 questions; similar to AMC23 and AIME24, we apply multi-sample decoding for evaluation on these datasets.

Table 1: Hyperparameter configurations and setups for online GRPO training.

Model	Len	Group Size	Epochs	Device
DeepSeek-R1-Distill-Qwen-7B	8192	8	12	8×H100
DeepSeek-R1-Distill-Llama-8B	8192	8	12	8×H100
Qwen2.5-7B-Instruct	8192	8	25	8×H100
Qwen2.5-14B-Instruct	8192	8	12	16×H100
Qwen2.5-32B-Instruct	4096	4	12	8×H200

Online training. For GRPO training, we adopt the AIME collections from 1997 to 2023, as used in DeepScaler (Luo et al., 2025b), providing approximately 1,000 questions per epoch. Notably, all of questions in our training set is written in English, whereas the evaluation benchmarks include multilingual questions. Interestingly, we observe that the negative samples learned during training generalize well to out-of-domain mathematical reasoning tasks. We encourage future work to adopt the same evaluation setup to ensure consistency and comparability.

For SPO training, we use the same configuration as in GRPO training. Note that process supervision from RLAIT is applied only to all-negative-sample groups during the first 3 epochs. This is because we believe that this duration is sufficient for the model to internalize the corrective signals – continued training beyond this point on unresolved examples likely reflects limitations in model capacity rather than learnability. These seemingly too difficult examples should be excluded from subsequent training iterations to avoid introducing noise or destabilizing the optimization process. Our results show that leveraging RLAIF to differentiate examples within all-negative-sample group in GRPO can improve the performance of LLMs, even under reduced context length and smaller group sizes.

Due to memory constraints, we reduce the group size to 4

Table 2: Evaluation results on offline RL training. Baseline performance is shown alongside RL training results using only negative or only positive samples across benchmarks and the LIMO training set.

	AMC23 avg@16	AIME24 avg@16	MATH500 pass@1	Olympiads pass@1	LIMO pass@1
Qwen2.5-14B-Instruct					
Baseline	58.59	14.58	**80.40**	41.78	31.70
w/ Neg. Samples only	**61.88**	**15.21**	**80.40**	**42.37**	30.11
w/ Pos. Samples only	61.72	14.58	79.80	42.07	**38.68**
Qwen2.5-32B-Instruct					
Baseline	64.22	17.08	**83.60**	45.93	34.64
w/ Neg. Samples only	**69.53**	**20.42**	83.00	46.37	36.47
w/ Pos. Samples only	66.87	18.75	**83.60**	**47.41**	**41.86**

for larger models or those with longer reasoning chains. In particular, for online GRPO training with the 32B model, we limit the output length to 4K tokens to fit within the memory capacity of 8 NVIDIA H200 GPUs (143 GB each). For all other models, we use 8 or 16 NVIDIA H100 HBM3 GPUs (81 GB memory each). We summarize the online GRPO training hyperparameter configurations and setups in Table 1.

4.2. Experimental results

Offline training. We conduct offline RL training to demonstrate that SPO using only all-negative-sample groups that are discarded by GRPO can improve the reasoning abilities of LLMs. Additionally, we include positive-only offline RL training as a comparison against the negative-sample setup. The results are shown in Table 2.

We make the following key observations: negative samples are as effective as positive samples on the training dataset itself. However, we find that negative samples can actually improve performance across a broader set of evaluation benchmarks, in some cases even outperforming models trained with positive samples. It's surprising to highlight that for 14B model experiment, even negative samples reduces the performance on training dataset (from 0.3170 to 0.3011), it brings a significant improvement across the other benchmarks comparing to the positive samples. Without doubt, this highlights the effectiveness of the negative samples, which should not be easily discarded in online GRPO training. Still, both negative and positive samples can also lead to performance degradation. This effect appears to be more dependent on the specific characteristics of the training dataset. We will elaborate it in the online learning results discussion.

Online training. Compared to the offline results, we include a broader range of model types (e.g., distilled models and base models without SFT), a wider selection of model families (Qwen2.5 and Llama), and more diverse benchmarks to better reflect the performance of stronger distilled models after training. We present the online learning results in Table 3.

In addition to the best results after tuning shown in Table 3, we also present additional experimental results in Appendix D, including evaluations on an even weaker model without distillation or SFT, as well as the robustness of SPO across multiple independent runs.

A key insight from our experimental results is that stronger models tend to produce higher-quality negative samples that can significantly benefit model learning. In brief, we argue that as model quality improves, so does the quality and informativeness of its negative samples. First, as previously discussed, due to memory constraints, larger models cannot generate long outputs (e.g., 16K or 32K tokens) needed to fully complete complex chain-of-thought reasoning required for certain difficult questions. As a result, negative samples can be broadly categorized into two types: **(i)** those that follow a correct reasoning process but fail to reach the final answer due to output length limitations, and **(ii)** those that contain incorrect reasoning steps or logical errors that lead to a wrong answer.

The first type offers meaningful reasoning trajectories that the model can still learn from—precisely the motivation behind our RLAIF framework with process supervision. In contrast, such samples are discarded in GRPO. Moreover, when a group of samples all yield incorrect answers, this often indicates the question itself is *genuinely challenging*. These cases should not be ignored, as they contain valuable signal, even if only partially correct. Learning from such difficult questions could ultimately be more beneficial than from questions the model can already solve.

It's also worth noting that for stronger distilled models, the average response length during training is approximately 6K tokens, whereas for weaker base models without distillation and SFT, the average is only around 1K tokens. As a result, the first type of negative samples—those containing correct reasoning but truncated due to output length limitations—occurs much more frequently in stronger models than in weaker ones. In a similar way, for the second type of negative samples—those with incorrect reasoning steps, stronger models also tend to produce more informative responses for the AI model to judge. Recall from Eq. (2) that we use logistic scaling to intentionally penalize responses that make mistakes early in the reasoning process. Specifically, responses receiving very low raw rewards (e.g., 0.1 or 0.2) are effectively treated similarly after scaling. In contrast, responses with more correct intermediate steps are amplified, allowing the model to learn from partially correct reasoning paths. This design makes the reward signal for type (ii) cases more stable and reduces sensitivity to minor inaccuracies in AI feedback.

Table 3: Benchmark performance of online learning between GRPO and SPO. Specifically, BASELINE is referring to the performance of the original model without RL finetuning. Overall is average performance across all the benchmarks. Note that the training dataset is `AIME1997-2023`.

	Kaoyan pass@1	GradeMath pass@1	MATH500 pass@1	Olympiads pass@1	CHMath24 avg@16	AIME25 avg@16	AIME24 avg@16	Gaokao avg@16	AMC23 avg@16	**Overall** avg
DeepSeek-R1-Distill-Qwen-7B										
BASELINE	50.25	41.43	87.00	49.93	73.75	**40.62**	52.92	80.22	89.53	62.85
GRPO	55.78	43.33	89.40	**56.00**	71.04	36.68	52.08	80.30	88.91	63.72
SPO	**57.79**	**46.19**	**90.80**	54.67	**75.00**	38.33	**54.58**	**81.33**	**90.00**	**65.41**
DeepSeek-R1-Distill-Llama-8B										
BASELINE	29.15	23.81	77.40	41.48	**61.46**	27.92	**42.29**	**72.78**	87.97	51.58
GRPO	35.68	28.33	**84.00**	46.32	57.08	**28.33**	42.08	68.99	86.72	53.06
SPO	**39.70**	**29.05**	83.60	**48.44**	58.96	24.58	39.37	71.52	**89.06**	**53.81**
Qwen2.5-14B-Instruct										
BASELINE	37.69	49.52	80.40	41.78	21.88	13.13	**14.58**	**41.14**	58.59	39.85
GRPO	**43.22**	47.14	80.20	43.11	21.88	13.13	13.33	39.16	**59.84**	40.11
SPO	38.69	**53.33**	**81.00**	**44.00**	**22.92**	**16.67**	14.17	39.00	59.22	**41.00**
Qwen2.5-32B-Instruct										
BASELINE	45.73	**53.81**	**83.60**	45.93	26.87	12.29	17.08	44.15	64.22	43.74
GRPO	**48.24**	52.86	83.20	45.93	22.50	12.08	**21.67**	**45.73**	67.34	44.39
SPO	**48.24**	**53.81**	83.00	**46.81**	**29.79**	**14.58**	19.58	45.09	**69.53**	**45.06**

The above insight is further supported by our experimental results, where all four models are trained under the same setup. A representative example is the `Kaoyan` dataset, where DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B exhibit improvements of 7% and 10% over their respective baselines after only 12 epochs of training. In contrast, the performance gains are significantly smaller for the larger Qwen2.5 base models, which possess weaker initial capabilities.

4.3. Discussions: offline versus online

We would like to reiterate the purpose of using both offline RL and online RL. In the offline setup, we use **only** negative samples for training—i.e., the model is explicitly trained to learn from its incorrect or incomplete reasoning trajectories. This setup is designed to directly evaluate whether such negative samples can improve model performance. In the online setup, we simulate the realistic GRPO learning scenario, where training batches consist of a random mix of both positive and negative samples. This allows us to demonstrate not only that negative samples are effective in independent setups, but also that they remain valuable in practical implementations where mixed samples and higher levels of noise are present.

Indeed, mixing both negative and positive samples introduces additional noise into the learning process, compared to simply discarding challenging questions that the model fails to solve. However, we also observe that discarding negative samples does not necessarily lead to a more stable learning environment. In fact, in several cases, the GRPO results degrade below the original baseline, yielding poorer performance when the model is trained solely on problems it can already solve.

This instability can be attributed to several factors, including poor out-of-domain (OOD) generalization, and catastrophic forgetting. Without exposure to challenging or partially correct reasoning trajectories, the model may overfit to a narrow subset of easy examples, losing its ability to generalize to harder or novel tasks. This can result in the model reinforcing shallow heuristics rather than developing robust problem-solving skills. Additionally, the lack of diverse failure cases may lead to catastrophic forgetting, where previously acquired knowledge is overwritten during training, further degrading performance on tasks that were once solvable. Therefore, including negative samples could to some extent alleviate these problems, as SPO perform typically better than GRPO on Chinese OOD math reasoning benchmarks. Still, there remains significant room for future work to explore how to achieve a stable learning environment by incorporating negative samples through further reward diversification mechanisms.

5. Conclusion

We propose a simple yet effective framework that introduces response diversity within all-negative-sample groups using AI feedback and provide a theoretical analysis, via a stylized model, showing how this diversification improves learning dynamics of GRPO. Experimental results show

the improved performance of GRPO across various model sizes (7B, 14B, 32B) in both offline and online learning settings with 10 benchmarks, including base and distilled variants. This demonstrates the importance of learning from all-negative-sample groups, which advances recent insights from Xiong et al. (2025). Future directions include the extension of our theoretical results to a more general model with multiple reasoning steps and applications of our framework to improve other RL methods.

Acknowledgment

We sincerely appreciate Buzz High Performance Computing (https://www.buzzhpc.ai/, info@buzzhpc.ai) for providing computational resources and support for this work.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. GPT-4 technical report. *ArXiv Preprint: 2303.08774*, 2023.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. Optimality and approximation with policy gradient methods in markov decision processes. In *COLT*, pp. 64–66, 2020.

Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., and Hooker, S. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *ACL*, pp. 12248–12267, 2024.

Arora, D. and Zanette, A. Training language models to reason efficiently. *ArXiv Preprint: 2502.04463*, 2025.

Azar, M. G., Guo, Z., Piot, B., Munos, R., Rowland, M., Valko, M., and Calandriello, D. A general theoretical paradigm to understand learning from human preferences. In *AISTATS*, pp. 4447–4455, 2024.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. Constitutional AI: Harmlessness from AI feedback. *ArXiv Preprint: 2212.08073*, 2022.

Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., et al. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI*, pp. 17682–17690, 2024.

Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., et al. Deepseek LLM: Scaling open-source language models with longtermism. *ArXiv Preprint: 2401.02954*, 2024.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *NeurIPS*, pp. 1877–1901, 2020.

Chen, H., He, G., Yuan, L., Cui, G., Su, H., and Zhu, J. Noise contrastive alignment of language models with explicit rewards. In *NeurIPS*, pp. 117784–117812, 2024a.

Chen, H., Zhao, H., Lam, H., Yao, D., and Tang, W. MallowsPO: Fine-tune your LLM with preference dispersions. In *ICLR*, 2025a. URL https://openreview.net/forum?id=d8cnezVcaW.

Chen, P., Chen, X., Yin, W., and Lin, T. ComPO: Preference alignment via comparison oracles. *ArXiv Preprint: 2505.05465*, 2025b.

Chen, Q., Qin, L., Liu, J., Peng, D., Guan, J., Wang, P., Hu, M., Zhou, Y., Gao, T., and Che, W. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *ArXiv Preprint: 2503.09567*, 2025c.

Chen, X., Xu, J., Liang, T., He, Z., Pang, J., Yu, D., Song, L., Liu, Q., Zhou, M., Zhang, Z., et al. Do not think that much for 2+3 = ? on the overthinking of o1-like LLMs. *ArXiv Preprint: 2412.21187*, 2024b.

Cheng, J. and Van Durme, B. Compressed chain of thought: Efficient reasoning through dense representations. *ArXiv Preprint: 2412.13171*, 2024.

Chialvo, D. R. and Bak, P. Learning from mistakes. *Neuroscience*, 90(4):1137–1148, 1999.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *NeurIPS*, pp. 4302–4310, 2017.

Chu, X., Huang, H., Zhang, X., Wei, F., and Wang, Y. GPG: A simple and strong reinforcement learning baseline for model reasoning. *ArXiv Preprint: 2504.02546*, 2025.

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. *ArXiv Preprint: 2110.14168*, 2021.

Dayan, P. Reinforcement comparison. In *Connectionist Models*, pp. 45–51. Elsevier, 1991.

Ding, B., Qin, C., Liu, L., Chia, Y. K., Li, B., Joty, S., and Bing, L. Is GPT-3 a good data annotator? In *ACL*, pp. 11173–11195, 2023.

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., Shum, K., and Zhang, T. RAFT: Reward ranked fine-tuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=m7p5O7zblY.

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. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a13aYUU9eU.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Model alignment as prospect theoretic optimization. In *ICML*, pp. 12634–12651, 2024.

Feng, G., Zhang, B., Gu, Y., Ye, H., He, D., and Wang, L. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *NeurIPS*, pp. 70757–70798, 2023.

Gandhi, K., Lee, D., Grand, G., Liu, M., Cheng, W., Sharma, A., and Goodman, N. D. Stream of search (SOS): Learning to search in language. *ArXiv Preprint: 2404.03683*, 2024.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *ICML*, pp. 10835–10866, 2023.

Gilardi, F., Alizadeh, M., and Kubli, M. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30): e2305016120, 2023.

Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Huang, M., Duan, N., and Chen, W. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *ICLR*, 2024. URL https://openreview.net/forum?id=Ep0TtjVoap.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *ArXiv Preprint: 2501.12948*, 2025a.

Guo, J., Wu, Y., Qiu, J., Huang, K., Juan, X., Yang, L., and Wang, M. Temporal consistency for LLM reasoning process error identification. *ArXiv Preprint: 2503.14495*, 2025b.

Hao, S., Gu, Y., Ma, H., Hong, J., Wang, Z., Wang, D., and Hu, Z. Reasoning with language model is planning with world model. In *EMNLP*, pp. 8154–8173, 2023.

Hao, S., Gu, Y., Luo, H., Liu, T., Shao, X., Wang, X., Xie, S., Ma, H., Samavedhi, A., Gao, Q., Wang, Z., and Hu, Z. LLM reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models. In *COLM*, 2024a. URL https://openreview.net/forum?id=b0y6fbSUG0.

Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., and Tian, Y. Training large language models to reason in a continuous latent space. *ArXiv Preprint: 2412.06769*, 2024b.

Havrilla, A., Du, Y., Raparthy, S. C., Nalmpantis, C., Dwivedi-Yu, J., Zhuravinskyi, M., Hambro, E., Sukhbaatar, S., and Raileanu, R. Teaching large language models to reason with reinforcement learning. *ArXiv Preprint: 2403.04642*, 2024.

He, C., Luo, R., Bai, Y., Hu, S., Thai, Z., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., et al. OlympiadBench: A challenging benchmark for promoting AGI with Olympiad-level bilingual multimodal scientific problems. In *ACL*, pp. 3828–3850, 2024.

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. In *NeurIPS Datasets and Benchmarks Track*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Hong, J., Lee, N., and Thorne, J. ORPO: Monolithic preference optimization without reference model. In *EMNLP*, pp. 11170–11189, 2024.

Hosseini, A., Yuan, X., Malkin, N., Courville, A., Sordoni, A., and Agarwal, R. V-STar: Training verifiers for self-taught reasoners. In *COLM*, 2024. URL https://openreview.net/forum?id=stmqBSW2dV.

Huang, J. and Chang, K. C.-C. Towards reasoning in large language models: A survey. In *ACL*, pp. 1049–1065, 2023.

Huang, K., Guo, J., Li, Z., Ji, X., Ge, J., Li, W., Guo, Y., Cai, T., Yuan, H., Wang, R., Wu, Y., Yin, M., Tang, S., Huang, Y., Jin, C., Chen, X., Zhang, C., and Wang, M. MATH-Perturb: Benchmarking LLMs' math reasoning abilities against hard perturbations. *ArXiv Preprint: 2502.06453*, 2025.

Huang, S., Ma, Z., Du, J., Meng, C., Wang, W., and Lin, Z. Mirror-consistency: Harnessing inconsistency in majority voting. In *EMNLP*, pp. 2408–2420, 2024.

Hwang, H., Kim, D., Kim, S., Ye, S., and Seo, M. Self-explore: Enhancing mathematical reasoning in language models with fine-grained rewards. In *EMNLP*, pp. 1444–1466, 2024.

Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. OpenAI o1 system card. *ArXiv Preprint: 2412.16720*, 2024.

Jain, N., Han, K., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *ICLR*, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., and Sabharwal, A. Decomposed prompting: A modular approach for solving complex tasks. In *ICLR*, 2023. URL https://openreview.net/forum?id=_nGgzQjzaRy.

Kwon, M., Xie, S. M., Bullard, K., and Sadigh, D. Reward design with language models. In *ICLR*, 2023. URL https://openreview.net/forum?id=10uNUgI5Kl.

Lampinen, A. K., Dasgupta, I., Chan, S. C. Y., Sheahan, H. R., Creswell, A., Kumaran, D., McClelland, J. L., and Hill, F. Language models, like humans, show content effects on reasoning tasks. *PNAS Nexus*, 3(7):pgae233, 2024.

LeCun, Y. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Lee, H., Phatale, S., Mansoor, H., Mesnard, T., Ferret, J., Lu, K., Bishop, C., Hall, E., Carbune, V., Rastogi, A., et al. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *ICML*, pp. 26874–26901, 2024.

Lehnert, L., Sukhbaatar, S., Su, D., Zheng, Q., McVay, P., Rabbat, M., and Tian, Y. Beyond $A^\star$: Better planning with transformers via search dynamics bootstrapping. In *COLM*, 2024. URL https://openreview.net/forum?id=SGoVIC0u0f.

Li, Z., Liu, H., Zhou, D., and Ma, T. Chain of thought empowers transformers to solve inherently serial problems. In *ICLR*, 2024a. URL https://openreview.net/forum?id=3EWTEy9MTM.

Li, Z., Xu, T., Zhang, Y., Lin, Z., Yu, Y., Sun, R., and Luo, Z.-Q. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *ICML*, pp. 29128–29163, 2024b.

Li, Z., Chen, C., Xu, T., Qin, Z., Xiao, J., Luo, Z.-Q., and Sun, R. Preserving diversity in supervised fine-tuning of large language models. In *ICLR*, 2025. URL https://openreview.net/forum?id=NQEe7B7bSw.

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. In *ICLR*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Liu, T., Zhao, Y., Joshi, R., Khalman, M., Saleh, M., Liu, P. J., and Liu, J. Statistical rejection sampling improves preference optimization. In *ICLR*, 2024a. URL https://openreview.net/forum?id=xbjSwwrQOe.

Liu, T., Qin, Z., Wu, J., Shen, J., Khalman, M., Joshi, R., Zhao, Y., Saleh, M., Baumgartner, S., Liu, J., et al. LiPO: Listwise preference optimization through learning-to-rank. In *NAACL*, pp. To appear, 2025a.

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. In *NeurIPS*, pp. 138663–138697, 2024b.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding R1-zero-like training: A critical perspective. *ArXiv Preprint: 2503.20783*, 2025b.

Luo, H., Shen, L., He, H., Wang, Y., Liu, S., Li, W., Tan, N., Cao, X., and Tao, D. o1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *ArXiv Preprint: 2501.12570*, 2025a.

Luo, L., Liu, Y., Liu, R., Phatale, S., Guo, M., Lara, H., Li, Y., Shu, L., Zhu, Y., Meng, L., et al. Improve mathematical reasoning in language models by automated process supervision. *ArXiv Preprint: 2406.06592*, 2024.

Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Li, L. E., Popa, R. A., and Stoica, I. DeepScaleR: Surpassing o1-preview with a 1.5B model by scaling RL. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Mode 2025b. Notion Blog.

Madaan, A., Hermann, K., and Yazdanbakhsh, A. What makes chain-of-thought prompting effective? a counterfactual study. In *EMNLP*, pp. 1448–1535, 2023.

Mei, J., Gao, Y., Dai, B., Szepesvari, C., and Schuurmans, D. Leveraging non-uniformity in first-order non-convex optimization. In *ICML*, pp. 7555–7564, 2021.

Meng, Y., Xia, M., and Chen, D. SimPO: Simple preference optimization with a reference-free reward. In *NeurIPS*, pp. 124198–124235, 2024.

Merrill, W. and Sabharwal, A. The expressive power of transformers with chain of thought. In *ICLR*, 2024. URL https://openreview.net/forum?id=NjNGlPh8Wh.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *NeurIPS*, pp. 27730–27744, 2022.

Pal, A., Karkhanis, D., Dooley, S., Roberts, M., Naidu, S., and White, C. Smaug: Fixing failure modes of preference optimisation with DPO-positive. *ArXiv Preprint: 2402.13228*, 2024.

Pang, R. Y., Yuan, W., He, H., Cho, K., Sukhbaatar, S., and Weston, J. Iterative reasoning preference optimization. In *NeurIPS*, pp. 116617–116637, 2024.

Park, R., Rafailov, R., Ermon, S., and Finn, C. Disentangling length from quality in direct preference optimization. In *ACL*, pp. 4998–5017, 2024.

Peng, X., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *ArXiv Preprint: 1910.00177*, 2019.

Peters, J. and Schaal, S. Reinforcement learning by reward-weighted regression for operational space control. In *ICML*, pp. 745–750, 2007.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, pp. 53728–53741, 2023.

Rafailov, R., Hejna, J., Park, R., and Finn, C. From r to q^*: Your language model is secretly a Q-function. In *COLM*, 2024. URL https://openreview.net/forum?id=kEVcNxtqXk.

Razin, N., Malladi, S., Bhaskar, A., Chen, D., Arora, S., and Hanin, B. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *ICLR*, 2025. URL https://openreview.net/forum?id=uaMSBJDnRv.

Roit, P., Ferret, J., Shani, L., Aharoni, R., Cideron, G., Dadashi, R., Geist, M., Girgin, S., Hussenot, L., Keller, O., et al. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *ACL*, pp. 6252–6272, 2023.

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. Trust region policy optimization. In *ICML*, pp. 1889–1897, 2015.

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

Setlur, A., Garg, S., Geng, X., Garg, N., Smith, V., and Kumar, A. RL on incorrect synthetic data scales the efficiency of LLM math reasoning by eight-fold. In *NeurIPS*, pp. 43000–43031, 2024.

Setlur, A., Nagpal, C., Fisch, A., Geng, X., Eisenstein, J., Agarwal, R., Agarwal, A., Berant, J., and Kumar, A. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *ICLR*, 2025. URL https://openreview.net/forum?id=A6Y7AqlzLW.

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

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. HybridFlow: A flexible and efficient RLHF framework. In *EuroSys*, pp. 1279–1297. ACM, 2025.

Shi, B., Tang, M., Narasimhan, K. R., and Yao, S. Can language models solve Olympiad programming? In *COLM*, 2024. URL https://openreview.net/forum?id=kGa4fMtP9l.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Singh, A., Co-Reyes, J. D., Agarwal, R., Anand, A., Patil, P., Garcia, X., Liu, P. J., Harrison, J., Lee, J., Xu, K., Parisi, A. T., Kumar, A., Alemi, A. A., Rizkowsky, A., Nova, A., Adlam, B., Bohnet, B., Elsayed, G. F., Sedghi, H., Mordatch, I., Simpson, I., Gur, I., Snoek, J., Pennington, J., Hron, J., Kenealy, K., Swersky, K., Mahajan, K., Culp, L. A., Xiao, L., Bileschi, M., Constant, N., Novak, R., Liu, R., Warkentin, T., Bansal, Y., Dyer, E., Neyshabur, B., Sohl-Dickstein, J., and Fiedel, N. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=lNAyUngGFK. Expert Certification.

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. ProgPrompt: Generating situated robot task plans using large

language models. In *ICRA*, pp. 11523–11530. IEEE, 2023.

Skalse, J., Howe, N. H. R., Krasheninnikov, D., and Krueger, D. Defining and characterizing reward hacking. In *NeurIPS*, pp. 9460–9471, 2022.

Snell, C. V., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *ICLR*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Song, F., Yu, B., Li, M., Yu, H., Huang, F., Li, Y., and Wang, H. Preference ranking optimization for human alignment. In *AAAI*, pp. 18990–18998, 2024.

Srivastava, S., PV, A., Menon, S., Sukumar, A., Philipose, A., Prince, S., Thomas, S., et al. Functional benchmarks for robust evaluation of reasoning performance, and the reasoning gap. *ArXiv Preprint: 2402.19450*, 2024.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In *NeurIPS*, pp. 3008–3021, 2020.

Su, D., Sukhbaatar, S., Rabbat, M., Tian, Y., and Zheng, Q. Dualformer: Controllable fast and slow thinking by learning with randomized reasoning traces. In *ICLR*, 2025. URL https://openreview.net/forum?id=bmbRCRiNDu.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*, volume 1. MIT Press, 1998.

Tajwar, F., Singh, A., Sharma, A., Rafailov, R., Schneider, J., Xie, T., Ermon, S., Finn, C., and Kumar, A. Preference fine-tuning of LLMs should leverage suboptimal, on-policy data. In *ICML*, pp. 47441–47474, 2024.

Tang, Y., Guo, Z., Zheng, Z., Calandriello, D., Munos, R., Rowland, M., Richemond, P. H., Valko, M., Pires, B., and Piot, B. Generalized preference optimization: A unified approach to offline alignment. In *ICML*, pp. 47725–47742, 2024.

Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., et al. Kimi k1.5: Scaling reinforcement learning with LLMs. *ArXiv Preprint: 2501.12599*, 2025.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *ArXiv Preprint: 2302.13971*, 2023.

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. Solving math word problems with process-and outcome-based feedback. *ArXiv Preprint: 2211.14275*, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, pp. 6000–6010, 2017.

Wang, H., Qian, C., Zhong, W., Chen, X., Qiu, J., Huang, S., Jin, B., Wang, M., Wong, K.-F., and Ji, H. OTC: Optimal tool calls via reinforcement learning. *ArXiv Preprint: 2504.14870*, 2025.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *ACL*, pp. 9426–9439, 2024.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, pp. 24824–24837, 2022.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Wu, Y., Sun, Z., Li, S., Welleck, S., and Yang, Y. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *ICLR*, 2025. URL https://openreview.net/forum?id=VNckp7JEHn.

Xiang, V., Snell, C., Gandhi, K., Albalak, A., Singh, A., Blagden, C., Phung, D., Rafailov, R., Lile, N., Mahan, D., et al. Towards system 2 reasoning in LLMs: Learning how to think with meta chain-of-though. *ArXiv Preprint: 2501.04682*, 2025.

Xiao, J., Li, Z., Xie, X., Getzen, E., Fang, C., Long, Q., and Su, W. J. On the algorithmic bias of aligning large language models with RLHF: Preference collapse and matching regularization. *ArXiv Preprint: 2405.16455*, 2024.

Xie, Y., Kawaguchi, K., Zhao, Y., Zhao, J. X., Kan, M.-Y., He, J., and Xie, M. Q. Self-evaluation guided beam search for reasoning. In *NeurIPS*, pp. 41618–41650, 2023.

Xiong, W., Dong, H., Ye, C., Wang, Z., Zhong, H., Ji, H., Jiang, N., and Zhang, T. Iterative preference learning

from human feedback: Bridging theory and practice for RLHF under KL-constraint. In *ICML*, pp. 54715–54754, 2024.

Xiong, W., Yao, J., Xu, Y., Pang, B., Wang, L., Sahoo, D., Li, J., Jiang, N., Zhang, T., Xiong, C., and Dong, H. A minimalist approach to LLM reasoning: From rejection sampling to reinforce. *ArXiv Preprint: 2504.11343*, 2025.

Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Van Durme, B., Murray, K., and Kim, Y. J. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *ICML*, pp. 55204–55224, 2024.

Yang, Y., Campbell, D., Huang, K., Wang, M., Cohen, J., and Webb, T. Emergent symbolic mechanisms support abstract reasoning in large language models. *ArXiv Preprint: 2502.20332*, 2025.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, pp. 11809–11822, 2023.

Ye, Y., Huang, Z., Xiao, Y., Chern, E., Xia, S., and Liu, P. LIMO: Less is more for reasoning. *ArXiv Preprint: 2502.03387*, 2025.

Yu, F., Gao, A., and Wang, B. OVM, Outcome-supervised value models for planning in mathematical reasoning. In *NAACL*, pp. 858–875, 2024a.

Yu, F., Jiang, L., Kang, H., Hao, S., and Qin, L. Flow of reasoning: Efficient training of LLM policy with divergent thinking. In *ICML*, pp. To appear, 2025a.

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J., Li, Z., Weller, A., and Liu, W. MetaMath: Bootstrap your own mathematical questions for large language models. In *ICLR*, 2024b. URL https://openreview.net/forum?id=N8N0hgNDRt.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Fan, T., Liu, G., Liu, L., Liu, X., et al. DAPO: An open-source LLM reinforcement learning system at scale. *ArXiv Preprint: 2503.14476*, 2025b.

Yuan, H., Yuan, Z., Tan, C., Wang, W., Huang, S., and Huang, F. RRHF: Rank responses to align language models with human feedback. In *NeurIPS*, pp. 10935–10950, 2023a.

Yuan, L., Cui, G., Wang, H., Ding, N., Wang, X., Shan, B., Liu, Z., Deng, J., Chen, H., Xie, R., Lin, Y., Liu, Z., Zhou, B., Peng, H., Liu, Z., and Sun, M. Advancing LLM reasoning generalists with preference trees. In *ICLR*, 2025. URL https://openreview.net/forum?id=2ea5TNVR0c.

Yuan, Z., Yuan, H., Li, C., Dong, G., Lu, K., Tan, C., Zhou, C., and Zhou, J. Scaling relationship on learning mathematical reasoning with large language models. *ArXiv Preprint: 2308.01825*, 2023b.

Yue, X., Qu, X., Zhang, G., Fu, Y., Huang, W., Sun, H., Su, Y., and Chen, W. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *ICLR*, 2024. URL https://openreview.net/forum?id=yLClGs770I.

Zelikman, E., Wu, Y., Mu, J., and Goodman, N. D. STaR: self-taught reasoner bootstrapping reasoning with reasoning. In *NeurIPS*, pp. 15476–15488, 2022.

Zhang, K., Hong, Y., Bao, J., Jiang, H., Song, Y., Hong, D., and Xiong, H. GVPO: Group variance policy optimization for large language model post-training. *ArXiv Preprint: 2504.19599*, 2025a.

Zhang, L., Hosseini, A., Bansal, H., Kazemi, M., Kumar, A., and Agarwal, R. Generative verifiers: Reward modeling as next-token prediction. In *The NeurIPS Workshop on System-2 Reasoning at Scale*, 2024. URL https://openreview.net/forum?id=aLgXy8A7k7.

Zhang, Z., Zheng, C., Wu, Y., Zhang, B., Lin, R., Yu, B., Liu, D., Zhou, J., and Lin, J. The lessons of developing process reward models in mathematical reasoning. *ArXiv Preprint: 2501.07301*, 2025b.

Zhao, H., Winata, G. I., Das, A., Zhang, S.-X., Yao, D., Tang, W., and Sahu, S. RainbowPO: A unified framework for combining improvements in preference optimization. In *ICLR*, 2025. URL https://openreview.net/forum?id=trKee5pIFv.

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., and Liu, P. J. SLiC-HF: Sequence likelihood calibration with human feedback. *ArXiv Preprint: 2305.10425*, 2023.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q. V., and Chi, E. H. Least-to-most prompting enables complex reasoning in large language models. In *ICLR*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *ArXiv Preprint: 1909.08593*, 2019.

A. Limitations

Robustness of the AI feedback. A common challenge faced by all process-based reward mechanisms is the potential inaccuracy of the reward signal, particularly when it's the feedback from a single judger model. These inaccuracies can arise due to hallucinations, misinterpretation of reasoning steps, or over-penalization of minor errors. To address this, future works could attempt more robust method that enhances the reliability of the reward signal at the cost of increased computational resources. Specifically, we suggest the following API voting scheme, where multiple independent reward models are queried to evaluate the same response. The final reward is then taken as the minimum score among the ensemble, ensuring that any clear flaw in reasoning is penalized, even if only one model detects it. This conservative strategy reduces the risk of overlooking logical mistakes and promotes more cautious learning behavior.

Improved analysis for reasoning. We analyze the learning dynamics of SPO and GRPO only using a stylized model where the group size G, the number of reasoning steps H, and the number of choices per step are all set as 2. The learning dynamics are also restricted to population-level, neglecting the stochasticity of sampled trajectories. Such simplifications enable analytical tractability but limit generality. Another important limitation is that, even within this simplified model, our theoretical results only show that SPO *eventually* outperforms GRPO in learning the optimal policy. In contrast, our empirical evaluation shows that SPO consistently outperforms GRPO across all iterations (see Figure 2 in Appendix C.4). Filling in this gap between theoretical guarantees and empirical evaluation, extending the analysis to $G > 2$, $H > 2$ and more complex action spaces, as well as incorporating stochastic policy gradient updates, remains a promising yet challenging direction.

Efficient reasoning for all-positive-sample groups. SPO is designed to improve learning efficiency by leveraging RLAIF to extract rich information from all-negative-sample groups. However, for all-positive-sample groups, where all responses are correct, SPO offers no further improvement. We believe that incorporating an appropriate length-controlled reward framework could be helpful. Intuitively, once the correctness is achieved, the next stage of improving reasoning ability should focus on producing responses that are more concise, elegant, and direct. This direction has been recently explored in (Arora & Zanette, 2025; Team et al., 2025), and integrating such techniques into our framework remains a promising avenue for future research.

B. Further Related Works

We comment on other topics, including more discussions on reasoning through test-time compute, chain-of-thought and its variants, direct preference alignment methods, and reward models. For an overview of reasoning models and methods, we refer to recent surveys (Huang & Chang, 2023; Chen et al., 2025c).

More discussions on reasoning through test-time compute. OpenAI-o1 (Jaech et al., 2024) is among the first large-scale applications of RL to reasoning, and achieved state-of-the-art performance upon release. Following this trend, DeepSeek-R1 (Guo et al., 2025a) is the first open-weight model to match or exceed OpenAI-o1. Their real-world success stories have involved several simple yet novel techniques that enhance LLM reasoning through more test-time compute, including chain-of-thought (Wei et al., 2022), self-consistency (Wang et al., 2023), best-of-N sampling (Snell et al., 2025), process reward models (Lightman et al., 2024), Monte Carlo tree search (Silver et al., 2016; Hao et al., 2023), tree-of-thought (Yao et al., 2023), and recent works on preventing overthinking (Chen et al., 2024b; Team et al., 2025; Luo et al., 2025a; Arora & Zanette, 2025) and compressing chain-of-thought (Hao et al., 2024b; Cheng & Van Durme, 2024). More specifically, *chain-of-thought* is a reasoning approach where intermediate steps are explicitly written to make complex problem-solving processes more transparent and logical. *Self-consistency* suggests generating multiple final answers and returning the mode of an empirical distribution, enhancing test-time performance when test-time verifiers are unavailable. Unfortunately, it is computationally expensive and effective only when answers can be clustered. *Best-of-N sampling* resolves this issue by sampling answers from the model and selecting the best at test time according to the scoring function; however, it is sensitive to the accuracy of test-time scoring functions (Gao et al., 2023). *Process reward models* offer fine-grained supervision of chain-of-thought reasoning, but they might be vulnerable to reward hacking and introduce computation overhead. *Monte Carlo tree search* is a generic technique that allocates computational resources toward the most promising regions of the search space, and *tree-of-thought* and its extension (Besta et al., 2024; Gandhi et al., 2024) simplified this idea by exploring multiple reasoning paths in a specific structure, allowing language models to select the most promising line of thought for complex problem-solving. Both *length regularization* and *compressed chain-of-thought* are developed to reduce inference costs for reasoning, which is crucial for the economic feasibility, user experience and environmental sustainability of LLMs. In addition, several works have focused on specific reasoning tasks (Lampinen et al., 2024; Yang et al., 2025; Srivastava

et al., 2024; Huang et al., 2025; 2024; Guo et al., 2025b; Gou et al., 2024; Wang et al., 2025), demonstrating promising performance.

Chain-of-Thought and its variants. Chain-of-thought (CoT) refers to as a broad class of methods that generate an intermediate reasoning process before arriving at a final answer. These approaches either prompt LLMs (Wei et al., 2022; Khot et al., 2023; Zhou et al., 2023) or train LLMs to generate reasoning chains through supervised fine-tuning (SFT) (Yue et al., 2024; Yu et al., 2024b; Li et al., 2025) and/or RL (Wang et al., 2024; Shao et al., 2024; Havrilla et al., 2024; Yu et al., 2025a). While CoT has proven effective for certain tasks, its auto-regressive generation nature makes it challenging to mimic human reasoning on more complex problems (LeCun, 2022; Hao et al., 2023), which require planning and search. Recent efforts were devoted to equipping LLMs with tree search methods (Xie et al., 2023; Yao et al., 2023; Hao et al., 2024a) or training LLMs on search trajectories (Lehnert et al., 2024; Gandhi et al., 2024; Su et al., 2025). Several other works have investigated why CoT is effective. For example, (Madaan et al., 2023) used a counterfactual prompting approach to examine the relative contributions of prompt elements, including symbols (digits, entities) and patterns (equations). (Feng et al., 2023; Merrill & Sabharwal, 2024; Li et al., 2024a) analyzed CoT from the perspective of model expressivity, and (Feng et al., 2023) showed that employing CoT increases the effective depth of a transformer since the generated outputs are looped back to the input. This insight motivated the chain-of-continuous-thought paradigm (Hao et al., 2024b), and a related approach has been proposed in (Cheng & Van Durme, 2024).

Direct preference alignment methods. *Direct preference alignment methods* (such as DPO (Rafailov et al., 2023)) are simple and stable offline alternatives to online RLHF. Various DPO variants with other objectives have been proposed, including ranking ones beyond pairwise preference data (Dong et al., 2023; Yuan et al., 2023a; Song et al., 2024; Chen et al., 2024a; Liu et al., 2025a) and simple ones that do not rely on a reference model (Hong et al., 2024; Meng et al., 2024). Since DPO does not train a reward model, the limited size of human labels becomes a bottleneck. To alleviate this limitation, subsequent works proposed to augment preference data using a trained SFT policy (Zhao et al., 2023) or a refined SFT policy with rejection sampling (Liu et al., 2024a). The DPO loss was recently extended to token-level MDP (Rafailov et al., 2024) given that the transition is deterministic – which has covered the fine-tuning of LLMs – and more general RL problems (Azar et al., 2024). There are other DPO variants (Ethayarajh et al., 2024; Park et al., 2024; Xu et al., 2024; Tang et al., 2024; Meng et al., 2024; Chen et al., 2025a; Zhao et al., 2025). For example, (Ethayarajh et al., 2024) designed the specific loss using a prospect theory, (Tang et al., 2024) optimized a general preference loss instead of the log-likelihood loss, and (Meng et al., 2024) aligned the reward function in the preference optimization objective with the generation metric. (Dong et al., 2024) and (Xiong et al., 2024) proposed to generate human feedback in an online fashion to mitigate the distribution-shift and over-parameterization phenomenon. This improves DPO for complex reasoning tasks (Pang et al., 2024). Several other works focus on *unintentional alignment* of DPO and developing new methods (Pal et al., 2024; Tajwar et al., 2024; Liu et al., 2024b; Xiao et al., 2024; Yuan et al., 2025; Razin et al., 2025; Chen et al., 2025b). Among these works, (Razin et al., 2025) proposed to measure the similarity between preferred and dispreferred responses using the centered hidden embedding similarity (CHES) score and showed that filtering out preference pairs with small CHES score improves DPO, while (Chen et al., 2025b) proposed a new method based on comparison oracles, and showed that combining it with DPO effectively alleviated the issue of unintentional alignment.

Reward models. For the prompt $\mathbf{x}$ with a ground-truth response $\mathbf{y}^\star_{\mathbf{x}}$, we evaluate by implementing a regular expression match on the final answer (Hendrycks et al., 2021): $r(\mathbf{x}, \mathbf{y}) = 1$ if $\mathbf{y}$ matches $\mathbf{y}^\star_{\mathbf{x}}$ on the *final answer* and $r(\mathbf{x}, \mathbf{y}) = 0$ otherwise. An *outcome reward* model (ORM) (Cobbe et al., 2021; Uesato et al., 2022) is trained for estimating $r(\mathbf{x}, \mathbf{y})$. In particular, we first choose $\mathbf{x} \in \mathcal{D}$ and collect training samples $(\mathbf{x}, \mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x}), r(\mathbf{x}, \mathbf{y}))$. Then, we take $(\mathbf{x}, \mathbf{y})$ as input and train an ORM to predict $r(\mathbf{x}, \mathbf{y})$. This can be done using binary classification (Cobbe et al., 2021; Yu et al., 2024a), direct preference optimization (Hosseini et al., 2024) or next-token prediction (Zhang et al., 2024). Previous works also train LLMs on self-generated data using the ground-truth outcome reward model with either supervised fine-tuning (Singh et al., 2024; Yuan et al., 2023b; Zelikman et al., 2022) or online RL (Bi et al., 2024; Guo et al., 2025a). A *process reward* model (PRM) is trained to score a_h at $\mathbf{s}_h = (\mathbf{x}, a_1, \ldots, a_{h-1})$ either using human annotations (Lightman et al., 2024) or the value functions based on LLM-generated data (Wang et al., 2024; Luo et al., 2024; Setlur et al., 2025); indeed, PRMs estimate either the likelihood of future success or the change in the likelihood of future success before and after taking a_h. In addition, PRMs were also developed to improve search methods (Snell et al., 2025; Wu et al., 2025), and to identify the "first pit" in an incorrect reasoning trajectory to construct preference pairs for direct preference alignment (Hwang et al., 2024; Setlur et al., 2024).

C. Missing Proofs

We first present the detailed setup for our stylized model and prove several technical lemmas. Then, we use these lemmas to prove our main result in Theorem 3.2.

C.1. Stylized model

We consider a policy parameterized by a softmax function, which is standard in the analysis of reinforcement learning methods (Agarwal et al., 2020; Mei et al., 2021; Li et al., 2024b):

$$\pi_\theta(a_{1:T} \,|\, \mathbf{x}) = \prod_{t=1}^{T} \pi_{\theta_t}(a_t \,|\, \mathbf{x}, a_{1:t-1}) = \prod_{t=1}^{T} \frac{\exp(\theta_t^{\mathbf{x}, a_{1:t-1}, a_t})}{\sum_{a_t' \in \mathcal{V}^\star} \exp(\theta_t^{\mathbf{x}, a_{1:t-1}, a_t'})},$$

By convention, we assume that $\pi_{\theta_1}(a_1 \,|\, \mathbf{x}, a_{1:0}) = \pi_{\theta_1}(a_1 \,|\, \mathbf{x})$.

For simplicity, we perform our analysis in the likelihood space rather than in the parameter space (i.e., θ) directly. Indeed, we define the key quantities as follows,

$$p \doteq \pi_{\theta_1}(a_1 = 2 \,|\, \mathbf{x}) = \frac{e^{\theta_1^{\mathbf{x},2}}}{e^{\theta_1^{\mathbf{x},1}} + e^{\theta_1^{\mathbf{x},2}}}, \qquad q \doteq \pi_{\theta_2}(a_2 = 2 \,|\, \mathbf{x}, a_1 = 2) = \frac{e^{\theta_2^{\mathbf{x},2,2}}}{e^{\theta_2^{\mathbf{x},2,1}} + e^{\theta_2^{\mathbf{x},2,2}}}.$$

Note that the original 4-dimensional parameter space defined by $\theta_1^{\mathbf{x},1}$, $\theta_1^{\mathbf{x},2}$, $\theta_2^{\mathbf{x},2,1}$ and $\theta_2^{\mathbf{x},2,2}$ in $\mathbb{R}$ is reduced to a 2-dimensional likelihood space defined by $p, q \in [0, 1]$.

We rewrite the generic GRPO update with a step size $\eta > 0$ as follows,

$$\theta^{(k+1)} = \theta^{(k)} + \eta \cdot g(\theta), \quad \text{where } g(\theta) = \frac{1}{NGH} \left(\sum_{i=1}^{N} \sum_{k=1}^{G} \sum_{h=1}^{H} s_\theta(\mathbf{x}^i, a_{1:h-1}^{i,k}) A_{i,k} \right),$$

where N is the number of prompts, G is the number of groups, H is the number of reasoning steps in each response, $s_\theta(\mathbf{x}^i, a_{1:h-1}^{i,k}) := \nabla_\theta \log \pi_\theta(a_t | \mathbf{x}, a_{1:h-1})$ is the score function, and the advantage $A_{i,k}$ is defined by

$$A_{i,k} = \frac{r(\mathbf{x}^i, \mathbf{y}^{i,k}) - (1/G) \sum_{j=1}^{G} r(\mathbf{x}^i, \mathbf{y}^{i,j})}{\sqrt{(1/G) \sum_{j=1}^{G} (r(\mathbf{x}^i, \mathbf{y}^{i,j}) - (1/G) \sum_{j'=1}^{G} r(\mathbf{x}^i, \mathbf{y}^{i,j'}))^2}}$$

To distinguish, we denote $g_{\texttt{GRPO}}(\cdot)$ as the gradient estimator using classical outcome reward model r, and $g_{\texttt{SPO}}(\cdot)$ as the gradient estimator using the reward $r_{\texttt{AIF}}$ as proposed in Section 3.1.

For our simple stylized model, we compute the score functions in terms of likelihood parameters p, q as follows,

$$s(a_1 = 1 \,|\, \mathbf{x}) = \begin{bmatrix} p \\ -p \\ 0 \\ 0 \end{bmatrix}, \quad s(a_1 = 2 \,|\, \mathbf{x}) = \begin{bmatrix} p - 1 \\ 1 - p \\ 0 \\ 0 \end{bmatrix},$$

and

$$s(a_2 = 1 \,|\, \mathbf{x}, a_1 = 2) = \begin{bmatrix} 0 \\ 0 \\ q \\ -q \end{bmatrix}, \quad s(a_2 = 2 \,|\, \mathbf{x}, a_1 = 2) = \begin{bmatrix} 0 \\ 0 \\ q - 1 \\ 1 - q \end{bmatrix}.$$

Note that we restrict the sample space to $\mathbf{y} \in \{(1, 1), (2, 1), (2, 2)\}$, excluding the sample $(1, 2)$. The responses can be drawn i.i.d. from the distribution as follows,

$$(a_1, a_2) = \begin{cases} (1, 1), & \text{w.p. } 1 - p, \\ (2, 1), & \text{w.p. } p(1 - q), \\ (2, 2), & \text{w.p. } pq. \end{cases}.$$

We set $G = 2$ and focus on the SPO and GRPO training dynamics with population-level policy gradient which can be computed exactly for the stylized model as follows,

$$\bar{g}_{\texttt{SPO}}(\theta) = \mathbb{E}[g_{\texttt{SPO}}(\theta)] = \frac{1}{2}\begin{bmatrix} p(p-1) \\ p(1-p) \\ p^2q(q-1) \\ p^2q(1-q) \end{bmatrix}, \quad \bar{g}_{\texttt{GRPO}}(\theta) = \mathbb{E}[g_{\texttt{GRPO}}(\theta)] = \frac{1}{2}\begin{bmatrix} p(p-1)q \\ p(1-p)q \\ pq(q-1) \\ pq(1-q) \end{bmatrix}.$$

Since $g_{\texttt{GRPO}}(\theta)$ and $g_{\texttt{SPO}}(\theta)$ concentrate around $\bar{g}_{\texttt{GRPO}}(\theta)$ and $\bar{g}_{\texttt{SPO}}(\theta)$ when the number of samples in each group is sufficiently large, it is reasonable to analyze the population-level dynamics at first. Note that the high-probability guarantees for the sample-level dynamics can be derived using concentration inequalities under certain conditions.

Now, we can explicitly write down the SPO and GRPO update rules with $\eta = 1$ using the likelihood parameters p and q as follows,

$$\begin{cases} p_{\texttt{SPO}}^{(k+1)} = \exp(f_{11}(p_{\texttt{SPO}}^{(k)})), \\ q_{\texttt{SPO}}^{(k+1)} = \exp(f_{12}(p_{\texttt{SPO}}^{(k)}, q_{\texttt{SPO}}^{(k)})), \end{cases} \quad \text{and} \quad \begin{cases} p_{\texttt{GRPO}}^{(k+1)} = \exp(f_{21}(p_{\texttt{GRPO}}^{(k)}, q_{\texttt{GRPO}}^{(k)})), \\ q_{\texttt{GRPO}}^{(k+1)} = \exp(f_{22}(p_{\texttt{GRPO}}^{(k)}, q_{\texttt{GRPO}}^{(k)})), \end{cases} \tag{3}$$

where the functions f_{ij} are defined by

$$\begin{aligned} f_{11}(p) &= \log(p) + p(1-p) - \log(1 - p + pe^{p(1-p)}), \\ f_{21}(p,q) &= \log(p) + p(1-p)q - \log(1 - p + pe^{p(1-p)q}), \\ f_{12}(p,q) &= \log(q) + p^2q(1-q) - \log(1 - q + qe^{p^2q(1-q)}), \\ f_{22}(p,q) &= \log(q) + pq(1-q) - \log(1 - q + qe^{pq(1-q)}). \end{aligned} \tag{4}$$

C.2. Technical lemmas

We provide several technical lemmas that are important to the subsequent proof of Theorem 3.2. Indeed, the first lemma summarizes the properties of particular functions related to the aforementioned functions f_{11}, f_{21}, f_{12} and f_{22} from Equation (4).

Lemma C.1. *The following statements hold true,*

(i) The function f_{11} is strictly increasing on $(0, 1)$.
(ii) The function $h_p(x) := x - \log(1 - p + pe^x)$ is strictly increasing for any fixed $p \in (0, 1)$.
(iii) The function f_{21} is strictly increasing in either p for any fixed q or q for any fixed p on $(0, 1)$.
(iv) The function $\varphi(x) := \log(1 + (1/2)e^{-e^x})$ is strictly concave on $(-\infty, 0)$.

Proof. First of all, we have

$$f_{11}'(p_1) = \frac{1+(1-2p_1)p_1(1-p_1)}{p_1(1+p_1+p_1e^{p_1(1-p_1)})} > \frac{3}{4p_1(1+p_1+p_1e^{p_1(1-p_1)})} > 0.$$

Thus, the function f_{11} is strictly increasing on $(0, 1)$.

Furthermore, we have

$$h_p'(x) = 1 - \frac{pe^x}{1-p+pe^x} = \frac{1-p}{1-p+pe^x} \overset{0<p<1}{>} 0$$

Thus, the function $h_p(x)$ is strictly increasing.

Moreover, we have

$$\begin{aligned} \frac{\partial f_{21}(p_1,p_2)}{p_1} &= \frac{1+p_2(1-2p_1)p_1(1-p_1)}{p_1(1+p_1+p_1e^{p_2p_1(1-p_1)})} > \frac{3}{4p_1(1+p_1+p_1e^{p_1(1-p_1)})} > 0, \\ \frac{\partial f_{21}(p_1,p_2)}{p_2} &= \frac{p_1(1-p_1)^2}{1+p_1+p_1e^{p_2p_1(1-p_1)}} > 0. \end{aligned}$$

Thus, the function f_{21} is strictly increasing in either p for any fixed q or q for any fixed p on $(0, 1)$.

Finally, we have

$$\varphi''(x) = \frac{(e^{x+e^x}/2 - e^{e^x}/2 - 1/4)e^x}{e^{2e^x} + e^{e^x} + 1/4}.$$

Since $u = e^x \in (0, 1)$ for $x < 0$, we have

$$(ue^u/2 - e^u/2 - 1/4)u = (e^u(u-1)/2 - 1/4)u < -(1/4)u < 0.$$

Thus, $\varphi''(x) < 0$ for all $x < 0$ which shows that f is strictly concave on $(-\infty, 0)$. □

The second lemma presents basic inequalities and we provide the proofs for the sake of completeness.

Lemma C.2. *The following statements hold true,*

(i) For $x \in (0, \frac{1}{2})$, we have

$$1 + x + x^2 + x^3 + x^4 + x^5 \le \tfrac{1}{1-x} \le 1 + x + x^2 + x^3 + x^4 + 2x^5.$$

(ii) For $x > 0$, we have

$$1 - x + \tfrac{x^2}{2} - \tfrac{x^3}{6} + \tfrac{x^4}{24} - \tfrac{x^5}{120} \le e^{-x} \le 1 - x + \tfrac{x^2}{2} - \tfrac{x^3}{6} + \tfrac{x^4}{24}.$$

(iii) For $x \in (-\frac{1}{2}, 0)$, we have

$$\sqrt{1+x} \le 1 + \tfrac{x}{2} - \tfrac{x^2}{8} + \tfrac{x^3}{16} - \tfrac{5x^4}{128} + \tfrac{7x^5}{256}.$$

Proof. First of all, we have

$$\tfrac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + \tfrac{x^5}{1-x}.$$

Since $x \in (0, \frac{1}{2})$, we have $x^5 \le \frac{x^5}{1-x} \le 2x^5$. Thus, the first desired inequality holds true.

Furthermore, we have $f^{(n)}(x) = (-1)^n e^{-x}$ where $f(x) = e^{-x}$. Then, we have $f^{(5)}(\xi) < 0$ for all $\xi > 0$ and $f^{(6)}(\xi) > 0$ for all $\xi > 0$. This implies

$$\begin{aligned} e^{-x} &= 1 - x + \tfrac{x^2}{2} - \tfrac{x^3}{6} + \tfrac{x^4}{24} + \tfrac{f^{(5)}(\xi_1)x^5}{5!} \le 1 - x + \tfrac{x^2}{2} - \tfrac{x^3}{6} + \tfrac{x^4}{24}, \\ e^{-x} &= 1 - x + \tfrac{x^2}{2} - \tfrac{x^3}{6} + \tfrac{x^4}{24} - \tfrac{x^5}{120} + \tfrac{f^{(6)}(\xi_2)x^6}{6!} \ge 1 - x + \tfrac{x^2}{2} - \tfrac{x^3}{6} + \tfrac{x^4}{24} - \tfrac{x^5}{120}. \end{aligned}$$

Thus, the second desired inequality holds true.

Finally, we have $f^{(6)}(x) = -\frac{105}{64}(1+x)^{-11/2}$ where $f(x) = \sqrt{1+x}$. Then, we have $f^{(6)}(\xi) < 0$ for all $\xi \in (-1/2, 0)$. This implies

$$\sqrt{1+x} = 1 + \tfrac{x}{2} - \tfrac{x^2}{8} + \tfrac{x^3}{16} - \tfrac{5x^4}{128} + \tfrac{7x^5}{256} + \tfrac{f^{(6)}(\xi)x^6}{6!} \le 1 + \tfrac{x}{2} - \tfrac{x^2}{8} + \tfrac{x^3}{16} - \tfrac{5x^4}{128} + \tfrac{7x^5}{256}.$$

Thus, the third desired inequality holds true. □

The third lemma presents an inequality which plays a key role in the subsequent proof of Theorem 3.2. It is proved using basic inequalities from Lemma C.2.

Lemma C.3. *We define the auxiliary functions as follows,*

$$A(x) = 1 + \left(\tfrac{1}{x} - 1\right) e^{-x(1-x)},\ B(x,y) = 1 + \left(\tfrac{1}{y} - 1\right) e^{-x^2 y(1-y)},\ C(x,y) = 1 + \left(\tfrac{1}{\sqrt{xy}} - 1\right) e^{-xy(1-\sqrt{xy})}.$$

Then, we have $C(x,y)^2 > A(x)B(x,y)$ for all x and y satisfying $\frac{31}{32} < y < x < 1$.

Proof. For simplicity, we let $h = 1 - y \in (0, \frac{1}{2})$, $a = 1 - x \in (0, \frac{1}{2})$ and $\varepsilon = \frac{x-y}{h} \in (0, 1)$. First, we rewrite $A(x)$ in terms of a as follows,

$$A(x) = 1 + \left(\tfrac{1}{1-a} - 1\right) e^{-a(1-a)}.$$

By using the upper bound in Item C.2(ii), we have

$$\begin{aligned} e^{-a(1-a)} &\le 1 - a(1-a) + \tfrac{a^2(1-a)^2}{2} - \tfrac{a^3(1-a)^3}{6} + \tfrac{a^4(1-a)^4}{24} \\ &= 1 - a + \tfrac{3}{2}a^2 - \tfrac{7}{6}a^3 + \tfrac{25}{24}a^4 - \tfrac{2}{3}a^5 + \tfrac{5}{12}a^6 - \tfrac{1}{6}a^7 + \tfrac{1}{24}a^8 \\ &= 1 - a + \tfrac{3}{2}a^2 - \tfrac{7}{6}a^3 + \tfrac{25}{24}a^4 + a^5 R_1(a), \end{aligned}$$

where R_1 is defined by

$$R_1(a) = -\tfrac{2}{3} + \tfrac{5}{12}a - \tfrac{1}{6}a^2 + \tfrac{1}{24}a^3.$$

Since $a \in (0, \frac{1}{2})$, we have

$$R_1'(a) = \tfrac{5}{12} - \tfrac{1}{3}a + \tfrac{1}{8}a^2 > \tfrac{1}{8}a^2 > 0.$$

Thus, we have $R_1(a) < R_1(1/2) < 0$. This together with $a > 0$ yields

$$e^{-a(1-a)} \leq 1 - a + \tfrac{3}{2}a^2 - \tfrac{7}{6}a^3 + \tfrac{25}{24}a^4.$$

Combining the above inequality with the upper bound in Item C.2(i) yields

$$\begin{aligned} A(x) &\leq 1 + (a + a^2 + a^3 + a^4 + 2a^5)\left(1 - a + \tfrac{3}{2}a^2 - \tfrac{7}{6}a^3 + \tfrac{25}{24}a^4\right) \\ &= 1 + a + \tfrac{3}{2}a^3 + \tfrac{1}{3}a^4 + \tfrac{19}{8}a^5 - \tfrac{5}{8}a^6 + \tfrac{23}{8}a^7 - \tfrac{31}{24}a^8 + \tfrac{25}{12}a^9 \\ &= 1 + a + \tfrac{3}{2}a^3 + \tfrac{1}{3}a^4 + \tfrac{19}{8}a^5 + a^6 R_2(a), \end{aligned}$$

where R_2 is defined by

$$R_2(a) = -\tfrac{5}{8} + \tfrac{23}{8}a - \tfrac{31}{24}a^2 + \tfrac{25}{12}a^3.$$

Since $a \in (0, \frac{1}{2})$, we have

$$R_2'(a) = \tfrac{23}{8} - \tfrac{31}{12}a + \tfrac{25}{4}a^2 > \tfrac{25}{4}a^2 > 0.$$

Thus, we have $R_2(a) < R_2(1/2) = 3/4$. This together with $a < \frac{1}{2}$ yields

$$A(x) \leq 1 + a + \tfrac{3}{2}a^3 + \tfrac{1}{3}a^4 + \tfrac{19}{8}a^5 + \tfrac{3}{4}a^6 \leq 1 + a + \tfrac{3}{2}a^3 + \tfrac{1}{3}a^4 + \tfrac{11}{4}a^5.$$

Finally, we rewrite the above inequality using $a = (1-\varepsilon)h$ as follows,

$$A(x) \leq 1 + (1-\varepsilon)h + \tfrac{3}{2}(1-\varepsilon)^3h^3 + \tfrac{1}{3}(1-\varepsilon)^4h^4 + \tfrac{11}{4}h^5. \tag{5}$$

Furthermore, we obtain an upper bound for $B(x, y)$ using Lemma C.2. Indeed, we have

$$(1 - h + \varepsilon h)^2 h(1-h) = h + (2\varepsilon - 3)h^2 + (\varepsilon^2 - 4\varepsilon + 3)h^3 - (1 - 2\varepsilon + \varepsilon^2)h^4.$$

By using the upper bound in Item C.2(ii), we have

$$e^{-(1-h+\varepsilon h)^2 h(1-h)} \leq 1 - h + \left(\tfrac{7}{2} - 2\epsilon\right)h^2 + \left(-\epsilon^2 + 6\epsilon - \tfrac{37}{6}\right)h^3 + 11h^4.$$

We rewrite $B(x, y)$ in terms of h and ε as follows,

$$B(x,y) = 1 + \left(\tfrac{1}{1-h} - 1\right)e^{-(1-h+\varepsilon h)^2 h(1-h)}$$

Applying the upper bound in Item C.2(i) yields

$$\tfrac{1}{1-h} \leq 1 + h + h^2 + h^3 + h^4 + 2h^5.$$

Putting these pieces together yields

$$B(x,y) \leq 1 + h + \left(\tfrac{7}{2} - 2\varepsilon\right)h^3 + \left(-\tfrac{8}{3} + 4\varepsilon - \varepsilon^2\right)h^4 + \tfrac{37}{8}h^5. \tag{6}$$

Combining Equation (5) and Equation (6) yields

$$\begin{aligned} A(x)B(x,y) \leq\ & 1 + (2-\varepsilon)h + (1-\varepsilon)h^2 \\ & + \left(5 - \tfrac{13\varepsilon}{2} + \tfrac{9}{2}\varepsilon^2 - \tfrac{3}{2}\varepsilon^3\right)h^3 + \left(\tfrac{8}{3} - \tfrac{22}{3}\varepsilon + \tfrac{15}{2}\varepsilon^2 - \tfrac{17}{6}\varepsilon^3 + \tfrac{1}{3}\varepsilon^4\right)h^4 + 13h^5. \end{aligned} \tag{7}$$

Finally, we obtain a lower bound for $C(x, y)$. Indeed, we set $v = 1 - \sqrt{xy} \in (0, \frac{1}{2})$ and rewrite $C(x, y)$ in terms of v as follows,

$$C(x,y) = 1 + \left(\tfrac{1}{1-v} - 1\right)e^{-v(1-v)^2} =: \phi(v).$$

By using the lower bound in Item C.2(ii), we have

$$e^{-v(1-v)^2} \geq 1 - v + \frac{5}{2}v^2 - \frac{19}{6}v^3 + \frac{97}{24}v^4 - \frac{601}{120}v^5.$$

Applying the lower bound in Item C.2(i) yields

$$\tfrac{1}{1-v} \geq 1 + v + v^2 + v^3 + v^4 + v^5.$$

Putting these pieces together yields

$$\phi(v) \geq 1 + v + \tfrac{5}{2}v^3 - \tfrac{2}{3}v^4 + 2v^5. \tag{8}$$

Note that $xy - 1 = (\varepsilon - 2)h + (1-\varepsilon)h^2$. Then, applying Item C.2(iii) to $xy - 1 \in (-\frac{1}{2}, 0)$ yields

$$\sqrt{xy} \leq 1 + \left(\tfrac{\varepsilon}{2} - 1\right)h - \tfrac{\varepsilon^2}{8}h^2 + \tfrac{1}{16}\varepsilon^2(\varepsilon - 2)h^3 + \tfrac{1}{8}\varepsilon^2(\varepsilon - 1)h^4 + \tfrac{1}{8}h^5.$$

which implies

$$v = 1 - \sqrt{xy} \geq \left(1 - \tfrac{\varepsilon}{2}\right)h + \tfrac{\varepsilon^2}{8}h^2 + \tfrac{1}{16}\varepsilon^2(2 - \varepsilon)h^3 + \tfrac{1}{8}\varepsilon^2(1 - \varepsilon)h^4 - \tfrac{1}{8}h^5. \tag{9}$$

Since $C(x,y) = \frac{1}{\exp(f_{21}(\sqrt{xy}, \sqrt{xy}))}$ is strictly decreasing in $\sqrt{xy}$ (c.f. Item C.1(iii)), we have that ϕ is strictly increasing in v. Thus, we have

$$\begin{aligned} C(x,y) = \phi(v) &\geq \phi\left(\left(1 - \tfrac{\varepsilon}{2}\right)h + \tfrac{\varepsilon^2}{8}h^2 + \tfrac{1}{16}\varepsilon^2(2 - \varepsilon)h^3 + \tfrac{1}{8}\varepsilon^2(1 - \varepsilon)h^4 - \tfrac{1}{8}h^5\right) \\ &\geq 1 + \left(1 - \tfrac{\varepsilon}{2}\right)h + \tfrac{\varepsilon^2}{8}h^2 + \left(-\tfrac{3}{8}\varepsilon^3 + 2\varepsilon^2 - \tfrac{15}{4}\varepsilon + \tfrac{5}{2}\right)h^3 + \left(-\tfrac{35}{48}\varepsilon^3 + \tfrac{1}{16}\varepsilon^2 + \tfrac{4}{3}\varepsilon\tfrac{2}{3}\right)h^4 - \tfrac{1}{100}h^5. \end{aligned}$$

where the first inequality uses Equation (9) and the second inequality uses Equation (8). This implies

$$\begin{aligned} C(x,y)^2 \geq\ & 1 + (2 - \varepsilon)h + \left(1 - \varepsilon + \tfrac{\varepsilon^2}{2}\right)h^2 \\ & + \left(-\tfrac{7\varepsilon^3}{8} + \tfrac{17\varepsilon^2}{4} - \tfrac{15\varepsilon}{2} + 5\right)h^3 + \left(\tfrac{25\varepsilon^4}{64} - \tfrac{101\varepsilon^3}{24} + \tfrac{63\varepsilon^2}{8} - \tfrac{22\varepsilon}{3} + \tfrac{11}{3}\right)h^4 - 2h^5. \end{aligned} \tag{10}$$

Since $\frac{31}{32} < y < x < 1$, we have $h < \frac{1}{32}$. This together with Eq. (7) and Eq. (10) yields

$$\begin{aligned} C(x,y)^2 - A(x)B(x,y) &\geq \tfrac{1}{2}\varepsilon^2h^2 + \left(-\varepsilon - \tfrac{1}{4}\varepsilon^2 + \tfrac{5}{8}\varepsilon^3\right)h^3 + \left(1 + \tfrac{3}{8}\varepsilon^2 - \tfrac{11}{8}\varepsilon^3\right)h^4 - 15h^5 \\ &> h^2\left(\tfrac{1}{2}\varepsilon^2 + \left(-\varepsilon - \tfrac{1}{4}\varepsilon^2\right)h + \left(\tfrac{17}{32} - \varepsilon^2\right)h^2\right) \\ &> 0.49\varepsilon^2 - \varepsilon h + \tfrac{17}{32}h^2 \ > \ 0, \end{aligned}$$

where the second inequality uses $\epsilon \in (0,1)$ and the last inequality holds true since the quadratic function in ε has negative discriminant. □

The last lemma presents some properties for the population-level SPO and GRPO dynamics.

Lemma C.4. *Under the assumptions from Theorem 3.2, the following statements hold true,*

(i) $p_{SPO}^{(k)}, q_{SPO}^{(k)}, p_{GRPO}^{(k)}, q_{GRPO}^{(k)} \in (0,1)$ *for all* $k \geq 0$.
(ii) $p_{SPO}^{(k)}, q_{SPO}^{(k)}, p_{GRPO}^{(k)}, q_{GRPO}^{(k)}$ *are strictly increasing in* k *and lie in* $(\frac{1}{2}, 1)$ *for all* $k \geq 1$.
(iii) $p_{SPO}^{(k)} > q_{SPO}^{(k)}$ *for all* $k \geq 1$.

Proof. We first rewrite the update rule in Equation (3) as follows,

$$\begin{aligned} p_{\text{SPO}}^{(k+1)} &= p_{\text{SPO}}^{(k)} \frac{e^{\Delta_{\text{SPO},p}^{(k)}}}{1 - p_{\text{SPO}}^{(k)} + p_{\text{SPO}}^{(k)} e^{\Delta_{\text{SPO},p}^{(k)}}}, && \text{where} \quad \Delta_{\text{SPO},p}^{(k)} = p_{\text{SPO}}^{(k)}(1 - p_{\text{SPO}}^{(k)}), \\ q_{\text{SPO}}^{(k+1)} &= q_{\text{SPO}}^{(k)} \frac{e^{\Delta_{\text{SPO},q}^{(k)}}}{1 - q_{\text{SPO}}^{(k)} + q_{\text{SPO}}^{(k)} e^{\Delta_{\text{SPO},q}^{(k)}}}, && \text{where} \quad \Delta_{\text{SPO},q}^{(k)} = (p_{\text{SPO}}^{(k)})^2 q_{\text{SPO}}^{(k)}(1 - q_{\text{SPO}}^{(k)}), \\ p_{\text{GRPO}}^{(k+1)} &= p_{\text{GRPO}}^{(k)} \frac{e^{\Delta_{\text{GRPO},p}^{(k)}}}{1 - p_{\text{GRPO}}^{(k)} + p_{\text{GRPO}}^{(k)} e^{\Delta_{\text{GRPO},p}^{(k)}}}, && \text{where} \quad \Delta_{\text{GRPO},p}^{(k)} = p_{\text{GRPO}}^{(k)}(1 - p_{\text{GRPO}}^{(k)}) q_{\text{GRPO}}^{(k)}, \\ q_{\text{GRPO}}^{(k+1)} &= q_{\text{GRPO}}^{(k)} \frac{e^{\Delta_{\text{GRPO},q}^{(k)}}}{1 - q_{\text{GRPO}}^{(k)} + q_{\text{GRPO}}^{(k)} e^{\Delta_{\text{GRPO},q}^{(k)}}}, && \text{where} \quad \Delta_{\text{GRPO},q}^{(k)} = p_{\text{GRPO}}^{(k)} q_{\text{GRPO}}^{(k)}(1 - q_{\text{GRPO}}^{(k)}). \end{aligned}$$

First of all, the uniform initialization yields the desired result for $k = 0$. Suppose $p_{\text{SPO}}^{(k)} \in (0, 1)$ for some $k \geq 0$. Then, we have

$$1 - p_{\text{SPO}}^{(k)} + p_{\text{SPO}}^{(k)} e^{\Delta_{\text{SPO},p}^{(k)}} > p_{\text{SPO}}^{(k)} e^{\Delta_{\text{SPO},p}^{(k)}} > 0,$$

which implies $p_{\text{SPO}}^{(k+1)} \in (0, 1)$. By induction, we have $p_{\text{SPO}}^{(k)} \in (0, 1)$ for all $k \geq 0$. Similarly, we can show that $q_{\text{SPO}}^{(k)}, p_{\text{GRPO}}^{(k)}, q_{\text{GRPO}}^{(k)} \in (0, 1)$ for all $k \geq 0$.

Furthermore, we have $\Delta_{\text{SPO},p}^{(k)} > 0$ since $p_{\text{SPO}}^{(k)} \in (0, 1)$. This implies

$$\frac{p_{\text{SPO}}^{(k+1)}}{p_{\text{SPO}}^{(k)}} = \frac{1}{(1-p_{\text{SPO}}^{(k)})e^{-\Delta_{\text{SPO},p}^{(k)}} + p_{\text{SPO}}^{(k)}} > \frac{1}{1-p_{\text{SPO}}^{(k)}+p_{\text{SPO}}^{(k)}} = 1.$$

Since $p_{\text{SPO}}^{(0)} = \frac{1}{2}$, we have $p_{\text{SPO}}^{(k)} \in (\frac{1}{2}, 1)$ for all $k \geq 1$. Similarly, we can show that $q_{\text{SPO}}^{(k)}, p_{\text{GRPO}}^{(k)}, q_{\text{GRPO}}^{(k)}$ are strictly increasing and lie in $(\frac{1}{2}, 1)$.

Finally, we have $p_{\text{SPO}}^{(0)} \geq q_{\text{SPO}}^{(0)}$. Thus, it suffices to show that $p_{\text{SPO}}^{(k)} \geq q_{\text{SPO}}^{(k)}$ implies $p_{\text{SPO}}^{(k+1)} > q_{\text{SPO}}^{(k+1)}$ for all $k \geq 0$. Indeed, Item C.1(i) and $p_{\text{SPO}}^{(k)} \geq q_{\text{SPO}}^{(k)}$ yield

$$p_{\text{SPO}}^{(k+1)} = \exp(f_{11}(p_{\text{SPO}}^{(k)})) \geq \exp(f_{11}(q_{\text{SPO}}^{(k)})) = \exp(\log(q_{\text{SPO}}^{(k)}) + h_{q_{\text{SPO}}^{(k)}}(q_{\text{SPO}}^{(k)}(1 - q_{\text{SPO}}^{(k)}))).$$

Then, Item C.1(ii) and $p_{\text{SPO}}^{(k)} \in (0, 1)$ yield

$$\exp(\log(q_{\text{SPO}}^{(k)}) + h_{q_{\text{SPO}}^{(k)}}(q_{\text{SPO}}^{(k)}(1 - q_{\text{SPO}}^{(k)}))) > \exp(\log q_{\text{SPO}}^{(k)} + h_{q_{\text{SPO}}^{(k)}}((p_{\text{SPO}}^{(k)})^2 q_{\text{SPO}}^{(k)}(1 - q_{\text{SPO}}^{(k)}))).$$

In addition, we have

$$q_{\text{SPO}}^{(k+1)} = \exp(f_{12}(p_{\text{SPO}}^{(k)}, q_{\text{SPO}}^{(k)})) = \exp(\log q_{\text{SPO}}^{(k)} + h_{q_{\text{SPO}}^{(k)}}((p_{\text{SPO}}^{(k)})^2 q_{\text{SPO}}^{(k)}(1 - q_{\text{SPO}}^{(k)}))).$$

Putting these pieces together yields $p_{\text{SPO}}^{(k+1)} > q_{\text{SPO}}^{(k+1)}$. □

C.3. Proof of Theorem 3.2

To show (i), recall that the sequence $(p_{\text{SPO}}^{(k)})_{k\in\mathbb{N}}$ is strictly increasing and bounded in $(0, 1)$ from Items C.4(i) and C.4(ii), so it converge to some value $c \in (0, 1]$. Take limit as $k \to \infty$:

$$1 = \lim_{k\to\infty} \frac{p_{\text{SPO}}^{(k+1)}}{p_{\text{SPO}}^{(k)}} = \lim_{k\to\infty} \frac{1}{(1-p_{\text{SPO}}^{(k)})e^{-\Delta_{\text{SPO},p}^{(k)}} + p_{\text{SPO}}^{(k)}} = \frac{1}{(1-c)e^{-c(1-c)}+c}.$$

Using the simple Taylor lower bound $e^{-x} \geq 1 - x$, we have

$$1 = \frac{1}{(1-c)e^{-c(1-c)}+c} \geq \frac{1}{(1-c)(1-c(1-c))+c} \implies (c-1)^2 \leq 0 \implies c = 1.$$

This shows $p_{\text{SPO}}^{(k)} \to 1$ as $k \to \infty$. Similarly, we can show $q_{\text{GRPO}}^{(k)}, p_{\text{SPO}}^{(k)}, q_{\text{SPO}}^{(k)} \to 1$ as $k \to \infty$.

To show (ii), consider the base case:

$$\begin{aligned} p_{\text{SPO}}^{(1)} &= \exp(f_{11}(p_{\text{SPO}}^{(0)})) = \exp(\log p_{\text{SPO}}^{(0)} + h_{p_{\text{SPO}}^{(0)}}(p_{\text{SPO}}^{(0)}(1 - p_{\text{SPO}}^{(0)}))) \\ &= \exp(\log p_{\text{GRPO}}^{(0)} + h_{p_{\text{GRPO}}^{(0)}}(p_{\text{GRPO}}^{(0)}(1 - p_{\text{GRPO}}^{(0)}))) \\ &> \exp(\log p_{\text{GRPO}}^{(0)} + h_{p_{\text{GRPO}}^{(0)}}(p_{\text{GRPO}}^{(0)}(1 - p_{\text{GRPO}}^{(0)})q_{\text{GRPO}}^{(0)})) = \exp(f_{21}(p_{\text{GRPO}}^{(0)}, q_{\text{GRPO}}^{(0)})) = p_{\text{GRPO}}^{(1)}, \end{aligned}$$

where the inequality follows from Item C.1(ii). Thus, we use induction and assume $p_{\text{SPO}}^{(k)} > p_{\text{GRPO}}^{(k)}$ for some $k \geq 1$. Then we have,

$$\begin{aligned} p_{\text{SPO}}^{(k+1)} &= \exp(f_{11}(p_{\text{SPO}}^{(k)})) > \exp(f_{11}(p_{\text{GRPO}}^{(k)})) = \exp(\log p_{\text{GRPO}}^{(k)} + h_{p_{\text{GRPO}}^{(k)}}(p_{\text{GRPO}}^{(k)}(1 - p_{\text{GRPO}}^{(k)}))) \\ &> \exp(\log p_{\text{GRPO}}^{(k)} + h_{p_{\text{GRPO}}^{(k)}}(p_{\text{GRPO}}^{(k)}(1 - p_{\text{GRPO}}^{(k)})q_{\text{GRPO}}^{(k)})) = \exp(f_{21}(p_{\text{GRPO}}^{(k)}, q_{\text{GRPO}}^{(k)})) = p_{\text{GRPO}}^{(k+1)}, \end{aligned}$$

where the first inequality uses Item C.1(i) and the second one uses Item C.1(ii). Thus, $p_{\texttt{SPO}}^{(k+1)} > p_{\texttt{GRPO}}^{(k+1)}$ and induction completes. We have proved that $p_{\texttt{SPO}}^{(k)} > p_{\texttt{GRPO}}^{(k)}$ for all $k \geq 1$.

To show (iii), first notice that we can show $p_{\texttt{GRPO}}^{(k)} = q_{\texttt{GRPO}}^{(k)}$ for all $k \geq 0$ by induction. The base case is trivial by initialization. Suppose $p_{\texttt{GRPO}}^{(k)} = q_{\texttt{GRPO}}^{(k)}$ for some $k \geq 0$, then by noticing that $f_{21}(p, p) = f_{22}(p, p)$, we have

$$\begin{aligned} p_{\texttt{GRPO}}^{(k+1)} &= \exp(f_{21}(p_{\texttt{GRPO}}^{(k)}, q_{\texttt{GRPO}}^{(k)})) = \exp(f_{21}(p_{\texttt{GRPO}}^{(k)}, p_{\texttt{GRPO}}^{(k)})) \\ &= \exp(f_{22}(p_{\texttt{GRPO}}^{(k)}, p_{\texttt{GRPO}}^{(k)})) = \exp(f_{22}(p_{\texttt{GRPO}}^{(k)}, q_{\texttt{GRPO}}^{(k)})) = q_{\texttt{GRPO}}^{(k+1)}. \end{aligned}$$

Thus, by induction, $p_{\texttt{GRPO}}^{(k)} = q_{\texttt{GRPO}}^{(k)}$ for all $k \geq 0$. Now, we can reduce the update rule of $p_{\texttt{GRPO}}^{(k)}$ as

$$p_{\texttt{GRPO}}^{(k+1)} = \frac{1}{(1/p_{\texttt{GRPO}}^{(k)} - 1)\exp(-(p_{\texttt{GRPO}}^{(k)})^2(1-p_{\texttt{GRPO}}^{(k)}))+1}.$$

Also recall the update rule of $p_{\texttt{SPO}}^{(k)}$ and $q_{\texttt{SPO}}^{(k)}$:

$$\begin{aligned} p_{\texttt{SPO}}^{(k+1)} &= \frac{1}{(1/p_{\texttt{SPO}}^{(k)} - 1)\exp(-p_{\texttt{SPO}}^{(k)}(1-p_{\texttt{SPO}}^{(k)}))+1} \\ q_{\texttt{SPO}}^{(k+1)} &= \frac{1}{(1/q_{\texttt{SPO}}^{(k)} - 1)\exp(-(p_{\texttt{SPO}}^{(k)})^2 q_{\texttt{SPO}}^{(k)}(1-q_{\texttt{SPO}}^{(k)}))+1} \end{aligned},$$

and it suffices to show $p_{\texttt{SPO}}^{(k)} q_{\texttt{SPO}}^{(k)} > (p_{\texttt{GRPO}}^{(k)})^2$ for all $k \geq 1$. We prove by induction. For the base case,

$$\sqrt{p_{\texttt{SPO}}^{(1)} q_{\texttt{SPO}}^{(1)}} = \sqrt{\frac{1}{1+(1/2)e^{-1/4}} \cdot \frac{1}{1+(1/2)e^{-1/16}}} > \frac{1}{1+(1/2)e^{-1/8}} = p_{\texttt{GRPO}}^{(1)}.$$

The above inequality holds true since Item C.1(iv) implies

$$2\log(1 + (1/2)e^{-1/8}) > \log(1 + (1/2)e^{-1/4}) + \log(1 + (1/2)e^{-1/16}),$$

It remains to show that $p_{\texttt{SPO}}^{(k)} q_{\texttt{SPO}}^{(k)} > (p_{\texttt{GRPO}}^{(k)})^2$ implies $p_{\texttt{SPO}}^{(k+1)} q_{\texttt{SPO}}^{(k+1)} > (p_{\texttt{GRPO}}^{(k+1)})^2$ for k large enough. By Item C.4(iii), we know $p_{\texttt{SPO}}^{(k)} > q_{\texttt{SPO}}^{(k)}$ for all $k \geq 1$. Thus, we can use Lemma C.3 to conclude that once $q_{\texttt{SPO}}^{(k)} > 31/32$, we will have

$$p_{\texttt{SPO}}^{(k+1)} q_{\texttt{SPO}}^{(k+1)} = \frac{1}{A(p_{\texttt{SPO}}^{(k)}) B(p_{\texttt{SPO}}^{(k)}, q_{\texttt{SPO}}^{(k)})} > \frac{1}{C(p_{\texttt{SPO}}^{(k)}, q_{\texttt{SPO}}^{(k)})^2}.$$

This is guaranteed to happen for large enough k because we have proved $q_{\texttt{SPO}}^{(k)} \to 1$ as $k \to \infty$. Using Item C.1(iii), we complete the induction by applying our induction hypothesis:

$$\frac{1}{C(p_{\texttt{SPO}}^{(k)}, q_{\texttt{SPO}}^{(k)})^2} = \exp(f_{21}(\sqrt{p_{\texttt{SPO}}^{(k)} q_{\texttt{SPO}}^{(k)}}, \sqrt{p_{\texttt{SPO}}^{(k)} q_{\texttt{SPO}}^{(k)}})) > \exp(f_{21}(p_{\texttt{GRPO}}^{(k)}, p_{\texttt{GRPO}}^{(k)})) = (p_{\texttt{GRPO}}^{(k+1)})^2.$$

This completes the proof. □

C.4. Empirical evaluations

We run some simple simulations to compare SPO to GRPO using our stylized model and plot the generated learning curves in Figure 2. In particular, the left figure displays the likelihood of learning a "good" action in the first step for each iteration k (i.e., $p^{(k)}_{\texttt{SPO}}$ v.s. $p^{(k)}_{\texttt{GRPO}}$), whereas the right figure displays the likelihood of learning the optimal policy, (i.e., $p^{(k)}_{\texttt{SPO}}q^{(k)}_{\texttt{SPO}}$ v.s. $p^{(k)}_{\texttt{GRPO}}q^{(k)}_{\texttt{GRPO}}$). We find that the empirical results reflect the qualitative behavior predicted by Theorem 3.2. The key difference is that the likelihood of learning the optimal policy for SPO consistently exceeds that for GRPO throughout the whole training, whereas Theorem 3.2 only guarantees such result in a local sense: SPO eventually surpasses GRPO without necessarily dominating at a few early iterations.

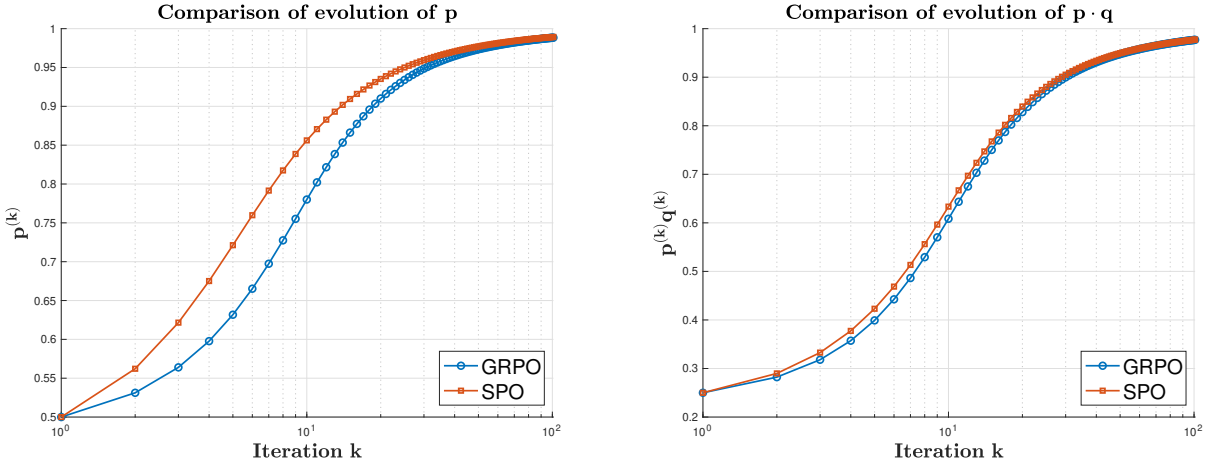

Figure 2: Evolution of SPO and GRPO algorithms for the stylized model

D. Additional Experimental Results

Experiments results. We present two additional experiment results for comparing SPO to GRPO on a weaker model in Table 4 and verifying robustness on multiple independent runs in Table 5.

Table 4: Further contrast between GRPO and SPO on weaker Qwen2.5-7B results.

	AMC23 avg@16	AIME24 avg@16	MATH500 pass@1	Olympiads pass@1
Qwen2.5-7B-Instruct				
GRPO	50.62	14.58	75.20	38.07
SPO	55.00	12.50	76.40	40.00

Table 5: Extra experiment results on DeepSeek-R1-Distill-Llama-8B across multiple independent runs. The mean average score and standard deviation is reported below.

	Kaoyan pass@1	GradeMath pass@1	MATH500 pass@1	Olympiads pass@1	CHMath24 avg@16	AIME25 avg@16	AIME24 avg@16	Gaokao avg@16	AMC23 avg@16
DeepSeek-R1-Distill-Llama-8B									
SPO	40.78±3.05	27.47±1.45	82.87±0.64	47.16±1.33	56.88±1.85	25.62±0.91	39.37±1.88	71.13±0.38	88.18±0.80

Prompt template. To better illustrate the process supervision mechanism, we present the following example prompt template, which identifies the first incorrect sentence using `char`-level indexing. An alternative approach is to return the full first incorrect sentence and match it against the model's response to compute the reward.

Example of RLAIF Feedback Prompt

Your task is to carefully read the agent's solution and identify the first sentence where a clear mistake occurs that deviates from the correct reasoning.

We will first provide you with a reference solution. Then, we will provide the agent's solution (the agent's solution is truncated at the maximum output length; just check the portion in the truncated portion). Please read the reference solution to understand how to approach this problem, and then check agent's solution.

Please go through the agent's solution sentence by sentence. For each sentence:

- If the sentence is correct, explain why it is correct.
- If the sentence shows a clear mistake that leads the reasoning away from the correct solution, mark it as the first incorrect sentence.

Here's the question:
{question}

Here's the reference answer:
{reference answer}

Here's the agent's solution:
{agent's response}

Please following the steps to reason agent's solution

1. 1) Check each sentence one by one
2. 1) For each correct sentence, explain why it's correct.
3. 2) Identify the first incorrect sentence with mistakes, explain why it leads to the wrong final answer and moving to the correct portion calculation

After identifying the first incorrect sentence:

1. 1) Count how many chars are there in the agent's answer
2. 2) Calculate the portion of correct characters in the reasoning: correct_portion = (position of first incorrect sentence's character)/(total character in the response)
3. 3) Output in this EXACT format: \Portion{{correct_portion}}

For example, if agent makes a mistake in the 2nd sentence, starting at character 232, and there are 19232 total characters, then: correct_portion = 232/19232 = 0.0121
Output should be: \Portion{{0.0121}}

THE OUTPUT MUST INCLUDE \Portion{} WITH THE CALCULATED correct_portion VALUE.