

STEPWISE GUIDED POLICY OPTIMIZATION: COLORING YOUR INCORRECT REASONING IN GRPO

Anonymous authors

Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has proven effective in strengthening the reasoning capabilities of large language models (LLMs). A widely adopted method, Group Relative Policy Optimization (GRPO) (Shao et al., 2024), has shown strong empirical results in training DeepSeek-R1 (Guo et al., 2025a). However, GRPO fails to update the policy when all responses within a group are incorrect (i.e., *all-negative-sample* groups). This limitation underscores a key gap between artificial and human intelligence: unlike humans, who can learn from mistakes, GRPO discards these signals. Our first contribution is to introduce a simple framework that mitigates the all-negative-sample issue by incorporating response diversity within groups using a *step-wise* judge model, which can be either directly trained or adapted from existing LLMs. We prove that this diversification can accelerate GRPO’s learning dynamics in a simplified setting. We also empirically validate the proposed stepwise guided policy optimization (SGPO) method, demonstrating consistent gains across model sizes (7B, 14B, 32B) in offline and online training on 9 benchmarks, including base and distilled variants. Our results highlight two advantages: (i) SGPO surpasses GRPO, especially in the early and mid-training stages where all-negative-sample groups are prevalent; and (ii) SGPO does not require judge models to generate correct answers, differentiating it from knowledge distillation methods.

1 INTRODUCTION

The 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*. Unlike instruction-tuned models (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023; Achiam et al., 2023), which produce quick responses by statistically inferring the next token, these new reasoning models deliberately decompose complex prompts (e.g., mathematical problems) into intermediate steps and work through chain-of-thought reasoning (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024; Xiang et al., 2025). This slower yet more rigorous process yields greater accuracy and makes them more human-like, enabling success on more complex and challenging tasks (Yang et al., 2018; Shi et al., 2024; Jain et al., 2025). As the generative AI applications move beyond simple conversational interfaces, these reasoning models are poised to become increasingly powerful and widely adopted, positioning them as a key frontier in practice.

At the heart of this revolution lies post-training with outcome-based 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), together with reinforcement learning (RL) methods (Schulman et al., 2015; 2017; Li et al., 2024b; Ahmadian et al., 2024; Shao et al., 2024; Xiong et al., 2025a), appreciated for their simplicity, intuitiveness, and practicality. A leading approach is proximal policy optimization (PPO) (Schulman et al., 2017), which relies on a critic (or value) model to estimate advantages. While essential in general RL tasks, this critic is often unnecessary in large language models (LLMs) due to their deterministic transition dynamics (Li et al., 2024b). This observation has inspired alternatives such as group relative policy optimization (GRPO) (Shao et al., 2024) and its extensions (Yu et al., 2025b; Liu et al., 2025b; Chu et al., 2025; Zhang et al., 2025a), which estimate advantages directly in a group-relative fashion.

A major limitation of these methods arises when all sampled responses in a group are incorrect (i.e., *all-negative-sample* groups), which eliminates the learning signal and halts policy updates. In GRPO, given a prompt $\mathbf{x}$, responses $\{y_i\}_{i=1}^G$ are drawn from the old policy π_{old} and assigned rewards $\{r_i\}_{i=1}^G$, where $r_i = 1$ if y_i is correct and 0 otherwise. Advantages are obtained by normalizing r_i within the group. If $r_i = 0$ for all i , the advantage vanishes, yielding no update. Such groups are frequent in early and mid-stages of training, when reasoning ability is weak¹. This shortcoming highlights a gap between artificial and human intelligence: humans effectively learn from mistakes, which act as essential signals during cognitive development (Chialvo & Bak, 1999). In mathematical reasoning, all-negative-sample groups prompt a child to revise rules and strengthen reasoning ability.

Recent studies suggest that negative samples in RL-based large reasoning model training carry more nuanced value than previously assumed (Xiong et al., 2025a). Instead of treating negative samples uniformly, they advocate for principled mechanisms to distinguish negative samples. One prominent direction is process reward models (PRMs) (Lightman et al., 2024; Wang et al., 2024; Luo et al., 2024; Setlur et al., 2025; Zhang et al., 2025b), which estimate either the probability of final success or its change after each reasoning step. However, their reliance on speculative value functions makes them prone to reward hacking (Skalse et al., 2022).

The key insight is that many reasoning tasks possess a structure where step-level correctness can be explicitly defined. This motivates the use of a step-wise judge model that evaluates trajectories by labeling each step as correct (1) or incorrect (0). Such a model can be trained directly (Xiong et al., 2025b) or adapted from existing LLMs (Zha et al., 2025; He et al., 2025)². By grounding rewards in step-level correctness rather than speculative value estimates, our method mitigates reward hacking and yields clearer signals. Intuitively, this allows negative samples to be differentiated through their trajectories: while early-stage reasoning trajectories are of low-quality, these remain informative – much like partial credit in education, where intermediate steps still guide learning.

Our approach enables a holistic evaluation of multi-step reasoning by transforming negative samples from binary outcome rewards into graded, step-level rewards. Consider a negative sample with five reasoning steps $(a_1, a_2, a_3, a_4, a_5)$. If the first error occurs at a_3 , then a_1 and a_2 are correct, yielding a correctness proportion of $\frac{2}{5}$. To improve reliability, we adopt a Grok4-Heavy -inspired strategy where multiple independent judgments are obtained from the judge model, and the error position is determined by the majority vote. We further introduce two scaling parameters β and γ to downweight noisy or unreliable signals (see Eq. (2)). Unlike PRMs, our approach avoids memory overhead and does not require costly step-level human annotations, thereby accelerating training.

Contribution. We propose and analyze a simple and efficient framework that introduces response diversity within all-negative-sample groups. It is both theoretically grounded in the simplified setting and empirically effective on various models, distinguishing our approach from existing heuristics. Our contributions can be summarized as follows:

1. We propose a *Stepwise Guided Policy Optimization* (SGPO) framework that leverages a step-wise judge model that identifies the first incorrect step that causes the trajectory to deviate from correctness. This makes evaluation computationally tractable and reliable. *It is important to emphasize that our contribution lies not in designing effective judge models, but in introducing a framework that leverages step-wise judges to effectively distinguish negative samples.* We also prove that SGPO outperforms GRPO in a simplified setting.
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 settings with nine benchmarks, including base and distilled variants. Our results reveal two key benefits: (i) SGPO delivers improvements beyond the reach of GRPO, especially in the early and mid-stages of training where all-negative-sample groups are common; (ii) SGPO does not rely on more powerful judge models generating correct answers, allowing it to be distinguish from knowledge distillation methods.

¹To reduce computational cost, training often uses small group sizes and short rollouts, further increasing the likelihood of all-negative-sample groups.

²We do not have access to their judge models as it’s not publicly released, so we adapt our own from existing LLMs.

The additional overhead from all-negative-sample groups remains modest, since the correctness can be efficiently verified against reference solutions, enabling rapid assessment of reasoning steps. As the computational and financial costs of closed-source judge models (o4-mini, Claude3.7) rise, SGPO accelerates learning dynamics, making the trade-off worthwhile. SGPO also outperforms GRPO with less powerful and more affordable open-source judge models (DeepSeek-V3-0324, Qwen3-235B-A22B, QwQ-32B), confirming that SGPO remains effective even without cutting-edge LLMs and underscoring its practicality in lower-resource settings.

2 PRELIMINARIES AND TECHNICAL BACKGROUND

Modern LLMs are built based on the Transformer architecture (Vaswani et al., 2017) and generate responses $\mathbf{y} = (a_1, \dots, a_H)$ to user prompts $\mathbf{x}$, where each token $a_h \in \mathcal{V}^*$, with $\mathcal{V}$ denoting the vocabulary and $\mathcal{V}^*$ the set of all possible token sequences. We view the LLM as a policy $\pi_\theta(\mathbf{y}|\mathbf{x})$ parameterized by θ , assigning probabilities to responses $\mathbf{y}$ given $\mathbf{x}$. The policy operates in an auto-regressive way as follows:

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

For a prompt $\mathbf{x}$ with ground-truth response $\mathbf{y}_x^*$, performance is evaluated using a regular-expression match on the final answer: $r(\mathbf{x}, \mathbf{y}) = 1$ if $\mathbf{y}$ matches $\mathbf{y}_x^*$ and $r(\mathbf{x}, \mathbf{y}) = 0$ otherwise (Hendrycks et al., 2021). We consider the reasoning tasks defined over a dataset $\mathcal{D} = (\mathbf{x}, \mathbf{y}_x^*)$, where each $\mathbf{x}$ is a problem and $\mathbf{y}_x^*$ its ground-truth solution.

The policy gradient methods (Williams, 1992; Sutton & Barto, 1998) aim to maximize the objective $J(\theta) = \mathbb{E}_{\mathbf{x} \sim \rho, \mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x}, \mathbf{y})]$ where ρ is the prompt distribution and π_θ is an LLM policy. Parameters are updated via $\theta \leftarrow \theta + \eta \nabla_\theta J(\theta)$. In practice, trajectories are sampled from an old policy $\pi_{\theta_{\text{old}}}$, which is different from π_θ , motivating the use of importance sampling 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, this estimator suffers from high variance when π_θ deviates from $\pi_{\theta_{\text{old}}}$. To stabilize training, clipped surrogate objectives are used as follows:

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

The group relative policy optimization (GRPO) and its variants (Yu et al., 2025b; Liu et al., 2025b; Chu et al., 2025; Zhang et al., 2025a) adopt this framework but estimate gradients using groups of samples. For each prompt $\mathbf{x}$, GRPO samples responses $\mathbf{y}_1, \dots, \mathbf{y}_G$ from $\pi_{\theta_{\text{old}}}$ and maximizes the objective function in the form of

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

where $\epsilon \in (0, 1)$ and the advantage A_i is computed as

$$A_i = \frac{r(\mathbf{x}, \mathbf{y}_i) - \text{mean}(\{r(\mathbf{x}, \mathbf{y}_1), \dots, r(\mathbf{x}, \mathbf{y}_G)\})}{\text{std}(\{r(\mathbf{x}, \mathbf{y}_1), \dots, r(\mathbf{x}, \mathbf{y}_G)\})}, \quad (1)$$

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

Remark 2.1. When rewards are identical across all samples within a group, $A_i = 0$ and no update occurs. This is appropriate for all-positive groups but constitutes a critical limitation for all-negative groups, where GRPO fails to exploit mistakes as learning signals.

3 MAIN RESULTS

We propose the Stepwise Guided Policy Optimization (SGPO) framework, which employs the stepwise judge model to detect the first incorrect step that leads a trajectory away from correctness. In a simplified setting, we prove that SGPO consistently accelerates GRPO’s learning dynamics.

3.1 A STEP-WISE JUDGE MODEL

We propose a principled reward mechanism for negative samples, wherein the step-wise judge model differentiates between structurally sound but partially incorrect reasoning and entirely erroneous responses. This design is motivated by the intuition that an incorrect final answer does not invalidate the entire reasoning process. For instance, a model may follow a logically coherent sequence of steps yet make a minor error – such as an arithmetic slip – that leads to an incorrect conclusion. Treating such cases the same as fundamentally flawed or incoherent reasoning does not make sense. This refinement remains effective under constraints such as reduced output length, where a model may be unable to complete the full solution but still demonstrates a valid reasoning trajectory.

Our step-wise judge model can be either trained directly or adapted from existing LLMs. It evaluates responses sequentially, identifying the first substantive error – such as a computational slip or a logical fallacy – that causes the trajectory to deviate from correctness. To formalize this, we define the *Reasoning Trajectory Score* (RTS) for an incorrect response $\mathbf{y}$, denoted as $\text{RTS}(\mathbf{y}) \in [0, 1]$. The judge model checks each step in order, pinpoints the first error, and treats all preceding steps as the valid reasoning segment. $\text{RTS}(\mathbf{y})$ is then computed as the ratio of the valid segment length to the total trajectory length. For example, if $\mathbf{y}$ consists of five steps $(a_1, a_2, a_3, a_4, a_5)$ and the first error occurs at a_4 , then $\text{RTS}(\mathbf{y}) = \frac{3}{5}$, indicating that three steps of reasoning are correct before erroneous.

In our experiment, we adapt the judge model from existing LLMs, either closed-source (o4-mini, Claude3.7) or open-source (DeepSeek-V3-0324, Qwen3-235B-A22B, QwQ-32B). To enhance reliability and further reduce variance in the reward signal, we employ the following protocol: (i) alongside the candidate response, we provide a reference solution drawn from a supervised fine-tuning dataset with correct answers and reasoning trajectories, anchoring the intended solution path and enabling error localization; and (ii) we elicit step-wise evaluation rather than holistic evaluation. The judge model justifies correctness or flags an error sentence by sentence, identifies the first clear mistake, and then traces how this error propagates to the final incorrect conclusion.

Based on the reasoning trajectory score, we introduce a new outcome reward function:

$$r_{\text{SGPO}}(\mathbf{y}) = \begin{cases} 1, & \text{if the final answer of } \mathbf{y} \text{ is correct,} \\ \frac{1}{1 + \exp(\beta(\text{RTS}(\mathbf{y}) - \gamma))}, & \text{otherwise.} \end{cases} \quad (2)$$

where $\gamma > 0$ and $\beta > 0$ are two 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_{SGPO} can be directly incorporated into the advantage calculation in Eq. (1). As a consequence, we refer to SGPO as GRPO using Eq. (2).

Remark 3.1. Our approach differs from process reward models (PRMs) (e.g. Lightman et al., 2024). For a prompt $\mathbf{x}$ and a prefix of reasoning steps $(a_1, \dots, a_t)$, a PRM typically predicts either (i) a prefix-level value $V(\mathbf{x}, a_{1:t}) = \mathbb{P}(\text{final answer correct} \mid \mathbf{x}, a_{1:t})$, or (ii) a step-level progress signal such as $\Delta_t = V(\mathbf{x}, a_{1:t}) - V(\mathbf{x}, a_{1:t-1})$. In practice, PRMs are trained by supervised ranking of intermediate steps and are used to re-rank trajectories or shape training at the prefix level, acting as approximate value (or Q-value) functions over prefixes. In contrast, SGPO introduces a different way of producing and using feedback signals: (i) Policy-guided rollouts without search. All trajectories are sampled from the current policy, without PRM-guided exploration or trajectory alteration; (ii) Post-hoc first-error identification. A step-wise judge inspects the entire trajectory, pinpoints the earliest error relative to a reference solution, and converts this into a calibrated scalar reward $r_{\text{SGPO}}(\mathbf{y})$ via the reasoning trajectory score; (iii) Stable credit assignment in all-negative-sample groups. By locating the first definitive mistake only after observing the full trace, SGPO eliminates the look-ahead ambiguity and feedback loops inherent to PRM-guided search (Zhang et al., 2024a), while avoiding the need for the judge to solve the problem or approximate a value function.

Remark 3.2. Our approach differs from knowledge distillation (e.g. Kang et al., 2023; Gu et al., 2024). The student model trained via knowledge distillation inherits the judge model’s failure, since it only imitates the judge model’s outputs. For instance, consider the AIME problem: “The twelve letters $\{A, B, C, D, E, F, G, H, I, J, K, L\}$ are randomly grouped into six pairs. Each pair is ordered alphabetically to form a two-letter word, and the six words are listed alphabetically, such as AB, CD, EF, GH, IJ, KL . The probability that the last word listed contains G is $\frac{m}{n}$ with m, n coprime. Find $m + n$ ”. Neither the student model (DeepSeek-R1-Distill-Qwen-7B) nor the judge models (DeepSeek-V3-0324) can solve this problem within 16 rollouts. In contrast,

SGPO leverages the judge model to identify mistakes in the student’s reasoning, providing learning signals that go beyond imitation and enabling improvements unattainable by knowledge distillation.

3.2 ACCELERATING LEARNING DYNAMICS

We present a theoretical analysis to explain why SGPO outperforms GRPO. To this end, we consider a simplified setting with a reasoning horizon of $H = 2$, where each step admits two possible actions $a_h \in 1, 2$ for $h = 1, 2$. This configuration follows prior works (Dayan, 1991; Li et al., 2024b), where analogous examples were employed to validate theoretical insights. Without loss of generality, we assume a unique ground-truth response $\mathbf{y}_x^* = (2, 2)$ for the prompt $\mathbf{x}$. The algorithm iteratively updates the policy parameter θ using samples drawn from the current policy π_θ . For clarity, we restrict the sample space to $(1, 1), (2, 1), (2, 2)$, excluding $(1, 2)$ since a correct reasoning step is unlikely to, and should not, follow an incorrect precursor.

To illustrate the effect of SPO, we analyze the learning dynamics of SGPO and GRPO in this simplified setting. Under GRPO, the rewards are assigned as $r((2, 2)) = 1$ and $r((2, 1)) = r((1, 1)) = 0$, meaning that only selecting the “good” action 2 at both steps yields a positive reward. In contrast, SGPO assigns $r_{\text{SGPO}}((2, 2)) = 1$, $r_{\text{SGPO}}((2, 1)) = \frac{1}{2}$ and $r_{\text{SGPO}}((1, 1)) = 0$. The difference is that partial progress – choosing the “good” action 2 in the first step but failing at the second – receives no credit in GRPO yet proportional credit in SGPO. Here, $\frac{1}{2}$ is chosen for illustrative purposes to convey the qualitative behavior of the reward mechanism, while the exact values used in experiments are determined by Eq. (2).

In our analysis, we examine the population-level learning dynamics with $G = 2$, omitting clipping and importance sampling. Let $p_{\text{GRPO}}^{(k)}$ and $q_{\text{GRPO}}^{(k)}$ denote the probability of selecting the “good” action in the first step at iteration k under GRPO, and that of selecting the “good” action in the second step conditioned on a correct first step. Analogously, $p_{\text{SGPO}}^{(k)}$ and $q_{\text{SGPO}}^{(k)}$ denote the corresponding probabilities under SGPO. Our theoretical findings are summarized in the following theorem.

Theorem 3.3. Suppose that $p_{\text{GRPO}}^{(0)} = q_{\text{GRPO}}^{(0)} = p_{\text{SGPO}}^{(0)} = q_{\text{SGPO}}^{(0)} = \frac{1}{2}$ and $\eta = 1$ ³ for GRPO and SGPO. Then, we have that (i) GRPO and SGPO achieve successful learning: $p_{\text{GRPO}}^{(k)}, q_{\text{GRPO}}^{(k)}, p_{\text{SGPO}}^{(k)}, q_{\text{SGPO}}^{(k)} \rightarrow 1$ as $k \rightarrow +\infty$; (ii) SGPO outperforms GRPO in learning the “good” action in the first step: $p_{\text{SGPO}}^{(k)} > p_{\text{GRPO}}^{(k)}$ for all $k \geq 1$; (iii) SGPO outperforms GRPO in learning the optimal policy: $p_{\text{SGPO}}^{(k)} q_{\text{SGPO}}^{(k)} > p_{\text{GRPO}}^{(k)} q_{\text{GRPO}}^{(k)}$ for all $k \geq 1$.

Proof Sketch. For (i), we first show the sequence $(p_{\text{SGPO}}^{(k)})_{k \geq 1}$ is strictly increasing and bounded in $(0, 1)$ (see Lemmas C.4(i) and C.4(ii)). Thus, it converge to some value $c \in (0, 1]$. Take limit as $k \rightarrow \infty$ on both sides of the update rule, we obtain the only feasible solution $c = 1$. Similarly, we show $q_{\text{SGPO}}^{(k)}, p_{\text{GRPO}}^{(k)}, q_{\text{GRPO}}^{(k)} \rightarrow 1$ as $k \rightarrow \infty$. For (ii), we use induction. For the base case, we use the update rule and Lemma C.1(ii). Assume $p_{\text{SGPO}}^{(k)} > p_{\text{GRPO}}^{(k)}$ for some $k \geq 1$, we derive $p_{\text{SGPO}}^{(k+1)} > p_{\text{GRPO}}^{(k+1)}$ using Lemmas C.1(i) and C.1(ii). For (iii), we show $p_{\text{GRPO}}^{(k)} = q_{\text{GRPO}}^{(k)}$ for all $k \geq 1$ using induction. It suffices to show $p_{\text{SGPO}}^{(k)} q_{\text{SGPO}}^{(k)} > (p_{\text{GRPO}}^{(k)})^2$ for all $k \geq 1$. We prove using induction again. For the base case, we use Lemma C.1(iv). It remains to show that $p_{\text{SGPO}}^{(k)} q_{\text{SGPO}}^{(k)} > (p_{\text{GRPO}}^{(k)})^2$ implies $p_{\text{SGPO}}^{(k+1)} q_{\text{SGPO}}^{(k+1)} > (p_{\text{GRPO}}^{(k+1)})^2$. Note that Lemma C.4(iii) guarantees $p_{\text{SGPO}}^{(k)} > q_{\text{SGPO}}^{(k)}$ for all $k \geq 1$. Thus, Lemma C.2 implies that

$$p_{\text{SGPO}}^{(k+1)} q_{\text{SGPO}}^{(k+1)} = \frac{1}{A(p_{\text{SGPO}}^{(k)})B(p_{\text{SGPO}}^{(k)}, q_{\text{SGPO}}^{(k)})} > \frac{1}{C(\sqrt{p_{\text{SGPO}}^{(k)} q_{\text{SGPO}}^{(k)}})^2},$$

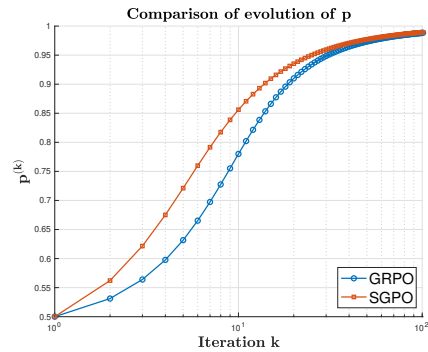


Figure 1: SGPO outperforms GRPO in learning a “good” action in the first step.

³We use the unit stepsize for simplicity. Our results are valid for any sufficiently small step size.

Table 1: Evaluation results on offline RL training. For each model, we report the baseline performance before RL training. We then report RL training results that uses only negative samples and positive samples, respectively. Performance across validation and training dataset (LIMO) is shown.

	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
Negative Samples only	61.88	15.21	80.40	42.37	30.11
Positive 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
Negative Samples only	69.53	20.42	83.00	46.37	36.47
Positive Samples only	66.87	18.75	83.60	47.41	41.86

where the functions $A(\cdot)$, $B(\cdot)$ and $C(\cdot)$ are defined in Lemma C.2. By using Lemma C.1(iii), we complete the induction by applying our induction hypothesis. $\square$

Remark 3.4. Theorem 3.3 presents one of the first theoretical analyses of GRPO with multiple samples and multi-step reasoning in the context of LLM reasoning. The first part establishes that SGPO converges to the optimal policy. The second and third parts demonstrate that SGPO both accelerates the acquisition of partially correct reasoning steps and preserves partial reasoning ability even when the final answer is incorrect. Importantly, the theorem provides a *per-iteration* comparison of learning under different reward mechanisms – an aspect rarely examined in previous works. The provable improvement in learning the optimal policy is also consistent with our numerical findings as partly shown in Figure 1; see Appendix C.4 for further details.

4 EXPERIMENTS

We demonstrate the benefits of differentiating negative samples through experiments in both offline and online settings. Offline RL is more computationally efficient, offering faster training, reduced memory consumption, and improved stability. In contrast, online RL provides greater flexibility and learning capacity, and has become the standard approach in large-scale reasoning models such as DeepSeek-R1 (Guo et al., 2025a).

4.1 OFFLINE TRAINING

For baselines, we consider strong models without further fine-tuned on math-specific SFT datasets, namely Qwen2.5-14B-Instruct and Qwen2.5-32B-Instruct. Prior work showed that a small set of carefully curated prompts significantly enhance the reasoning capability. Accordingly, we adopt the GAIR/LIMO dataset (Ye et al., 2025) as the training set, which has demonstrated strong potential for improving the reasoning performance of large-scale (32B) models in offline SFT. Evaluation is conducted on four standard math reasoning benchmarks: AIME24, AMC23, MATH500 (Hendrycks et al., 2021), and OlympiadBench (He et al., 2024). Our aim is to highlight the rich learning signal contained in all-negative-sample groups, showing that training exclusively on them can still yield performance gains. For benchmarks with fewer than 100 questions (AMC23, AIME24), we report avg@16 results with a decoding temperature of 0.6 and Top_P = 0.95. For benchmarks with more than 100 questions, we report pass@1 results using greedy decoding. Across all experiments, the maximum decoding length is set to 32768 tokens.

We conduct all response generation and model updates using offline RL (Peters & Schaal, 2007) with the standard GRPO mechanism. Specifically, the model is updated with advantages estimated from the offline dataset (see, e.g., Peng et al., 2019; Li et al., 2024b). For each prompt, we sample six responses per group and identify all-negative-sample groups in which all responses yield incorrect answers. Within these groups, we apply the step-wise judge model to assign differentiated rewards to negative samples, which are then used for offline RL updates. The model is trained for three epochs with a learning rate of 2×10^{-6} . As a contrastive baseline, we also perform offline RL using

Table 2: Evaluation results on online RL training. We refer to BASELINE as 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. For DeepSeek-R1-Distill-Qwen-7B, we report additional results, including (i) compatibility with more judge models and (ii) ablation on the stability parameters β and γ .

	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
SGPO+o4-MINI-0416	57.79	46.19	90.80	54.67	75.00	38.33	54.58	81.33	90.00	65.41
SGPO+DEEPSEEK-V3-0324	54.77	47.17	91.00	55.11	77.29	40.42	56.87	82.28	90.83	66.19
SGPO+QWEN3-235B-A22B	56.78	46.67	92.00	54.67	73.33	37.92	55.63	81.17	90.63	65.42
SGPO+QwQ-32B	52.26	45.24	92.00	53.78	75.00	35.21	56.46	82.28	91.88	64.91
SGPO+QwQ-32B w/o $\{\beta, \gamma\}$	58.29	42.38	90.20	55.11	74.58	38.69	53.63	81.24	88.75	65.08
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
SGPO+CLAUDE-3.7	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
SGPO+o4-MINI-0416	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
SGPO+o4-MINI-0416	48.24	53.81	83.00	46.81	29.79	14.58	19.58	45.09	69.53	45.06
QwQ-32B										
BASELINE	64.32	62.38	94.60	68.74	89.39	68.54	77.71	86.88	97.03	78.84
GRPO	71.36	63.81	94.60	69.48	88.75	64.38	75.83	87.11	97.03	79.15
SGPO+DEEPSEEK-V3-0324	73.37	64.76	95.00	70.22	88.33	66.46	78.33	87.11	97.97	80.17

only positive rollouts with correct answers. This parallel setup enables a direct comparison between learning from exclusively negative reasoning trajectories and from exclusively positive ones.

We conduct offline RL training to demonstrate that utilizing all-negative-sample groups can enhance the reasoning abilities of LLMs. For comparison, we also include positive-only offline RL training. As shown in Table 1, SGPO with negative samples consistently improve performance across most of benchmarks, in some cases even surpassing models trained solely on positive samples. Notably, in the 14B model experiment, training on negative samples yields improvements on four benchmarks relative to the positive-sample baseline. These findings underscore the utility of negative samples, which should not be discarded in online GRPO training; see more discussions in Section 4.4.

4.2 ONLINE TRAINING

For baselines, we consider applying Qwen2.5-14B-Instruct, Qwen2.5-32B-Instruct, QwQ-32B, 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 the step-wise judge model, we adopt a diverse set of LLMs, ranging from closed-source models with strong reasoning capabilities (o4-mini, Claude3.7) to open-source models that are more accessible to the community, including DeepSeek-V3-0324, Qwen3-235B-A22B, and QwQ-32B.

Compared to offline RL, online RL yields larger improvements in a model’s reasoning capabilities. Since our baselines already include strong distillation models, some benchmarks used in offline evaluation are nearing saturation. To provide a better assessment, we expand our evaluation suite beyond AMC23, AIME24, MATH500, and OlympiadBench by including AIME25, GradeSchool (Ye et al., 2025), CHMath24, Kaoyan, and Gaokao. Specifically, CHMath24 is the benchmark from the 2024 Chinese High School Mathematics League Competition, Gaokao from China’s 2024 National College Entrance Examination, Kaoyan from the Chinese Graduate School Entrance Examinations, and GradeSchool targets elementary-level mathematical reasoning. Among these, CHMath24 and Gaokao each contain fewer than 100 questions, for which we apply the temperature-based decoding for evaluation.

For GRPO training, we use the AIME collections from 1997 to 2023 provided in DeepScaler (Luo et al., 2025b), training for 12 epochs. All training questions are in English, while evaluation benchmarks include multilingual questions. Notably, negative samples learned during training generalize

Table 3: Evaluation results are reported for DeepSeek-R1-Distill-Qwen-7B across four independent runs. First column indicates judge models and its corresponding reward stability setup.

	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-SGPO										
+Qwen3-235B-A22B	53.90 $\pm$ 2.10	46.55 $\pm$ 0.24	91.30 $\pm$ 0.87	53.45 $\pm$ 1.35	74.48 $\pm$ 1.10	37.40 $\pm$ 1.05	55.73 $\pm$ 1.39	81.33 $\pm$ 0.18	90.19 $\pm$ 0.48	64.92 $\pm$ 0.37
+QwQ-32B	53.89 $\pm$ 1.66	44.88 $\pm$ 1.36	91.15 $\pm$ 0.84	53.71 $\pm$ 0.74	74.33 $\pm$ 1.29	37.03 $\pm$ 1.23	54.76 $\pm$ 1.87	81.91 $\pm$ 0.53	90.08 $\pm$ 1.23	64.64 $\pm$ 0.41
+QwQ-32B w/o $\{\beta, \gamma\}$	56.14 $\pm$ 2.76	44.53 $\pm$ 2.41	90.10 $\pm$ 0.66	53.64 $\pm$ 1.08	73.89 $\pm$ 1.13	38.70 $\pm$ 1.80	53.63 $\pm$ 2.15	81.24 $\pm$ 0.50	88.83 $\pm$ 0.81	64.52 $\pm$ 0.57

Table 4: Evaluation results are reported for DeepSeek-R1-Distill-Qwen-7B as base model and QwQ-32B as judge model with and without majority voting.

	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-SGPO										
+QwQ-32B	53.89 $\pm$ 1.66	44.88 $\pm$ 1.36	91.15 $\pm$ 0.84	53.71 $\pm$ 0.74	74.33 $\pm$ 1.29	37.03 $\pm$ 1.23	54.76 $\pm$ 1.87	81.91 $\pm$ 0.53	90.08 $\pm$ 1.23	64.64 $\pm$ 0.41
+QwQ-32B with voting	56.66 $\pm$ 1.66	45.24 $\pm$ 2.16	91.35 $\pm$ 0.50	53.82 $\pm$ 0.94	74.19 $\pm$ 0.92	37.35 $\pm$ 2.25	55.81 $\pm$ 0.99	82.12 $\pm$ 0.43	90.53 $\pm$ 1.07	65.23 $\pm$ 0.18

Table 5: Evaluation results are reported in terms of pass@16 across benchmarks. The first two columns show the total number of questions and the number solved within 16 attempts, while the last two columns report the number of unique questions solved by one method but not the other.

	SGPO - pass@16	GRPO - pass@16	SGPO \ GRPO	GRPO \ SGPO
AIME24	23/30	19/30	4	0
AIME25	21/30	21/30	1	1
Gaokao	70/79	68/79	2	0
AMC23	39/40	38/40	1	0
CHMath24	27/30	25/30	2	0

well to out-of-domain mathematical reasoning tasks. SGPO training follows the same setup. With batch-simultaneous processing, judge model calls take 90 seconds per batch of negatives, adding 10% wall-clock time relative to rollout and update. Step-wise supervision is applied only to all-negative-sample groups during the first three epochs, as we expect this duration to suffice for the model to internalize corrective signals; beyond this point, unresolved examples are more indicative of model capacity limits than learnability. For all models, rollout length is fixed at 8192 tokens and group size at 8. Models less than 8B are trained on 8 H100, 14B models on 16 H100, and 32B models on 32 H200. We adopt the default KL coefficient and learning rate from the `verl` training script (Sheng et al., 2025), and use the LIMO evaluation script (Ye et al., 2025), both of which are standard practices in the community.

A key insight from Table 2 is that stronger models generate higher-quality negative samples, which substantially aid learning. As model capability improves, so does the informativeness of its mistakes. Negative samples broadly fall into two categories: (i) correct reasoning trajectories truncated by output length limits, and (ii) trajectories containing logical errors. The first type remains highly valuable – yet discarded in GRPO – since it preserves meaningful reasoning steps, motivating our step-wise judge model. The second type, though incorrect, still provides informative signals, especially when all samples fail on genuinely challenging problems. Notably, stronger distilled models average 6K tokens per response, compared to only 1K tokens for weaker base models, making truncated but informative negative samples more common in the stronger case. Likewise, their erroneous responses also tend to be richer and more useful for step-level judgment.

4.3 OTHER ABLATION STUDIES

To assess reliability of judge models, we evaluate our approach not only with strong closed-source reasoning models but also with publicly available models of weaker capacity: DeepSeek-V3, Qwen3-235B and QwQ-32B. As shown in Table 2 (best-tuned results) and Table 3 (multiple runs with weaker judges), performance remains stable, indicating that weaker judges do not significantly degrade outcomes. We attribute this reliability to two design choices: (i) first-step error identification with a reference answer. SGPO requires the judge only to verify each step against the reference, not to solve the problem, thereby reducing task difficulty and avoiding the pitfalls of generic PRMs; (ii) reward stability parameters β and γ , which filter and smooth noisy signals. As confirmed by

ablations, removing β and γ increases variance and weakens performance. To improve verification, we incorporate a Grok4-Heavy-inspired strategy: multiple independent evaluations by the judge model, with the error position selected by majority voting. Using QwQ-32B as the judge model, DeepSeek-R1-Distill-Qwen-7B as the base model, and four rollouts per judgment, we have observed noticeable gains in consistency and stability (see Table 4).

While avg@16 measures average performance across rollouts, pass@16 reflects the ability to solve new questions with multiple attempts. As shown in Table 5, SGPO’s gains in pass@16 stem directly from leveraging negative samples. Learning only from solvable problems reinforces existing ability, whereas all-negative-sample groups correspond to genuinely difficult questions where GRPO consistently fails. These are precisely the cases where additional feedback can be most valuable. By providing step-level signals, SGPO rewards near-misses by reinforcing correct reasoning up to the first error, penalizes early failures by discouraging persistent error modes, and exposes blind spots by turning hard cases into informative training signals. In this regard, SGPO provides benefits that GRPO cannot match, covering more hard problems and providing sharper credit assignment, which translates to faster and more reliable learning under realistic compute budgets.

By leveraging richer early-stage signals from negative samples, SGPO can achieve faster and stronger performance than GRPO. As shown in Figure 2, SGPO continues improving beyond epoch 5 by solving several additional hard training problems, whereas GRPO plateaus. This improvement stems from informative negative samples that help resolve previously unsolved problems as also shown in Table 5. We also find that SGPO converges more rapidly to a deterministic policy, as illustrated by the training trajectories in Figure 3 (see Appendix B).

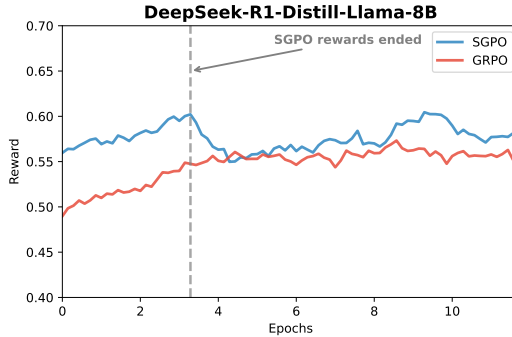


Figure 2: Evaluation results on GRPO and SGPO. SGPO rewards end at epoch 3.

4.4 DISCUSSIONS

We highlight the motivation for evaluating both offline and online RL. In the offline setup, training uses **only** negative samples, allowing us to directly test whether incorrect or incomplete reasoning trajectories can improve performance. In the online setup, we simulate realistic GRPO training, where batches contain a random mix of positive and negative samples. This demonstrates that negative samples are not only effective in isolation but also remain valuable in practical settings with noisier, mixed data. While mixing positives and negatives introduces noise, simply discarding negative samples does not stabilize training; in several cases, the performance of GRPO even drops below baseline, as the model overfits to problems it can already solve.

This instability arises from limited out-of-domain generalization and catastrophic forgetting. Without exposure to challenging or partially correct reasoning, the model risks overfitting to easy cases, reinforcing shallow heuristics instead of developing robust problem-solving skills. The absence of diverse failure cases can also cause catastrophic forgetting, degrading performance on previously solvable tasks. Incorporating negative samples mitigates these issues, as SGPO consistently outperforms GRPO on Chinese OOD math benchmarks. Nonetheless, further work is needed to design more stable training frameworks, including richer reward diversification mechanisms for handling negative samples and corresponding efficient RL methods beyond GRPO.

5 CONCLUSION

We propose a simple and efficient framework that introduces response diversity within all-negative-sample groups and prove, in a simplified setting, that such diversification can accelerate the learning dynamic of GRPO. Empirically, our approach can yield consistent improvements across model sizes in both offline and online training over nine benchmarks, including base and distilled variants. Future works include extending theoretical results to broader multi-step reasoning tasks, applying response diversity to accelerate other RL methods, and designing lightweight, task-specific reward models that evaluate reasoning steps correctly even if they cannot solve the full problem.

REFERENCES

- J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. GPT-4 technical report. *ArXiv Preprint: 2303.08774*, 2023. (Cited on page 1.)
- A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *COLT*, pp. 64–66, 2020. (Cited on page 21.)
- S. Agarwal, Z. Zhang, L. Yuan, J. Han, and H. Peng. The unreasonable effectiveness of entropy minimization in LLM reasoning. *ArXiv Preprint: 2505.15134*, 2025. (Cited on page 21.)
- A. Ahmadian, C. Cremer, M. Gallé, M. Fadaee, J. Kreutzer, O. Pietquin, A. Üstün, and S. Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *ACL*, pp. 12248–12267, 2024. (Cited on page 1.)
- D. Arora and A. Zanette. Training language models to reason efficiently. *ArXiv Preprint: 2502.04463*, 2025. (Cited on page 19.)
- M. G. Azar, Z. Guo, B. Piot, R. Munos, M. Rowland, M. Valko, and D. Calandriello. A general theoretical paradigm to understand learning from human preferences. In *AISTATS*, pp. 4447–4455, 2024. (Cited on page 20.)
- Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *ArXiv Preprint: 2212.08073*, 2022. (Cited on page 20.)
- M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI*, pp. 17682–17690, 2024. (Cited on pages 1 and 19.)
- X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu, et al. Deepseek LLM: Scaling open-source language models with longtermism. *ArXiv Preprint: 2401.02954*, 2024. (Cited on page 20.)
- T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *NeurIPS*, pp. 1877–1901, 2020. (Cited on page 1.)
- H. Chen, G. He, L. Yuan, G. Cui, H. Su, and J. Zhu. Noise contrastive alignment of language models with explicit rewards. In *NeurIPS*, pp. 117784–117812, 2024a. (Cited on page 20.)
- H. Chen, H. Zhao, H. Lam, D. Yao, and W. Tang. MallowsPO: Fine-tune your LLM with preference dispersions. In *ICLR*, 2025a. URL <https://openreview.net/forum?id=d8cnezVcaW>. (Cited on page 20.)
- P. Chen, X. Chen, W. Yin, and T. Lin. ComPO: Preference alignment via comparison oracles. *ArXiv Preprint: 2505.05465*, 2025b. (Cited on page 20.)
- Q. Chen, L. Qin, J. Liu, D. Peng, J. Guan, P. Wang, M. Hu, Y. Zhou, T. Gao, and W. Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *ArXiv Preprint: 2503.09567*, 2025c. (Cited on page 19.)
- X. Chen, J. Xu, T. Liang, Z. He, J. Pang, D. Yu, L. Song, Q. Liu, M. Zhou, Z. Zhang, et al. Do not think that much for $2+3=?$ on the overthinking of o1-like LLMs. *ArXiv Preprint: 2412.21187*, 2024b. (Cited on page 19.)
- J. Cheng and B. Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *ArXiv Preprint: 2412.13171*, 2024. (Cited on page 19.)
- D. R. Chialvo and P. Bak. Learning from mistakes. *Neuroscience*, 90(4):1137–1148, 1999. (Cited on page 2.)

- A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. (Cited on page 1.)
- P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, pp. 4302–4310, 2017. (Cited on page 20.)
- X. Chu, H. Huang, X. Zhang, F. Wei, and Y. Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. *ArXiv Preprint: 2504.02546*, 2025. (Cited on pages 1 and 3.)
- K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *ArXiv Preprint: 2110.14168*, 2021. (Cited on pages 1 and 20.)
- G. Cui, Y. Zhang, J. Chen, L. Yuan, Z. Wang, Y. Zuo, H. Li, Y. Fan, H. Chen, W. Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *ArXiv Preprint: 2505.22617*, 2025. (Cited on page 20.)
- P. Dayan. Reinforcement comparison. In *Connectionist Models*, pp. 45–51. Elsevier, 1991. (Cited on page 5.)
- B. Ding, C. Qin, L. Liu, Y. K. Chia, B. Li, S. Joty, and L. Bing. Is GPT-3 a good data annotator? In *ACL*, pp. 11173–11195, 2023. (Cited on page 20.)
- H. Dong, W. Xiong, D. Goyal, Y. Zhang, W. Chow, R. Pan, S. Diao, J. Zhang, K. Shum, and T. Zhang. RAFT: Reward ranked fine-tuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. URL <https://openreview.net/forum?id=m7p507zb1Y>. (Cited on page 20.)
- H. Dong, W. Xiong, B. Pang, H. Wang, H. Zhao, Y. Zhou, N. Jiang, D. Sahoo, C. Xiong, and T. Zhang. RLHF workflow: From reward modeling to online RLHF. *Transactions on Machine Learning Research*, 2024. URL <https://openreview.net/forum?id=a13aYU09eU>. (Cited on page 20.)
- K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela. Model alignment as prospect theoretic optimization. In *ICML*, pp. 12634–12651, 2024. (Cited on page 20.)
- G. Feng, B. Zhang, Y. Gu, H. Ye, D. He, and L. Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *NeurIPS*, pp. 70757–70798, 2023. (Cited on page 19.)
- K. Gandhi, D. Lee, G. Grand, M. Liu, W. Cheng, A. Sharma, and N. D. Goodman. Stream of search (SOS): Learning to search in language. *ArXiv Preprint: 2404.03683*, 2024. (Cited on page 19.)
- L. Gao, J. Schulman, and J. Hilton. Scaling laws for reward model overoptimization. In *ICML*, pp. 10835–10866, 2023. (Cited on page 19.)
- F. Gilardi, M. Alizadeh, and M. Kubli. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. (Cited on page 20.)
- Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, M. Huang, N. Duan, and W. Chen. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *ICLR*, 2024. URL <https://openreview.net/forum?id=Ep0TtjVoap>. (Cited on page 19.)
- Y. Gu, L. Dong, F. Wei, and M. Huang. MiniLLM: Knowledge distillation of large language models. In *ICLR*, 2024. URL <https://openreview.net/forum?id=5h0qf7IBZZ>. (Cited on page 4.)
- D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *ArXiv Preprint: 2501.12948*, 2025a. (Cited on pages 1, 6, 19, and 20.)

- J. Guo, Y. Wu, J. Qiu, K. Huang, X. Juan, L. Yang, and M. Wang. Temporal consistency for LLM reasoning process error identification. *ArXiv Preprint: 2503.14495*, 2025b. (Cited on page 19.)
- S. Hao, Y. Gu, H. Ma, J. Hong, Z. Wang, D. Wang, and Z. Hu. Reasoning with language model is planning with world model. In *EMNLP*, pp. 8154–8173, 2023. (Cited on page 19.)
- S. Hao, Y. Gu, H. Luo, T. Liu, X. Shao, X. Wang, S. Xie, H. Ma, A. Samavedhi, Q. Gao, Z. Wang, and Z. Hu. 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>. (Cited on page 19.)
- S. Hao, S. Sukhbaatar, D. Su, X. Li, Z. Hu, J. Weston, and Y. Tian. Training large language models to reason in a continuous latent space. *ArXiv Preprint: 2412.06769*, 2024b. (Cited on page 19.)
- A. Havrilla, Y. Du, S. C. Raparthy, C. Nalmpantis, J. Dwivedi-Yu, M. Zhuravinskyi, E. Hambro, S. Sukhbaatar, and R. Raileanu. Teaching large language models to reason with reinforcement learning. *ArXiv Preprint: 2403.04642*, 2024. (Cited on page 19.)
- C. He, R. Luo, Y. Bai, S. Hu, Z. Thai, J. Shen, J. Hu, X. Han, Y. Huang, Y. Zhang, et al. Olympiad-Bench: A challenging benchmark for promoting AGI with Olympiad-level bilingual multimodal scientific problems. In *ACL*, pp. 3828–3850, 2024. (Cited on page 6.)
- T. He, R. Mu, L. Liao, Y. Cao, M. Liu, and B. Qin. Good learners think their thinking: Generative PRM makes large reasoning model more efficient math learner. *ArXiv Preprint: 2507.23317*, 2025. (Cited on page 2.)
- D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks Track*, 2021. URL <https://openreview.net/forum?id=7Bywt2mQsCe>. (Cited on pages 3, 6, and 20.)
- J. Hong, N. Lee, and J. Thorne. ORPO: Monolithic preference optimization without reference model. In *EMNLP*, pp. 11170–11189, 2024. (Cited on page 20.)
- A. Hosseini, X. Yuan, N. Malkin, A. Courville, A. Sordoni, and R. Agarwal. V-Star: Training verifiers for self-taught reasoners. In *COLM*, 2024. URL <https://openreview.net/forum?id=stmqBSW2dV>. (Cited on pages 1 and 20.)
- J. Huang and K. C-C. Chang. Towards reasoning in large language models: A survey. In *ACL*, pp. 1049–1065, 2023. (Cited on page 19.)
- K. Huang, J. Guo, Z. Li, X. Ji, J. Ge, W. Li, Y. Guo, T. Cai, H. Yuan, R. Wang, Y. Wu, M. Yin, S. Tang, Y. Huang, C. Jin, X. Chen, C. Zhang, and M. Wang. MATH-Perturb: Benchmarking LLMs’ math reasoning abilities against hard perturbations. *ArXiv Preprint: 2502.06453*, 2025. (Cited on page 19.)
- S. Huang, Z. Ma, J. Du, C. Meng, W. Wang, and Z. Lin. Mirror-consistency: Harnessing inconsistency in majority voting. In *EMNLP*, pp. 2408–2420, 2024. (Cited on page 19.)
- H. Hwang, D. Kim, S. Kim, S. Ye, and M. Seo. Self-explore: Enhancing mathematical reasoning in language models with fine-grained rewards. In *EMNLP*, pp. 1444–1466, 2024. (Cited on page 20.)
- A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al. OpenAI o1 system card. *ArXiv Preprint: 2412.16720*, 2024. (Cited on pages 1 and 19.)
- N. Jain, K. Han, A. Gu, W-D. Li, F. Yan, T. Zhang, S. Wang, A. Solar-Lezama, K. Sen, and I. Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *ICLR*, 2025. URL <https://openreview.net/forum?id=chfJJYC3iL>. (Cited on page 1.)
- M. Kang, S. Lee, J. Baek, K. Kawaguchi, and S. J. Hwang. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. In *NeurIPS*, pp. 48573–48602, 2023. (Cited on page 4.)

- T. Khot, H. Trivedi, M. Finlayson, Y. Fu, K. Richardson, P. Clark, and A. Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. In *ICLR*, 2023. URL https://openreview.net/forum?id=_nGgzQjzaRy. (Cited on page 19.)
- M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh. Reward design with language models. In *ICLR*, 2023. URL <https://openreview.net/forum?id=10uNUgI5Kl>. (Cited on page 20.)
- A. K. Lampinen, I. Dasgupta, S. C. Y. Chan, H. R. Sheahan, A. Creswell, D. Kumaran, J. L. McClelland, and F. Hill. Language models, like humans, show content effects on reasoning tasks. *PNAS Nexus*, 3(7):pgae233, 2024. (Cited on page 19.)
- Y. LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022. (Cited on page 19.)
- H. Lee, S. Phatale, H. Mansoor, T. Mesnard, J. Ferret, K. Lu, C. Bishop, E. Hall, V. Carbune, A. Rastogi, et al. RLAIIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *ICML*, pp. 26874–26901, 2024. (Cited on page 20.)
- L. Lehnert, S. Sukhbaatar, D. Su, Q. Zheng, P. McVay, M. Rabbat, and Y. Tian. Beyond A*: Better planning with transformers via search dynamics bootstrapping. In *COLM*, 2024. URL <https://openreview.net/forum?id=SGoVIC0u0f>. (Cited on page 19.)
- Z. Li, H. Liu, D. Zhou, and T. Ma. Chain of thought empowers transformers to solve inherently serial problems. In *ICLR*, 2024a. URL <https://openreview.net/forum?id=3EWTEy9MTM>. (Cited on page 19.)
- Z. Li, T. Xu, Y. Zhang, Z. Lin, Y. Yu, R. Sun, and Z-Q. Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *ICML*, pp. 29128–29163, 2024b. (Cited on pages 1, 5, 6, and 21.)
- Z. Li, C. Chen, T. Xu, Z. Qin, J. Xiao, Z-Q. Luo, and R. Sun. Preserving diversity in supervised fine-tuning of large language models. In *ICLR*, 2025. URL <https://openreview.net/forum?id=NQEe7B7bSw>. (Cited on page 19.)
- H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let’s verify step by step. In *ICLR*, 2024. URL <https://openreview.net/forum?id=v8L0pN6EOi>. (Cited on pages 1, 2, 4, 19, and 20.)
- T. Liu, Y. Zhao, R. Joshi, M. Khalman, M. Saleh, P. J. Liu, and J. Liu. Statistical rejection sampling improves preference optimization. In *ICLR*, 2024a. URL <https://openreview.net/forum?id=xbjSwwrQOe>. (Cited on page 20.)
- T. Liu, Z. Qin, J. Wu, J. Shen, M. Khalman, R. Joshi, Y. Zhao, M. Saleh, S. Baumgartner, J. Liu, et al. LiPO: Listwise preference optimization through learning-to-rank. In *NAACL*, pp. To appear, 2025a. (Cited on page 20.)
- Z. Liu, M. Lu, S. Zhang, B. Liu, H. Guo, Y. Yang, J. Blanchet, and Z. Wang. Provably mitigating overoptimization in RLHF: Your SFT loss is implicitly an adversarial regularizer. In *NeurIPS*, pp. 138663–138697, 2024b. (Cited on page 20.)
- Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin. Understanding R1-zero-like training: A critical perspective. *ArXiv Preprint: 2503.20783*, 2025b. (Cited on pages 1 and 3.)
- H. Luo, L. Shen, H. He, Y. Wang, S. Liu, W. Li, N. Tan, X. Cao, and D. Tao. o1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *ArXiv Preprint: 2501.12570*, 2025a. (Cited on page 19.)
- L. Luo, Y. Liu, R. Liu, S. Phatale, M. Guo, H. Lara, Y. Li, L. Shu, Y. Zhu, L. Meng, et al. Improve mathematical reasoning in language models by automated process supervision. *ArXiv Preprint: 2406.06592*, 2024. (Cited on pages 2 and 20.)

- M. Luo, S. Tan, J. Wong, X. Shi, W. Y. Tang, M. Roongta, C. Cai, J. Luo, L. E. Li, R. A. Popa, and I. Stoica. DeepScaleR: Surpassing ol-preview with a 1.5B model by scaling RL. <https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-Ol-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005b> 2025b. Notion Blog. (Cited on page 7.)
- A. Madaan, K. Hermann, and A. Yazdanbakhsh. What makes chain-of-thought prompting effective? a counterfactual study. In *EMNLP*, pp. 1448–1535, 2023. (Cited on page 19.)
- J. Mei, Y. Gao, B. Dai, C. Szepesvari, and D. Schuurmans. Leveraging non-uniformity in first-order non-convex optimization. In *ICML*, pp. 7555–7564, 2021. (Cited on page 21.)
- Y. Meng, M. Xia, and D. Chen. SimPO: Simple preference optimization with a reference-free reward. In *NeurIPS*, pp. 124198–124235, 2024. (Cited on page 20.)
- W. Merrill and A. Sabharwal. The expressive power of transformers with chain of thought. In *ICLR*, 2024. URL <https://openreview.net/forum?id=NjNGLPh8Wh>. (Cited on page 19.)
- L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, pp. 27730–27744, 2022. (Cited on page 20.)
- A. Pal, D. Karkhanis, S. Dooley, M. Roberts, S. Naidu, and C. White. Smaug: Fixing failure modes of preference optimisation with DPO-positive. *ArXiv Preprint: 2402.13228*, 2024. (Cited on page 20.)
- R. Y. Pang, W. Yuan, H. He, K. Cho, S. Sukhbaatar, and J. Weston. Iterative reasoning preference optimization. In *NeurIPS*, pp. 116617–116637, 2024. (Cited on page 20.)
- R. Park, R. Rafailov, S. Ermon, and C. Finn. Disentangling length from quality in direct preference optimization. In *ACL*, pp. 4998–5017, 2024. (Cited on page 20.)
- X. Peng, A. Kumar, G. Zhang, and S. Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *ArXiv Preprint: 1910.00177*, 2019. (Cited on page 6.)
- J. Peters and S. Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *ICML*, pp. 745–750, 2007. (Cited on page 6.)
- R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, pp. 53728–53741, 2023. (Cited on page 19.)
- R. Rafailov, J. Hejna, R. Park, and C. Finn. From $\$r\$$ to $\$q^*\$$: Your language model is secretly a Q-function. In *COLM*, 2024. URL <https://openreview.net/forum?id=kEVcNxtqXk>. (Cited on page 20.)
- N. Razin, S. Malladi, A. Bhaskar, D. Chen, S. Arora, and B. Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *ICLR*, 2025. URL <https://openreview.net/forum?id=uaMSBJDnRv>. (Cited on page 20.)
- P. Roit, J. Ferret, L. Shani, R. Aharoni, G. Cideron, R. Dadashi, M. Geist, S. Girgin, L. Hussenot, O. Keller, et al. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *ACL*, pp. 6252–6272, 2023. (Cited on page 20.)
- J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. In *ICML*, pp. 1889–1897, 2015. (Cited on page 1.)
- J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *ArXiv Preprint: 1707.06347*, 2017. (Cited on page 1.)
- A. Setlur, C. Nagpal, A. Fisch, X. Geng, J. Eisenstein, R. Agarwal, A. Agarwal, J. Berant, and A. Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *ICLR*, 2025. URL <https://openreview.net/forum?id=A6Y7AqlzLW>. (Cited on pages 1, 2, and 20.)

- Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. RL on incorrect synthetic data scales the efficiency of LLM math reasoning by eight-fold. In *NeurIPS*, pp. 43000–43031, 2024. (Cited on page 20.)
- Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *ArXiv Preprint: 2402.03300*, 2024. (Cited on pages 1 and 19.)
- G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu. HybridFlow: A flexible and efficient RLHF framework. In *EuroSys*, pp. 1279–1297. ACM, 2025. (Cited on pages 7 and 8.)
- B. Shi, M. Tang, K. R. Narasimhan, and S. Yao. Can language models solve Olympiad programming? In *COLM*, 2024. URL <https://openreview.net/forum?id=kGa4fMtP9l>. (Cited on page 1.)
- D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. (Cited on page 19.)
- A. Singh, J. D. Co-Reyes, R. Agarwal, A. Anand, P. Patil, X. Garcia, P. J. Liu, J. Harrison, J. Lee, K. Xu, A. T. Parisi, A. Kumar, A. A. Alemi, A. Rizkowsky, A. Nova, B. Adlam, B. Bohnet, G. F. Elsayed, H. Sedghi, I. Mordatch, I. Simpson, I. Gur, J. Snoek, J. Pennington, J. Hron, K. Kenealy, K. Swersky, K. Mahajan, L. A. Culp, L. Xiao, M. Bileschi, N. Constant, R. Novak, R. Liu, T. Warkentin, Y. Bansal, E. Dyer, B. Neyshabur, J. Sohl-Dickstein, and N. Fiedel. 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=1NAyUngGFK>. Expert Certification. (Cited on page 20.)
- I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg. ProgPrompt: Generating situated robot task plans using large language models. In *ICRA*, pp. 11523–11530. IEEE, 2023. (Cited on page 1.)
- J. Skalse, N. H. R. Howe, D. Krashenninnikov, and D. Krueger. Defining and characterizing reward hacking. In *NeurIPS*, pp. 9460–9471, 2022. (Cited on page 2.)
- C. V. Snell, J. Lee, K. Xu, and A. Kumar. 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>. (Cited on pages 19 and 20.)
- F. Song, B. Yu, M. Li, H. Yu, F. Huang, Y. Li, and H. Wang. Preference ranking optimization for human alignment. In *AAAI*, pp. 18990–18998, 2024. (Cited on page 20.)
- S. Srivastava, A. PV, S. Menon, A. Sukumar, A. Philipose, S. Prince, S. Thomas, et al. Functional benchmarks for robust evaluation of reasoning performance, and the reasoning gap. *ArXiv Preprint: 2402.19450*, 2024. (Cited on page 19.)
- N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. In *NeurIPS*, pp. 3008–3021, 2020. (Cited on page 20.)
- D. Su, S. Sukhbaatar, M. Rabbat, Y. Tian, and Q. Zheng. Dualformer: Controllable fast and slow thinking by learning with randomized reasoning traces. In *ICLR*, 2025. URL <https://openreview.net/forum?id=bmbRCRiNDu>. (Cited on page 19.)
- R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*, volume 1. MIT Press, 1998. (Cited on page 3.)
- F. Tajwar, A. Singh, A. Sharma, R. Rafailov, J. Schneider, T. Xie, S. Ermon, C. Finn, and A. Kumar. Preference fine-tuning of LLMs should leverage suboptimal, on-policy data. In *ICML*, pp. 47441–47474, 2024. (Cited on page 20.)

- Y. Tang, Z. Guo, Z. Zheng, D. Calandriello, R. Munos, M. Rowland, P. H. Richemond, M. Valko, B. Pires, and B. Piot. Generalized preference optimization: A unified approach to offline alignment. In *ICML*, pp. 47725–47742, 2024. (Cited on page 20.)
- K. Team, A. Du, B. Gao, B. Xing, C. Jiang, C. Chen, C. Li, C. Xiao, C. Du, C. Liao, et al. Kimi k1.5: Scaling reinforcement learning with LLMs. *ArXiv Preprint: 2501.12599*, 2025. (Cited on pages 1 and 19.)
- H. Touvron, T. Lavril, G. Izacard, X. Martinet, M-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *ArXiv Preprint: 2302.13971*, 2023. (Cited on page 1.)
- J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving, and I. Higgins. Solving math word problems with process-and outcome-based feedback. *ArXiv Preprint: 2211.14275*, 2022. (Cited on pages 1 and 20.)
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, pp. 6000–6010, 2017. (Cited on page 3.)
- H. Wang, C. Qian, W. Zhong, X. Chen, J. Qiu, S. Huang, B. Jin, M. Wang, K-F. Wong, and H. Ji. OTC: Optimal tool calls via reinforcement learning. *ArXiv Preprint: 2504.14870*, 2025. (Cited on page 19.)
- P. Wang, L. Li, Z. Shao, R. Xu, D. Dai, Y. Li, D. Chen, Y. Wu, and Z. Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *ACL*, pp. 9426–9439, 2024. (Cited on pages 1, 2, 19, and 20.)
- X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023. URL <https://openreview.net/forum?id=1PLlNIMMrw>. (Cited on page 19.)
- J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, pp. 24824–24837, 2022. (Cited on pages 1 and 19.)
- R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. (Cited on page 3.)
- Y. Wu, Z. Sun, S. Li, S. Welleck, and Y. Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *ICLR*, 2025. URL <https://openreview.net/forum?id=VNckp7JEHn>. (Cited on page 20.)
- V. Xiang, C. Snell, K. Gandhi, A. Albalak, A. Singh, C. Blagden, D. Phung, R. Rafailov, N. Lile, D. Mahan, et al. Towards system 2 reasoning in LLMs: Learning how to think with meta chain-of-thought. *ArXiv Preprint: 2501.04682*, 2025. (Cited on page 1.)
- J. Xiao, Z. Li, X. Xie, E. Getzen, C. Fang, Q. Long, and W. J. Su. On the algorithmic bias of aligning large language models with RLHF: Preference collapse and matching regularization. *ArXiv Preprint: 2405.16455*, 2024. (Cited on page 20.)
- Y. Xie, K. Kawaguchi, Y. Zhao, J. X. Zhao, M-Y. Kan, J. He, and M. Q. Xie. Self-evaluation guided beam search for reasoning. In *NeurIPS*, pp. 41618–41650, 2023. (Cited on page 19.)
- W. Xiong, H. Dong, C. Ye, Z. Wang, H. Zhong, H. Ji, N. Jiang, and T. Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. In *ICML*, pp. 54715–54754, 2024. (Cited on page 20.)
- W. Xiong, J. Yao, Y. Xu, B. Pang, L. Wang, D. Sahoo, J. Li, N. Jiang, T. Zhang, C. Xiong, and H. Dong. A minimalist approach to LLM reasoning: From rejection sampling to reinforce. *ArXiv Preprint: 2504.11343*, 2025a. (Cited on pages 1, 2, and 19.)
- W. Xiong, W. Zhao, W. Yuan, O. Golovneva, T. Zhang, J. Weston, and S. Sukhbaatar. StepWiser: Stepwise generative judges for wiser reasoning. *ArXiv Preprint: 2508.19229*, 2025b. (Cited on page 2.)

- H. Xu, A. Sharaf, Y. Chen, W. Tan, L. Shen, B. Van Durme, K. Murray, and Y. J. Kim. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *ICML*, pp. 55204–55224, 2024. (Cited on page 20.)
- Y. Yang, D. Campbell, K. Huang, M. Wang, J. Cohen, and T. Webb. Emergent symbolic mechanisms support abstract reasoning in large language models. *ArXiv Preprint: 2502.20332*, 2025. (Cited on page 19.)
- Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. 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. (Cited on page 1.)
- S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, pp. 11809–11822, 2023. (Cited on pages 1 and 19.)
- Y. Ye, Z. Huang, Y. Xiao, E. Chern, S. Xia, and P. Liu. LIMO: Less is more for reasoning. *ArXiv Preprint: 2502.03387*, 2025. (Cited on pages 6, 7, and 8.)
- F. Yu, A. Gao, and B. Wang. OVM, Outcome-supervised value models for planning in mathematical reasoning. In *NAACL*, pp. 858–875, 2024a. (Cited on page 20.)
- F. Yu, L. Jiang, H. Kang, S. Hao, and L. Qin. Flow of reasoning: Efficient training of LLM policy with divergent thinking. In *ICML*, pp. To appear, 2025a. (Cited on page 19.)
- L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. Kwok, Z. Li, A. Weller, and W. Liu. MetaMath: Bootstrap your own mathematical questions for large language models. In *ICLR*, 2024b. URL <https://openreview.net/forum?id=N8N0hgNDrt>. (Cited on page 19.)
- Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, T. Fan, G. Liu, L. Liu, X. Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *ArXiv Preprint: 2503.14476*, 2025b. (Cited on pages 1 and 3.)
- H. Yuan, Z. Yuan, C. Tan, W. Wang, S. Huang, and F. Huang. RRHF: Rank responses to align language models with human feedback. In *NeurIPS*, pp. 10935–10950, 2023a. (Cited on page 20.)
- L. Yuan, G. Cui, H. Wang, N. Ding, X. Wang, B. Shan, Z. Liu, J. Deng, H. Chen, R. Xie, Y. Lin, Z. Liu, B. Zhou, H. Peng, Z. Liu, and M. Sun. Advancing LLM reasoning generalists with preference trees. In *ICLR*, 2025. URL <https://openreview.net/forum?id=2ea5TNVR0c>. (Cited on page 20.)
- Z. Yuan, H. Yuan, C. Li, G. Dong, K. Lu, C. Tan, C. Zhou, and J. Zhou. Scaling relationship on learning mathematical reasoning with large language models. *ArXiv Preprint: 2308.01825*, 2023b. (Cited on page 20.)
- X. Yue, X. Qu, G. Zhang, Y. Fu, W. Huang, H. Sun, Y. Su, and W. Chen. MAMmoTH: Building math generalist models through hybrid instruction tuning. In *ICLR*, 2024. URL <https://openreview.net/forum?id=yLC1Gs770I>. (Cited on page 19.)
- E. Zelikman, Y. Wu, J. Mu, and N. D. Goodman. STaR: self-taught reasoner bootstrapping reasoning with reasoning. In *NeurIPS*, pp. 15476–15488, 2022. (Cited on pages 1 and 20.)
- K. Zha, Z. Gao, M. Shen, Z-W. Hong, D. S. Boning, and D. Katabi. RL Tango: Reinforcing generator and verifier together for language reasoning. *ArXiv Preprint: 2505.15034*, 2025. (Cited on page 2.)
- D. Zhang, S. Zhoubian, Z. Hu, Y. Yue, Y. Dong, and J. Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. In *NeurIPS*, pp. 64735–64772, 2024a. (Cited on page 4.)
- K. Zhang, Y. Hong, J. Bao, H. Jiang, Y. Song, D. Hong, and H. Xiong. GVPO: Group variance policy optimization for large language model post-training. *ArXiv Preprint: 2504.19599*, 2025a. (Cited on pages 1 and 3.)

- L. Zhang, A. Hosseini, H. Bansal, M. Kazemi, A. Kumar, and R. Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *The NeurIPS Workshop on System-2 Reasoning at Scale*, 2024b. URL <https://openreview.net/forum?id=aLgXy8A7k7>. (Cited on page 20.)
- Z. Zhang, C. Zheng, Y. Wu, B. Zhang, R. Lin, B. Yu, D. Liu, J. Zhou, and J. Lin. The lessons of developing process reward models in mathematical reasoning. *ArXiv Preprint: 2501.07301*, 2025b. (Cited on pages 1 and 2.)
- H. Zhao, G. I. Winata, A. Das, S-X. Zhang, D. Yao, W. Tang, and S. Sahu. RainbowPO: A unified framework for combining improvements in preference optimization. In *ICLR*, 2025. URL <https://openreview.net/forum?id=trKee5pIFv>. (Cited on page 20.)
- Y. Zhao, R. Joshi, T. Liu, M. Khalman, M. Saleh, and P. J. Liu. SLiC-HF: Sequence likelihood calibration with human feedback. *ArXiv Preprint: 2305.10425*, 2023. (Cited on page 20.)
- D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. V. Le, and E. H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *ICLR*, 2023. URL <https://openreview.net/forum?id=WZH7099tgfM>. (Cited on page 19.)
- D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences. *ArXiv Preprint: 1909.08593*, 2019. (Cited on page 20.)

A RELATED WORKS

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

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. The recent findings Xiong et al. (2025a) 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.

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. These methods (e.g., 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 to use comparison oracles, and showed that combining it with DPO effectively alleviated the issue of unintentional alignment.

Reward models. For the prompt x with a ground-truth response y_x^* , we evaluate by implementing a regular expression match on the final answer (Hendrycks et al., 2021): $r(x, y) = 1$ if y matches y_x^* on the *final answer* and $r(x, y) = 0$ otherwise. An *outcome reward* model (ORM) (Cobbe et al., 2021; Uesato et al., 2022) is trained for estimating $r(x, y)$. In particular, we first choose $x \in \mathcal{D}$ and collect training samples $(x, y \sim \pi_\theta(\cdot|x), r(x, y))$. Then, we take (x, y) as input and train an ORM to predict $r(x, y)$. 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., 2024b). 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 $s_h = (x, a_1, \dots, a_{h-1})$ 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).

Reinforcement learning from AI feedback. 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 RLAI, 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 approach can leverage 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 $r_i \in [0, 1]$.

B ADDITIONAL EXPERIMENTAL RESULTS

Beyond aggregate results, we provide a targeted analysis of SGPO’s impact. In line with our theoretical finding that SGPO converges faster than GRPO, empirical metrics offer supporting evidence. Prior work on RLVR entropy highlights its link to performance: Cui et al. (2025) showed that lower

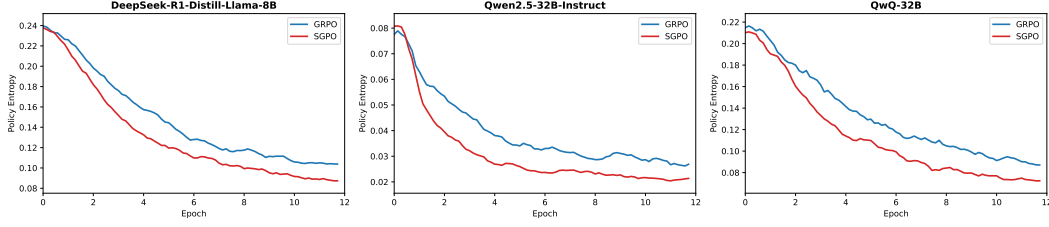


Figure 3: Policy entropy levels during training for GRPO and SGPO across different base models.

policy entropy under correct signals correlates with stronger policies, while Agarwal et al. (2025) demonstrated that directly minimizing entropy can improve performance. As shown in Figure 3, SGPO reduces policy entropy more rapidly than GRPO, indicating faster convergence toward deterministic RLVR behavior with higher rollout confidence. This matches our theoretical results, confirming that step-wise signals accelerate convergence.

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 3.3.

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):

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

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,

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

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 $g_{\text{GRPO}}(\cdot)$ as the gradient estimator using classical outcome reward model r , and $g_{\text{SGPO}}(\cdot)$ as the gradient estimator using the reward r_{SGPO} 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 SGPO and GRPO training dynamics with population-level policy gradient which can be computed exactly for the stylized model as follows,

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

Since $g_{\text{GRPO}}(\theta)$ and $g_{\text{SGPO}}(\theta)$ concentrate around $\bar{g}_{\text{GRPO}}(\theta)$ and $\bar{g}_{\text{SGPO}}(\theta)$ 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.

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

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

where the functions f_{ij} are defined by

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

C.2 TECHNICAL LEMMAS

We provide several technical lemmas that are important to the subsequent proof of Theorem 3.3. 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 Eq. (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

$$h'_p(x) = 1 - \frac{pe^x}{1-p+pe^x} = \frac{1-p}{1-p+pe^x} \stackrel{0 < p < 1}{>} 0.$$

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 $(-\infty, 0)$. $\square$

The second lemma presents an inequality which plays a key role in the proof of Theorem 3.3.

Lemma C.2. *We define the auxiliary functions as follows,*

$$\begin{aligned}A(x) &= 1 + \left(\frac{1}{x} - 1\right) e^{-x(1-x)}, \quad B(x, y) = 1 + \left(\frac{1}{y} - 1\right) e^{-x^2y(1-y)}, \\ C(z) &= 1 + \left(\frac{1}{z} - 1\right) e^{-z^2(1-z)}.\end{aligned}$$

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

Proof. We consider the lower and upper bound of e^{-u} when $u > 0$:

$$1 - u + \frac{u^2}{2} - \frac{u^3}{6} < e^{-u} < 1 - u + \frac{u^2}{2}.$$

Since $1/x - 1$, $1/y - 1$ and $1/\sqrt{xy} - 1$ are all positive, we have

$$\begin{aligned}A(x) &\leq 1 + \frac{1-x}{x} \left(1 - x(1-x) + \frac{x^2(1-x)^2}{2}\right) = \frac{1}{x} - (1-x)^2 + \frac{x(1-x)^3}{2}, \\ B(x, y) &\leq 1 + \frac{1-y}{y} \left(1 - x^2y(1-y) + \frac{x^4y^2(1-y)^2}{2}\right) = \frac{1}{y} - x^2(1-y)^2 + \frac{x^4y(1-y)^3}{2}, \\ C(z) &\geq 1 + \frac{1-z}{z} \left(1 - z^2(1-z) + \frac{z^4(1-z)^2}{2} - \frac{z^6(1-z)^3}{6}\right) \\ &= \frac{1}{z} - z(1-z)^2 + \frac{z^3(1-z)^3}{2} - \frac{z^5(1-z)^4}{6}.\end{aligned}$$

Set $z^2 = xy$, the original statement is equivalent to $(zC(z))^2 > (xA(x))(yB(x, y))$. Using the above upper and lower bound, it suffices to show $C_1(\sqrt{xy})^2 > A_1(x)B_1(x, y)$ where

$$\begin{aligned}A_1(x) &= 1 - x(1-x)^2 + x^2(1-x)^3/2, \quad B_1(x, y) = 1 - x^2y(1-y)^2 + x^4y^2(1-y)^3/2, \\ C_1(z) &= 1 - z^2(1-z)^2 + z^4(1-z)^3/2 - z^6(1-z)^4/6.\end{aligned}$$

By Lemma C.3, this is indeed true. This completes the proof. $\square$

Lemma C.3. *Define functions*

$$\begin{aligned}A_1(x) &= 1 - x(1-x)^2 + x^2(1-x)^3/2, \quad B_1(x, y) = 1 - x^2y(1-y)^2 + x^4y^2(1-y)^3/2, \\ C_1(z) &= 1 - z^2(1-z)^2 + z^4(1-z)^3/2 - z^6(1-z)^4/6.\end{aligned}$$

Then, $C_1(\sqrt{xy})^2 > A_1(x)B_1(x, y)$ for all $1/2 < y < x < 1$.

Proof. Let $x = u^2$ and $y = v^2$, then $1 > u > v > 1/\sqrt{2}$ and $z = uv$. We next show the desired inequality holds on a larger region, i.e., $1 > u > v > 2/3$. On this larger region, we have the reparameterization as follows:

$$u = \frac{2s+3}{3s+3}, \quad v = \frac{2r+2s+3}{3r+3s+3}, \quad s, r \in (0, +\infty),$$

or equivalently,

$$s = \frac{3(1-u)}{3u-2}, \quad r = \frac{3(u-v)}{(3u-2)(3v-2)}, \quad 1 > u > v > \frac{2}{3}.$$

It is easy to see this defines a one-to-one correspondence from (u, v) -space to (s, r) -space. Thus, we aim to prove the following function f is positive:

$$F(s, r) := C_1 \left(\frac{2s+3}{3s+3} \cdot \frac{2r+2s+3}{3r+3s+3} \right)^2 - A_1 \left(\left(\frac{2s+3}{3s+3} \right)^2 \right) B_1 \left(\left(\frac{2s+3}{3s+3} \right)^2, \left(\frac{2r+2s+3}{3r+3s+3} \right)^2 \right).$$

By leveraging SymPy's symbolic engine, the function expands and simplifies to:

$$F(s, r) = \frac{f(s, r)}{c(s+1)^{20}(r+s+1)^{20}}, \text{ where } f(s, r) := \sum_{k=0}^{20} c_{20-k}(s) r^{20-k},$$

where $c > 0$ is a universal constant, and single-variable polynomials $c_{20}(s), \dots, c_2(s), c_0(s) > 0$ and $\Delta_2 := c_1(s)^2 - 4c_2(s)c_0(s) < 0$, for all $s > 0$ (see Table 6 for details). Notice that from the table, we can see the only nontrivial parts are $c_3(s) > 0$ and $c_2(s) > 0$ because only these two contain negative coefficients. The positivity of $c_3(s)$ is simple because there is only one term (s^9) with negative coefficient and for all $s > 0$,

$$19471456710454363005152664s^{10} + 9684588377731643071927236s^8 > 14413823109350224541499726s^9.$$

To see this, simple estimation and AM-GM inequality yield

$$\text{LHS} > 1.9 \times 10^{25} s^{10} + 9.6 \times 10^{24} s^8 > 2\sqrt{182.4} \times 10^{24} s^9 > 2.7 \times 10^{25} s^9 > \text{RHS}.$$

The positivity of $c_2(s)$ is more complicated because it has 4 negative terms s^{10}, s^9, s^8, s^7 . However, we can use similar idea, i.e., choosing a pair of positive terms to bound a negative term:

$$\begin{aligned} 95791062786555508724088742320s^{15} + 571809550541807937530952s^5 &> 70273595236432368329707716s^{10}; \\ 43785862330162499052209529768s^{14} + 184789343789534461150530s^4 &> 99854072704322871537392604s^9; \\ 16326736853527122991715155824s^{13} + 35488375569622472169240s^3 &> 35726031377969792088188925s^8; \\ 4608219050084326790748933153s^{12} + 4362950858813170449228s^2 &> 6099037895307670142287608s^7. \end{aligned}$$

To see this, simple estimation and AM-GM inequality yield

$$\begin{aligned} \text{LHS} &> 9.5 \times 10^{28} s^{15} + 5.7 \times 10^{23} s^5 > 2\sqrt{541.5} \times 10^{25} s^{10} > 4.6 \times 10^{26} s^{10} > \text{RHS}; \\ \text{LHS} &> 4.3 \times 10^{28} s^{14} + 1.8 \times 10^{23} s^4 > 2\sqrt{77.4} \times 10^{25} s^9 > 1.7 \times 10^{26} s^{10} > \text{RHS}; \\ \text{LHS} &> 1.6 \times 10^{28} s^{13} + 3.5 \times 10^{22} s^3 > 2\sqrt{5.6} \times 10^{25} s^9 > 4.7 \times 10^{25} s^8 > \text{RHS}; \\ \text{LHS} &> 4.6 \times 10^{27} s^{12} + 4.3 \times 10^{21} s^2 > 2\sqrt{19.7} \times 10^{24} s^7 > 8.8 \times 10^{24} s^7 > \text{RHS}. \end{aligned}$$

In conclusion, we have all coefficient $c_i(s)$ positive except $c_1(s)$, but it doesn't affect the positivity of f because $\Delta_2 < 0$. This completes the proof. $\square$

Table 6: Coefficient Lists of $F(s, r)$

Notation	Value
c	437675956526049436836
c_{20}	$2(2s + 3)^2(1714774320744848750s^{18} + 26610409260691576200s^{17} +$ $191778746468802317181s^{16} + 850194149855082319224s^{15} +$ $2587082434290045806049s^{14} + 5704415906039160731874s^{13} +$ $9366197581963232054460s^{12} + 11563054951307567026248s^{11} +$ $10670965452123886149660s^{10} + 7187176769582075261292s^9 +$ $3372972168996579430017s^8 + 1072082836158220703952s^7 +$ $370302094042890771285s^6 + 329901677376902425818s^5 +$ $282799986805616267862s^4 + 151168270170893365008s^3 +$ $49139849518345513368s^2 + 9090603727935062976s + 742484948385838248)$
c_{19}	$8(2s + 3)^2(8573871603724243750s^{19} + 141183751848798840450s^{18} +$ $1085065457535611084097s^{17} + 5159905424662678527663s^{16} +$ $16962017821014041355285s^{15} + 40761515892906930261393s^{14} +$ $73777358861762677983126s^{13} + 101968163476291942277643s^{12} +$ $107719550183279202945336s^{11} + 85951871005332247942347s^{10} +$ $50477142420763872747039s^9 + 21228461710484227270812s^8 +$ $7175576253360286202193s^7 + 3821642119447908810138s^6 +$ $3343140602539615070982s^5 + 2308058741380310946144s^4 +$ $1046528691565621471344s^3 + 300769285860744146028s^2 +$ $50403014643440592936s + 3785769852305984190)$
c_{18}	$2(2s + 3)^2(325807120941521262500s^{20} + 5673987380977312396200s^{19} +$ $46320143614200183575358s^{18} + 235164624868455434314740s^{17} +$ $830330782346499878631402s^{16} + 2159105892766696625508432s^{15} +$ $4268066364777710628878112s^{14} + 6521177726276586191628264s^{13} +$ $7742933419720025359660131s^{12} + 7110911361873582109051992s^{11} +$ $4976474876383070663195517s^{10} + 2600230719135591148269222s^9 +$ $1035307928116150386109695s^8 + 420495220158165783672300s^7 +$ $287436345862168026209421s^6 + 225749912117527047120354s^5 +$ $13220553924765757023286s^4 + 52546262098747895532864s^3 +$ $13596497258861930544108s^2 + 2087305777245729888936s +$ $145293329180967197454)$
c_{17}	$12(2s + 3)^2(325807120941521262500s^{21} + 5982992191700268855300s^{20} +$ $51700930512613327403406s^{19} + 27907978077259477944590s^{18} +$ $1053187317945045364767966s^{17} + 2945458902809849586876810s^{16} +$ $6310900331865363012743844s^{15} + 10554182290179327214273482s^{14} +$ $13894164004300292404029789s^{13} + 14397511755502695216399777s^{12} +$ $11648187532971253859583195s^{11} + 7253672843249894875203621s^{10} +$ $3463934076062711984148183s^9 + 1401311871124636922510679s^8 +$ $709635073453803384330663s^7 + 526601300781722718969621s^6 +$ $366411462920903557014120s^5 + 187790938366917450633606s^4 +$ $66828265788979238418684s^3 + 15774993964318985462592s^2 +$ $2238177282159012945966s + 145311275743961970078)$
c_{16}	$3(2s + 3)^2(5538721056005861462500s^{22} + 106963949041194830344800s^{21} +$ $975372417387075095557110s^{20} + 5577701869567635624516312s^{19} +$ $22401070138854784562671602s^{18} + 67034725561809321462257220s^{17} +$ $154689592660061775780683034s^{16} + 280889539801332767094091608s^{15} +$ $405664328993005936098158220s^{14} + 467432234597450323377654624s^{13} +$ $428162911701836470453488816s^{12} + 308851193177419859648120664s^{11} +$ $173923711507624595412555792s^{10} + 78611043902295277305014364s^9 +$ $34502723932108659968068191s^8 + 20784869678087847011350512s^7 +$ $15211070491275270899260479s^6 + 9439795169478817720557390s^5 +$ $4323642777299210867466840s^4 + 1398336255225225668686176s^3 +$ $304165103383419145366680s^2 + 40170170984468888396880s +$ $2445688799534592091926)$

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

c_{15}	$12(2s + 3)^2(4430976844804689170000s^{23} + 89773624658788072119600s^{22} + 861449789967478967902968s^{21} + 5202052340570363961799272s^{20} + 22150958674722147636887880s^{19} + 70610768977431597138502416s^{18} + 174547747719038741579249148s^{17} + 341846798830220759744006802s^{16} + 537017897494050473220887475s^{15} + 680386317088643909072652357s^{14} + 694900917445527709102606203s^{13} + 568854294351452352559155357s^{12} + 370162500862325208186949407s^{11} + 192089934615274810143113637s^{10} + 85560000374404871078044743s^9 + 42188979069325595690484894s^8 + 27986482983635448916149477s^7 + 19349473000527948423160098s^6 + 10826549522417650582347903s^5 + 4499119009419911850147903s^4 + 1337268081929646109116429s^3 + 270188727447397870150299s^2 + 33412191448722202871793s + 1916481339467789227047)$
c_{14}	$3(2s + 3)^2(44309768448046891700000s^{24} + 939760900846202799633600s^{23} + 9465882231449979270581616s^{22} + 60189415644287265430240224s^{21} + 270831851136366961169859120s^{20} + 916082243422713193340650464s^{19} + 2414605718335618880853970032s^{18} + 5071784805050410035823167576s^{17} + 8606003351786628019139811024s^{16} + 11882072721559268002578594336s^{15} + 13373135736066588915267225192s^{14} + 12233789753017327381398656592s^{13} + 9038087243602138768195078704s^{12} + 5369286552690873446276117328s^{11} + 2623535852573549267091555300s^{10} + 1196071780982939651865144096s^9 + 660078244623496780263588075s^8 + 451393195997666208667458852s^7 + 290343677073268941864046305s^6 + 148042042832629803967321050s^5 + 56464193331043404116741784s^4 + 15559478430192850829818824s^3 + 2939179015494775847121192s^2 + 342046929585497061253176s + 18557914646800278459054)$
c_{13}	$6(2s + 3)^2(44309768448046891700000s^{25} + 981785555104524878071200s^{24} + 10357132334422169539112688s^{23} + 69166131783384274997062320s^{22} + 327906609514495942625889552s^{21} + 1172872936442019296825004672s^{20} + 3283063721629310545911589320s^{19} + 7360333239220785966714322212s^{18} + 13411238091959519801173855314s^{17} + 20030874331258639447616405430s^{16} + 24612632066584435063045693896s^{15} + 24861807943243247848564702728s^{14} + 20554616309938676564088193632s^{13} + 13828701774644445607198489296s^{12} + 7593307018019776947591316692s^{11} + 3573164437682539651140298914s^{10} + 1711205400898634741432588709s^9 + 1026436201325482873227731181s^8 + 693346698158130213351442587s^7 + 413729468327452341092783823s^6 + 194077643830831926535960674s^5 + 68561905220231775619051080s^4 + 17640970508803332546397944s^3 + 3132633769453198732801032s^2 + 344560424227000935565860s + 17745096925489168423620)$
c_{12}	$3(2s + 3)^2(144006747456152398025000s^{26} + 3327383180429252608653600s^{25} + 36686806677921921575024412s^{24} + 256712663785782687868607040s^{23} + 1278878488443918869718977532s^{22} + 4822501965635852078399708760s^{21} + 14284937167605407625123613032s^{20} + 34039675734020531218713639960s^{19} + 66269194087347804842641890936s^{18} + 106420251181999059483739591464s^{17} + 141673718347886548611527896944s^{16} + 156512253734744849831503592064s^{15} + 143119458698672790284020359156s^{14} + 107772919457957006535562903236s^{13} + 66583563316818931905117287625s^{12} + 3420802797787101480006575072s^{11} + 15821430510294339891416781741s^{10} + 8030913522092664743057080482s^9 + 5050521785734143734257145460s^8 + 3292374301511579644939102872s^7 + 1827212365211212115171329068s^6 + 795146625577647417589837740s^5 + 262142182103903879108812389s^4 + 63354427214296180937779584s^3 + 10625610419721898094320272s^2 + 1108770079900593722922360s + 54371622599418650521707)$

c_{11}	$2(2s + 3)^2(288013494912304796050000s^{27} + 6927926613537598727151600s^{26} + 79684994050724088696229560s^{25} + 583013575348670021732718984s^{24} + 3044719667796555717818159544s^{23} + 12071152881690852686028406920s^{22} + 37719751802283064561860228312s^{21} + 95186570754684978954315680724s^{20} + 197141443250509137121100298018s^{19} + 338621398294617306936300888486s^{18} + 485320394172125494078362290142s^{17} + 581787906451611153835763035926s^{16} + 582802786083324408675746802192s^{15} + 485993832849417808677870096480s^{14} + 335594444689992516446373541470s^{13} + 191818678234951313275318187166s^{12} + 93215508141687941579846043591s^{11} + 42995488556619072965421127845s^{10} + 22993334205445704498638551659s^9 + 14702002876383491619167125293s^8 + 9151802478157350485470780638s^7 + 4746495171599998113140409294s^6 + 1930238883493800552821467836s^5 + 597661705826550277052178582s^4 + 136369421095118152409287875s^3 + 21690051407678516824173015s^2 + 2154400447443907595487183s + 100873131758694046028745)$
c_{10}	$(2s + 3)^2(633629688807070551310000s^{28} + 15842391105676722921391200s^{27} + 189762104968446645014104296s^{26} + 1448899233419905865696230704s^{25} + 7915017749037631812296989272s^{24} + 32911374092963881171128851808s^{23} + 108184113680419836957062571120s^{22} + 288181457405913203915560147656s^{21} + 632563077547125768289724175852s^{20} + 1156966248842167228077853758672s^{19} + 1775610230389152091840398326292s^{18} + 2294661769917892997575576903488s^{17} + 2498192965867701162268688835924s^{16} + 2285659112305489254434518711416s^{15} + 1748991971403770816761602941934s^{14} + 1113797684163785846264967323376s^{13} + 592491395159743574390096742111s^{12} + 274476600715541565615011213796s^{11} + 127044619585271350650188845452s^{10} + 70538582326147723865407765680s^9 + 44889681102540017067072498465s^8 + 26602898887160759567836334520s^7 + 12967769746561749479001701280s^6 + 4960789745148078555411299844s^5 + 1450732098731590049595423084s^4 + 313930894468329082145284956s^3 + 47525670620612611058618019s^2 + 4506795195444098216749962s + 201981066438920088074121)$
c_9	$2(2s + 3)^2(288013494912304796050000s^{29} + 7474247118895785746840400s^{28} + 93085137978826092989684184s^{27} + 740402244183890613610238616s^{26} + 4222492016436341447317422840s^{25} + 18373474587737757143576597976s^{24} + 63374311264983526695777429384s^{23} + 177689296198984770408646605492s^{22} + 411992129847683864453855977890s^{21} + 799270570087512158479167778014s^{20} + 1307454563265970530661662310488s^{19} + 1811440906673507850196226957292s^{18} + 2129008833631551803404401966618s^{17} + 2120318902997465121470552812398s^{16} + 1782752578142774040882126045846s^{15} + 1258348636755929955587480574282s^{14} + 742222950605778500648135120808s^{13} + 368759532677717760323871620448s^{12} + 163292340177616665345297878034s^{11} + 75628135544208165254280371040s^{10} + 42858735954613893102489442569s^9 + 26815609860493555439017106967s^8 + 15154066307595953401565854680s^7 + 6986502898234763627448498006s^6 + 2529774938694747662152377363s^5 + 702330501810674770487862723s^4 + 144732655027863948827862261s^3 + 20924547208371262485560295s^2 + 1899529777608715114521399s + 81667865388403953518175)$

c_8	$3(2s + 3)^2(144006747456152398025000s^{30} + 3873703685787439628342400s^{29} + 50086950606023925868479036s^{28} + 414347293343103311413357200s^{27} + 2462455716417895553466814404s^{26} + 11190311839066344259536714024s^{25} + 40409295972773177050104566556s^{24} + 118947134811306669490272616200s^{23} + 290460448001449937986609641984s^{22} + 595650982876961700444247916136s^{21} + 1034394503332837044832871973660s^{20} + 1529122522398341452237748264328s^{19} + 1929170112144408531295247240352s^{18} + 2077275729835779898627972452564s^{17} + 1904400092584849147398803257857s^{16} + 1479295019246765244421617960408s^{15} + 967232587581523134128767690737s^{14} + 529827149456294513579866857762s^{13} + 245850149371151220247602350838s^{12} + 103586181981415962365079350136s^{11} + 47441969831005210765856651490s^{10} + 27017370328613731242988726116s^9 + 16543604074621342316534833896s^8 + 8959808104815503968120901160s^7 + 3934148480503417587877177653s^6 + 1356708501926603220744080694s^5 + 359356010050812330787473279s^4 + 70800398025365428901919252s^3 + 9805705639441860010993074s^2 + 854294427989006197205052s + 35306186075611540243056)$
c_7	$12(2s + 3)^2(22154884224023445850000s^{31} + 616966740327228674348400s^{30} + 8270907073696162683430488s^{29} + 71054888374958447419331400s^{28} + 439319157110546803811896968s^{27} + 2081153035704566658130624800s^{26} + 7851699408458680802242634484s^{25} + 24207516760625423072207073414s^{24} + 62092517428805875866794674881s^{23} + 134191349159021239688414857431s^{22} + 246518378732311775029464265197s^{21} + 387229048800653531738114739999s^{20} + 521849077561894657636398449655s^{19} + 604010499659026951460897607741s^{18} + 599706193952136964321127694249s^{17} + 508932600092700362308014694866s^{16} + 366923845830106735087199088303s^{15} + 222970678878271885535159028828s^{14} + 113510751404816691589761355815s^{13} + 48959621580159733845289884633s^{12} + 19343819733170955544319552163s^{11} + 8570967648243419298660377211s^{10} + 4842281705213527108416513279s^9 + 2909138374562317990360031394s^8 + 1522921303596173996138575905s^7 + 642212965528640572310436879s^6 + 212347408792609836644815359s^5 + 53940504869138165983355556s^4 + 10200981421548866272272021s^3 + 1357609423748226427157778s^2 + 113782587792457476788865s + 4528476210896135134182)$
c_6	$3(2s + 3)^2(44309768448046891700000s^{32} + 1275958134912779427134400s^{31} + 17712124648743520374246000s^{30} + 157800783303192342484807776s^{29} + 1013476002096822653934247536s^{28} + 4996346811471994438240970400s^{27} + 19656891049003759997738219664s^{26} + 63343018585986823082619729288s^{25} + 170258537043632779229473723584s^{24} + 386719255970188993917224574336s^{23} + 749195973037122202127202135768s^{22} + 1245952789865163842159925500256s^{21} + 1785990969360715046000117363568s^{20} + 2210911897085152430737822315488s^{19} + 2363325656949833631869930474232s^{18} + 2176386534475868543250178452024s^{17} + 1718519269597730769046219931925s^{16} + 1154898997656814617038309424924s^{15} + 653993799682370716585040976045s^{14} + 309097812387589842994049088882s^{13} + 122649986514805927975811013678s^{12} + 44139321593386486437541304232s^{11} + 18194600512073131179652474218s^{10} + 10080729455247177149466360108s^9 + 6012154495009832863143860703s^8 + 3086713901904485302077423912s^7 + 1264536513002644891553573532s^6 + 404269004313535997709064188s^5 + 99085318554137395893467265s^4 + 18066869592965845573731768s^3 + 2318037559219684454533893s^2 + 187335877784959577298978s + 7192069815364796066229)$

c_5	$6(2s + 3)^2(8861953689609378340000s^{33} + 263596557834220301114400s^{32} + 3784460184258343211722032s^{31} + 34920466929143934879808368s^{30} + 232642130166877120425752976s^{29} + 1191697800945367086458055072s^{28} + 4880694751880441559218476632s^{27} + 16406950824469547680560778764s^{26} + 46112865280418851489322074134s^{25} + 109812274944960351155991786018s^{24} + 223725856433516713998849805632s^{23} + 392660140917098227883941019880s^{22} + 596450332899276085006754000436s^{21} + 786241559139425306167747437324s^{20} + 900162472445706270827205735408s^{19} + 894118456198813762374744874842s^{18} + 768019765700356821541048064211s^{17} + 567188919515029404795124639815s^{16} + 356892849173690588784279865923s^{15} + 188887726440058090918123569807s^{14} + 82758519677573937797801658678s^{13} + 29755841968295155180938377160s^{12} + 9265655123985315388229427762s^{11} + 3269804705124270960946179492s^{10} + 1745378750437674672254525343s^9 + 1067124171571276165079329161s^8 + 553192327447935426146669544s^7 + 224471881547232310097558892s^6 + 70271287113829333092849234s^5 + 16759856517295120894936656s^4 + 2963427571961631174417201s^3 + 367988093962382467875321s^2 + 28750493973686584294998s + 1066338997876680421860)$
c_4	$3(2s + 3)^2(5538721056005861462500s^{34} + 170000930428677948001200s^{33} + 2521542870629614052494182s^{32} + 24069033861073900393194288s^{31} + 166112140239471730224127950s^{30} + 882861984743248195220227068s^{29} + 3758135559481861075860190560s^{28} + 13155787658258741235890755800s^{27} + 38587134973448786366330248032s^{26} + 96129109345583765886638321232s^{25} + 205447147746380335664768754954s^{24} + 379449453375926778871028692368s^{23} + 608776780980435381145177931646s^{22} + 851251839731897497067600541072s^{21} + 1039128387112340409627618422211s^{20} + 1107338907683241287910817574616s^{19} + 1028236425910911889110211542327s^{18} + 828653388719550997841334599814s^{17} + 575679860099214460991336767122s^{16} + 341113980250874385715195498752s^{15} + 169602112160966873244672554562s^{14} + 69019121880524178108395152092s^{13} + 22192275965509681657763172138s^{12} + 5513039324469607430622101184s^{11} + 1308552733582056616685717799s^{10} + 622822426968272935610750562s^9 + 441208218615564366021478680s^8 + 251086626744997918806221832s^7 + 105542726343197962447668330s^6 + 33182712041751038192684904s^5 + 7819078036046387444720541s^4 + 1353372939412637489000388s^3 + 163593372233172281873202s^2 + 12396727299661082512092s + 444813790293506728368)$
c_3	$6(2s + 3)^2(651614241883042525000s^{35} + 20618119083643318565400s^{34} + 315621334999692786151116s^{33} + 3113065178572047666007860s^{32} + 22229797545183952680520284s^{31} + 122422561739941481599835484s^{30} + 540840514528742111761776912s^{29} + 1968384017248063183323104976s^{28} + 6014334916396782599305697730s^{27} + 15642733509059107994947830402s^{26} + 34991063427920339849664070098s^{25} + 67834512242456744868412053510s^{24} + 114610735641635823232680558264s^{23} + 169420289219412443682245006436s^{22} + 219630129457905943676457475806s^{21} + 249914888458917575863280485878s^{20} + 249458032509437740525240777731s^{19} + 217920380868106493481515756421s^{18} + 165868299839291545410161133771s^{17} + 109196804131621756456248426789s^{16} + 61455743971044438451107683232s^{15} + 29010820857450160933812140136s^{14} + 11111724769820435809112639700s^{13} + 3230978841571830756780993930s^{12} + 597734071316630030096071452s^{11} + 19471456710454363005152664s^{10} + 14413823109350224541499726s^9 + 9684588377731643071927236s^8 + 12560219583039185504039910s^7 + 6629813221106696409536742s^6 + 2267528918316638233964400s^5 + 549510851612447197946100s^4 + 95169593716835317546698s^3 + 11333379076369208961606s^2 + 837761545501050427146s + 29118359247617829474)$

1566		
1567		
1568		
1569		
1570		
1571		
1572		
1573		
1574		
1575		
1576		
1577		
1578		
1579		
1580		
1581		
1582		
1583		
1584		
1585		
1586		
1587		
1588		
1589		
1590		
1591		
1592		
1593		
1594		
1595		
1596		
1597		
1598		
1599		
1600		
1601		
1602		
1603		
1604		
1605		
1606		
1607		
1608		
1609		
1610		
1611		
1612		
1613		
1614		
1615		
1616		
1617		
1618		
1619		

c_2	$(2s + 3)^2(651614241883042525000s^{36} + 21236128705089231483600s^{35} + 335176828913504517258492s^{34} + 3412480272349060314590760s^{33} + 25184187621847674304218612s^{32} + 1435324779126332041071111040s^{31} + 657199533413487737746005840s^{30} + 2483057802762141981372604872s^{29} + 7890444256414221204984097182s^{28} + 21386706626264438997499257504s^{27} + 49968626609865032054943532722s^{26} + 101443944207534124952038283748s^{25} + 180022708498134425424774257358s^{24} + 280473316548653330703764939232s^{23} + 384770188717561290211106882724s^{22} + 465568061409025346140795055268s^{21} + 497070528817890647576496874221s^{20} + 467853455550722338016560291068s^{19} + 387255091639245361342752551670s^{18} + 280658492377779215502417761676s^{17} + 176840580743027422215976620477s^{16} + 95791062786555508724088742320s^{15} + 43785862330162499052209529768s^{14} + 16326736853527122991715155824s^{13} + 4608219050084326790748933153s^{12} + 763090880285610579007863108s^{11} - 70273595236432368329707716s^{10} - 99854072704322871537392604s^9 - 35726031377969792088188925s^8 - 6099037895307670142287608s^7 + 385997000711223832356168s^6 + 571809550541807937530952s^5 + 184789343789534461150530s^4 + 35488375569622472169240s^3 + 4362950858813170449228s^2 + 320602140994390122456s + 10806813741383936712)$
c_1	$2s^2(2s + 3)^2(5s + 6)^2(1371819456595879000s^{33} + 42716345570559668088s^{32} + 643544475178313701908s^{31} + 6247566555702580389060s^{30} + 43916765565269249016660s^{29} + 238129437052493452170540s^{28} + 1036074226442125183419168s^{27} + 3714930106381256786671824s^{26} + 11187728687072575053763704s^{25} + 28696708254175557235886484s^{24} + 63352891406570842442956878s^{23} + 121330161626918848657558398s^{22} + 202765255131175811905486752s^{21} + 296952156159684842999476188s^{20} + 382193906007924533060066196s^{19} + 432977106373155008061033636s^{18} + 431882942555537334945182376s^{17} + 378922778321739951076319160s^{16} + 291693785851369028780662644s^{15} + 196147224158011892839433394s^{14} + 114410349624644626423212729s^{13} + 57248884332430976396451627s^{12} + 24130617950503756984051008s^{11} + 8287189086050003227856022s^{10} + 2152391916458370195915867s^9 + 325184614597411140314601s^8 - 33007972351878903475404s^7 - 41792510938686638897304s^6 - 15831185972996449730358s^5 - 3877006197061115690130s^4 - 665506024092175855680s^3 - 78419814392209911120s^2 - 5759186746951521828s - 200126180395998828)$
c_0	$s^4(2s + 3)^2(5s + 6)^4(5487277826383516s^{30} + 162900207047936448s^{29} + 2337377142714373098s^{28} + 21588165729897598296s^{27} + 144210704373637237422s^{26} + 742206852421449807276s^{25} + 3061285798160471289822s^{24} + 10391895636644586020112s^{23} + 29588363727735612069036s^{22} + 71651404108146138897096s^{21} + 149117620073027461547436s^{20} + 268806705178958727187248s^{19} + 422185992215286625454736s^{18} + 580191752386498986323712s^{17} + 699694724310231624064272s^{16} + 741757261801656111225984s^{15} + 691666247103583662351612s^{14} + 567033087779716710023352s^{13} + 408047131644656580559296s^{12} + 257036644794565634383008s^{11} + 141145264141826804073576s^{10} + 67178460532288884909516s^9 + 27499663057942951141041s^8 + 9582491278489242855672s^7 + 2803365139419823150782s^6 + 675682953313836552876s^5 + 130659498671205647052s^4 + 19489563056909654496s^3 + 2105305456045150908s^2 + 146594486051390088s + 4941387170271576)$

1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673

Δ_2

$$\begin{aligned}
& -36s^4(2s+3)^4(5s+6)^4(188188862149501274759871978264100000s^{66} \\
& +11719822793673389354852048917721870400s^{65} \\
& +359034059515843764915172404946627408992s^{64} \\
& +7212088571266452516322419090023965301952s^{63} \\
& +106839132025368784093448214456114080089896s^{62} \\
& +1244653930745343462400612341123366430556112s^{61} \\
& +11874531272586759157048871425273776419646480s^{60} \\
& +95397174639780849117400217925290860797543072s^{59} \\
& +658601421196162394675006092409270655783387096s^{58} \\
& +3968022355733696091765263130542489256493297296s^{57} \\
& +21117182157874536980169655465156091744276920380s^{56} \\
& +100234695532226306436649215523120163350432040784s^{55} \\
& +427723545408455739277440670239201960498527445168s^{54} \\
& +1651693204898583110001983831675383703838803111280s^{53} \\
& +5803881305128515928064472759889503906134840719788s^{52} \\
& +18645262399111732404927179918528703191829668776944s^{51} \\
& +54982160923970862907155853987157692828051002129468s^{50} \\
& +149340327131211909506021191619303279476069307001480s^{49} \\
& +374738908504925756556865067810806785110272348870998s^{48} \\
& +870959171363388767125455183938474548684433032733976s^{47} \\
& +1879125423105677523532695113591118653251030543219650s^{46} \\
& +3770893425502604495219029177077658520798571516698604s^{45} \\
& +7050015221784174490794108388251813073259643663091008s^{44} \\
& +12297516978145400887377430125405972391309535236404816s^{43} \\
& +20038164484164236659289158464333223322979861456369558s^{42} \\
& +30532204930658301922179960307897986850960973110401060s^{41} \\
& +43539740165642307747449376716152274738997971952381288s^{40} \\
& +58147942722340474688455284463585466926829447124690864s^{39} \\
& +72765609819670245981642226229421333542816165300786478s^{38} \\
& +85352226271944117403522054472274990615987412077658020s^{37} \\
& +93861185123580180986528832241395054124125210332787978s^{36} \\
& +96773676228707454794185949836280770022200702637789136s^{35} \\
& +93535505340566829835280969592461902479874869737070324s^{34} \\
& +84726573070331222224010481384087646174794999517275720s^{33} \\
& +71892923025396467242385567135204296459367606931171072s^{32} \\
& +57107505385821670765861094421169358296399126222638720s^{31} \\
& +42429475579369200294980396815322839839865021711481712s^{30} \\
& +29453402248671311370918975586371950400100875859184424s^{29} \\
& +19076887418815475591391225344828673331985433949859745s^{28} \\
& +11509630902613742430056710113025979946368929363591560s^{27} \\
& +6455314570589262935425947995355398219841083619460275s^{26} \\
& +3357432561061020603580382785922585900536396393910862s^{25} \\
& +1614483581569691108434974127018929362203153022365980s^{24} \\
& +715184392123946764432274154396140356129136774278152s^{23} \\
& +290561593707882411154971270303899134550457836920080s^{22} \\
& +107686614725038641896140906093508540897806554334884s^{21} \\
& +36175123767827876571287168571772096066641842189715s^{20} \\
& +10936295413621417840147244658584579127207174389352s^{19} \\
& +2956231446951945885452453177600650222360031507853s^{18} \\
& +714328315495087334767452511829810830341704532294s^{17} \\
& +157862256397341201315796262528952028945210160178s^{16} \\
& +34669184815408932243484290688498684622916000552s^{15} \\
& +8792208859688249070338942374434094956929968434s^{14} \\
& +2728958913497904679605629304348283067662149156s^{13} \\
& +908514477310679518991239534084776498569549130s^{12} \\
& +281997384651571824431416925444015818101696600s^{11} \\
& +7659571256132994441138942454188872738496992s^{10}
\end{aligned}$$

1674	+17823235368805916505433544016195415996681584s ⁹
1675	+3522507407940359387775769801382214828420516s ⁸
1676	+586782005064608000832043474107919395519552s ⁷
1677	+81484523152763229188477555846803640661672s ⁶
1678	+9275689993243887053899463828498929963344s ⁵
1679	+843808167321923057379954129881612634096s ⁴
1680	+58988087589412129623633466476137297856s ³
1681	+2972581287433071568118115850779633072s ²
1682	+95923391203691628771401672158154016s
1683	+1483351410366365393372190806569392)

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

Lemma C.4. *Under the assumptions from Theorem 3.3, the following statements hold true,*

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

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

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

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

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

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

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

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

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

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

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

Then, Lemma C.1(ii) and $p_{SGPO}^{(k)} \in (0, 1)$ yield

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

In addition, we have

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

Putting these pieces together yields $p_{SGPO}^{(k+1)} > q_{SGPO}^{(k+1)}$. $\square$

C.3 PROOF OF THEOREM 3.3

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

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

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 $p_{\text{SGPO}}^{(k)} \rightarrow 1$ as $k \rightarrow \infty$. Similarly, we can show $q_{\text{GRPO}}^{(k)}, p_{\text{SGPO}}^{(k)}, q_{\text{SGPO}}^{(k)} \rightarrow 1$ as $k \rightarrow \infty$.

To show (ii), consider the base case:

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

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

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

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

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

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

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

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

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

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

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

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

The above inequality holds true since Lemma 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 $p_{\text{SGPO}}^{(k)} q_{\text{SGPO}}^{(k)} > (p_{\text{GRPO}}^{(k)})^2$ implies $p_{\text{SGPO}}^{(k+1)} q_{\text{SGPO}}^{(k+1)} > (p_{\text{GRPO}}^{(k+1)})^2$ for $k \geq 1$. By Lemma C.4(iii), we know $p_{\text{SGPO}}^{(k)} > q_{\text{SGPO}}^{(k)}$ for all $k \geq 1$. Thus, Lemma C.2 implies that

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

Using Lemma C.1(iii), we complete the induction by applying our induction hypothesis:

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

This completes the proof.

C.4 EMPIRICAL EVALUATIONS

We conduct simple simulations to compare SGPO and GRPO under the stylized model and plot the resulting learning curves in Figure 4. The left panel shows the probability of selecting the “good” action in the first step at iteration k (i.e., $p_{\text{SGPO}}^{(k)}$ vs. $p_{\text{GRPO}}^{(k)}$), while the right panel shows the probability of learning the optimal policy (i.e., $p_{\text{SGPO}}^{(k)}q_{\text{SGPO}}^{(k)}$ vs. $p_{\text{GRPO}}^{(k)}q_{\text{GRPO}}^{(k)}$). The results align with the predictions of Theorem 3.3, demonstrating that the likelihood of learning the optimal policy under SGPO consistently exceeds that of GRPO across training.

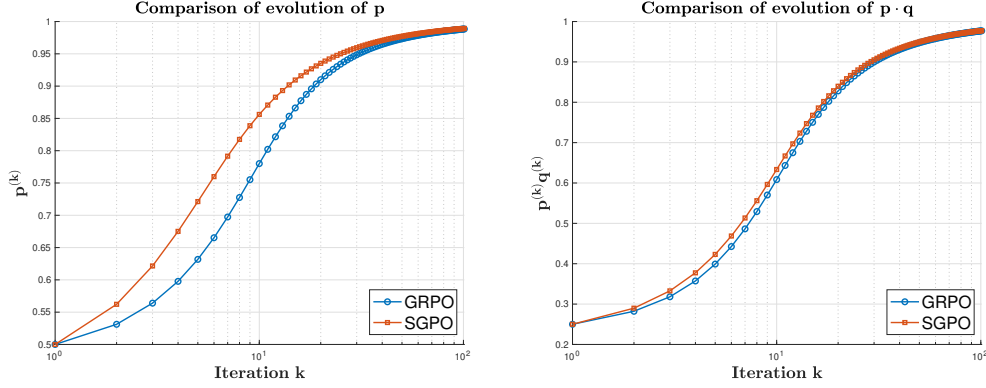


Figure 4: Learning dynamics of GRPO and SGPO in the simplified setting.

D THE USE OF LARGE LANGUAGE MODELS (LLMs)

In addition to serving as the step-wise judge model, we also use LLMs to aid and polish the writing. Specifically, we employ them for grammar and style checking to improve the readability of this work.