GTPO: Stabilizing Group Relative Policy Optimization via Gradient and Entropy Control

Marco Simoni[1,3]*, **Aleksandar Fontana**[2,3]*, **Giulio Rossolini**[2], **Andrea Saracino**[2], **Paolo Mori**[3]
[1] National Doctorate on Artificial Intelligence, Sapienza Università di Roma
[2] Department of Excellence in Robotics and AI, TeCIP, Scuola Superiore Sant'Anna, Pisa
[3] Institute of Informatics and Telematics, National Research Council of Italy, Pisa

Abstract

Group Relative Policy Optimization (GRPO) is a promising policy-based approach for Large Language Model alignment, yet its performance is often limited by training instability and suboptimal convergence. In this paper, we identify and analyze two main GRPO issues: *(i)* the token-level penalization, where valuable tokens shared across different responses receive contradictory feedback signals, leading to conflicting gradient updates that can reduce their likelihood; and *(ii)* the policy collapse, where negatively rewarded completions may penalize confident responses and shift model decisions toward unlikely tokens, destabilizing training process. To address these issues we introduce *GTPO (Group-relative Trajectory-based Policy Optimization)*, which prevents conflicting gradients on valuable tokens by skipping negative updates while amplifying positive ones and filters out completions whose entropy exceeds a provable threshold, to prevent policy collapse. By omitting KL-divergence regularization, GTPO eliminates the reference model while achieving superior stability and performance over GRPO and DAPO. Extensive evaluations across GSM8K, MATH, R1, AIME 2024–2025, AMC 2023, and MMLU validate these gains. The code is available here.[1].

1 Introduction

Recent advances in the training of large language models (LLMs) have led to the use of policy-based optimization methods, where the model is treated as a policy that is updated to better match human preferences. Methods such as DPO (Rafailov et al., 2023), RLHF (Ouyang et al., 2022), and RFT (Yuan et al., 2023) follow this idea: they collect human or heuristic feedback on model outputs and use it as a reward signal to increase the likelihood of preferred or desired responses. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is a more recent RL-based method that builds on PPO (Schulman et al., 2017) but removes the need for a separate critic network, i.e., the value function estimator. GRPO compares multiple completions for the same question, each consisting of natural-language reasoning followed by the final answer, and computes advantages from their relative rewards based on completion correctness and formatting quality. The rewards are computed using RL with Verifiable Rewards (RLVR) (Shao et al., 2024), ensuring that they are deterministic and verifiable.

Despite the improvements introduced by GRPO, there are two key issues identified and analyzed in this work: *(i)* undesired *gradient conflicts* affect potentially correct tokens shared across multiple completions of the same group that receive both positive and negative advantages, and *(ii)* a *policy collapse* phenomenon, defined as a degradation in policy performance during training, in which negatively rewarded completions may penalize confident responses and flatten tokens likelihood distribution, shifting model decisions toward unlikely tokens.

To address these issues, we propose *Group-relative Trajectory-based Policy Optimization* (GTPO), which treats the sequence of generated tokens (i.e., the completion) as a trajectory of decisions taken by the LLM policy. The core idea behind GTPO is to control gradient conflict and entropy explosion while keeping the average output entropy from becoming too low, so that exploration is not prematurely suppressed. To do so, GTPO applies a *conflict-aware gradient correction* mechanism that mitigates gradient conflicts on shared tokens, particularly in the ini-

*Equal contribution.
[1] Github Repo

Transactions of the Association for Computational Linguistics, vol. 14, pp. 2258–2280, 2026. https://doi.org/10.1162/TACL.a.800
Action Editor: Dilek Hakkani-Tur. Submission batch: 5/2025; Revision batch: 12/2025; Published 8/2026.

tial and final parts of completions, thus preserving consistency across trajectories. Furthermore, we show that while the Kullback–Leibler (KL) divergence of GRPO often reacts too slowly in preventing policy collapse, monitoring the average entropy of the output tokens distribution offers a clearer signal of training instability. Building on this, entropy-based regularization terms are applied in GTPO to control the exploration of trajectories in the same group, also filtering out completions (regardless of their corresponding advantage) whose entropy exceeds a provable threshold. In this way, GTPO prevents the penalization of important tokens and mitigates policy collapse, improving both the formatting and accuracy of generated completions. Moreover, GTPO removes the KL divergence term used in vanilla GRPO, avoiding the reference model (Shao et al., 2024) and making training faster. In summary, the contributions of the work are:

- We identify two critical issues in GRPO: (i) gradient conflicts on tokens shared across completions within the same group, and (ii) a policy collapse phenomenon that exposes the limitations of the KL term.
- We introduce GTPO, a novel policy-based optimization that addresses the previous issues via *conflict-aware gradient corrections* and *entropy control*.
- We validate GTPO on mathematical reasoning tasks across multiple models and benchmarks, showing consistent gains over GRPO, DAPO (Yu et al., 2026), and SFT, improved training stability, and stronger in- and out-of-distribution performance.

The paper is structured as follows: Section 2 reviews related work, and Section 3 introduces key preliminaries. Section 4 analyzes GRPO's limitations, while Section 5 presents GTPO approach. Section 6 reports experimental results, and Section 7 concludes the paper.

2 Related Work

Reinforcement Learning in LLMs. RL has been widely adopted in decision-making tasks (Mnih et al., 2016, 2015; Berner et al., 2019), and nowadays is increasingly applied to the alignment and fine-tuning of LLMs. A prominent approach is RL from Human Feedback (RLHF), introduced in InstructGPT (Ouyang et al., 2022), and further developed by Anthropic (Bai et al., 2022). RLHF has become central to the training pipelines of state-of-the-art LLMs such as Claude 3 (Anthropic, 2024), Gemini (Anil et al., 2023), and GPT-4 (OpenAI, 2023). It typically includes supervised fine-tuning, a reward model, and the adoption of Proximal Policy Optimization (PPO) (Schulman et al., 2017). Here, PPO improves training stability by constraining updates through a clipped surrogate objective, offering a more practical alternative to Trust Region Policy Optimization (TRPO) (Schulman et al., 2015). However, it remains sensitive to reward scaling and can suffer from training instability (Wang et al., 2019; Garg et al., 2021; Moalla et al., 2024), requiring multiple refinements (Huang et al., 2022). Therefore, multiple variations have been proposed over the years, such as TRGPPO (Wang et al., 2019), alphaPPO (Xu et al., 2023), and PPO-ALR (Jia et al., 2024).

Advancements and Limitations in GRPO. To overcome the need for a critic model, Deepseek introduced GRPO (Shao et al., 2024; Guo et al., 2025), an approach that, given a question, compares multiple responses (completions) to derive relative rewards. GRPO demonstrates state-of-the-art performance on math benchmarks and achieves human-like alignment without relying on explicit manual feedback or critic networks (Li et al., 2025). Despite its promise, potential limitations of GRPO have been recently emerged, as bias effects (He et al., 2025), gradient imbalance (Yang et al., 2025b), which lead to undertraining of rare yet informative tokens (Liu et al., 2025), and degradations (or even collapse) of model performance, like in PPO (Dohare et al., 2023). Recently, Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) has been proposed as a GRPO variant that performs sequence-level optimization to improve stability and efficiency, while coinciding with GRPO at the first iteration. Similarly, DAPO (Yu et al., 2026) provides another alternative to further refine policy optimization strategies.

We identify two key issues in GRPO: token-level penalization and policy collapse, for which we show that KL divergence provides limited protection, while entropy-based analysis offers clearer corrective signals (Cui et al., 2025). These insights motivate GTPO, which improves stability and performance during training and evaluation.

3 Preliminaries

During training, given an input prompt q, the sampling LLM $\pi_{\theta_{\text{old}}}(\cdot \mid q)$ generates a group of G completions $\{o_i\}_{i=1}^{G}$, where each completion consists of tokens denoted by $o_{i,t}$. Each completion receives a scalar reward R_i, combining accuracy and formatting verifiable scores. These rewards are then normalized into advantages $\hat{A}_i = (R_i - \bar{R})/\text{std}(R)$, where $\bar{R}$ and $\text{std}(R)$ denote the mean and standard deviation of the R_i values within the group G. The GRPO objective is:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\bar{\mathcal{C}}_i - \beta \cdot D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right], \tag{1}$$

where $D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ refers to KL divergence scaled by β and $\bar{\mathcal{C}}_i$ is the average over tokens:

$$\bar{\mathcal{C}}_i = \sum_{t=1}^{|o_i|} \frac{\min\left(r_{i,t}(\theta)\,\hat{A}_i,\ \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_i\right)}{|o_i|}, \tag{2}$$

where the importance sampling ratio is defined as: $r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|s_{i,t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|s_{i,t})}$, where θ and θ_{old} denote the parameters of the current policy being optimized and the policy that generated the completions (Schulman et al., 2017). The state at step t is $s_{i,t} = (q, o_{i,<t})$, with $o_{i,<t}$ denoting the sequence of tokens generated before step t for completion i.

The policy π_θ is obtained from logits f_θ via $\pi_\theta(o_{i,t} \mid s_{i,t}) = \text{softmax}(f_\theta) \in \mathbb{R}^V$, where V is the vocabulary size. The KL term in Eq. (1) penalizes deviations from a reference policy π_{ref}, scaled by β. For clarity, in the gradient expression we omit the clipping and keep only the dependence on the ratio $r_{i,t}$ (full derivation in App. 9.1). Under the standard policy-gradient approximation,

$$\nabla_\theta \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\frac{\hat{A}_i}{|o_i|}\sum_{t=1}^{|o_i|} g_{i,t} - \beta \cdot \nabla_\theta \text{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right)\right] \tag{3}$$

Where the token-level contribution $g_{i,t}$ depends only on the ratio and on the policy gradient:

$$g_{i,t} := r_{i,t}(\theta)\,\nabla_\theta \log \pi_\theta(o_{i,t} \mid s_{i,t}) \tag{4}$$

Writing $g_{i,t}$ in terms of logits, let j be the index of $o_{i,t}$, with probability π_θ^j; then

$$g_{i,t} = r_{i,t}(\theta)\left[\nabla_\theta f_\theta^j (1 - \pi_\theta^j) - \sum_{k \neq j} \pi_\theta^k \nabla_\theta f_\theta^k\right] \tag{5}$$

A positive normalized advantage $\hat{A}_i > 0$ increases the logit of the selected token (scaled by $r_{i,t}$) while decreasing those of its alternatives; negative advantages have the opposite effect.

4 GRPO Issues

We highlight two major limitations of GRPO: *token-level penalization* and *policy collapse*.

4.1 Token-level Penalization

In this section, we show that GRPO can negatively affect updates to tokens shared across completions in the same group, even when those tokens are crucial for maximizing reward.

Prefix Tokens Penalization. Given a prompt q, the model generates a set of completions $\{o_i\}_{i=1}^{G}$, which may begin with the same sequence of tokens and then diverge at a specific point, a token that acts as a crossroads. This behavior is typical in instruction-tuned models, where early tokens tend to be similar across completions (Wang et al., 2024). However, not all completions in the group G necessarily share the same prefix. In practice, we often observe multiple distinct prefixes, each shared by a subset of the completions. To model this, we partition the G completions into K disjoint groups $\mathcal{G}_1, \ldots, \mathcal{G}_K$ so that all completions in a group $\mathcal{G}_k$ have a common prefix $S_{\text{pfx}}^{(k)}$. If all G completions share the same prefix, then $K = 1$. Tokens in $S_{\text{pfx}}^{(k)}$ are affected by the combined advantages of all completions in the group, while other tokens only depend on the advantage of their own completions. Under this structure, we rewrite the GRPO gradient (Eq. 3) by summing the contributions of all groups, where each group contributes a prefix term and a set of individual terms. The gradient, excluding the KL term, becomes:

$$\nabla_\theta \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{k=1}^{K}\left[\sum_{i\in\mathcal{G}_k}\left(\frac{\hat{\mathcal{A}}_i}{|o_i|}\sum_{j\notin S_{\text{pfx}}^{(k)}} g_{i,j}\right) + \underbrace{\sum_{i\in\mathcal{G}_k}\left(\frac{\hat{\mathcal{A}}_i}{|o_i|}\sum_{t\in S_{\text{pfx}}^{(k)}} g_{i,t}\right)}_{\Delta_{\text{pfx}}^{(k)}}\right]\right] \tag{6}$$

The term $\Delta_{\text{pfx}}^{(k)}$ represents the gradient update associated with the shared prefix of group $\mathcal{G}_k$. If a completion does not share any prefix with others, then $\mathcal{G}_k$ contains only that single completion, resulting in $S_{\text{pfx}}^{(k)} = \emptyset$ and $\Delta_{\text{pfx}}^{(k)} = 0$. Since the log-probability terms $g_{i,t}$ are identical and so constant

across completions in the group $S^{(k)}_{\text{pfx}}$, the gradient component Δ_{pfx} is primarily driven by the sum of normalized advantages $\sum_{i \in \mathcal{G}_k} \hat{\mathcal{A}}_i / |o_i|$. This implies that the update to a prefix mainly depends on the relative balance of advantages across all completions in group k. For instance, the sum can be negative, when correct completions (with positive advantages) are generally longer than incorrect ones. In such cases, the denominator $|o_i|$ for the correct completions becomes larger, reducing their contribution to the overall gradient. As a result, the model may be penalized for generating desirable and beneficial prefix tokens. This introduces a bias against shared initial tokens, such as formatting tokens or reasoning tags (`<reasoning>`), essential for the response structure and correctness.

Dependencies From Completion Advantage. GRPO can also induce undesirable penalization on tokens that appear after the prefix. Certain tokens, in particular formatting ones, may occur in multiple completions at varying positions and within different contexts, some correct, others incorrect, making them more susceptible to inconsistent and potentially harmful updates. This induces potential issues in the update. For instance, if a token τ frequently appears in completions with negative advantage, its probability may be reduced, even if the token is syntactically correct and required. Additionally, further issues can arise when a completion only partially follows the expected format. For example, if a completion closes a first part correctly with a formatting token τ (e.g. `</reasoning>`) but then omits `<answer>`, τ may receive a low or even negative reward. This, in turn, penalizes `</reasoning>`, despite it being a correct token in the completion. This occurs because the update for each token primarily depends on the advantage assigned to the entire completions in which it appears, without directly considering whether the token's individual contribution was beneficial. This phenomenon is especially impactful for formatting tokens and shared final tokens (common suffix) across completions.

4.2 Policy Collapse

The GRPO gradient, Eq. (3), highlights a dependency on both the advantage and the probability scores assigned by the policy π_θ. To gain a deeper understanding of how the advantage influences the learning dynamics, we analyze the average Shannon entropy of the output distribution π_θ over completions i and tokens t during training. The Shannon entropy is defined as $H(\mathbf{p}) = -\sum_{j=1}^{n} p_j \ln p_j$, where $\mathbf{p}$ is a probability vector. For a token position t in a completion o_i, we consider $\mathbf{p} = \pi_\theta(o_{i,t}|s_{i,t})$, and we define $\langle H \rangle_i$ as the average entropy across in the completion o_i. At a generation step, let us consider a low entropy (e.g., $\langle H \rangle_i \ll \ln 2$) [2] , indicating that the model is highly confident, i.e., it assigns almost all probability mass to a single token index j, with $\pi^j_\theta \approx 1$ and $\pi^k_\theta \approx \epsilon \ll 1$, where $k \neq j$. If this token leads to a negative outcome ($\hat{\mathcal{A}}_i < 0$), GRPO penalizes it sharply, since the gradient of the GRPO loss at index τ for completion i becomes:

$$\nabla_\theta \mathcal{J}^{(i,\tau)}_{\text{GRPO}}(\theta) \approx \begin{cases} -\dfrac{|\hat{\mathcal{A}}_i|}{|o_i|} r_{j,i}(\theta) \nabla_\theta f^j_\theta \cdot (1-\epsilon) & \text{for } j \\ \dfrac{|\hat{\mathcal{A}}_i|}{|o_i|} r_{k,i}(\theta) \nabla_\theta f^k_\theta \cdot \epsilon & \text{for } k \neq j \end{cases} \tag{7}$$

Note that the KL term has been omitted here, its contribution will be addressed next. Both $r_{k,i}(\theta)$ and $r_{j,i}(\theta)$ are equal to 1 at the first iteration and remain approximately 1 in the subsequent ones. This results in a negative update for the selected token j, and small positive updates for all other tokens $k \neq j$, even if they are potentially syntactically or semantically incorrect. While each of these positive updates is small, they can accumulate over time, gradually increasing the probability of initially implausible tokens. The effect, necessary for exploration (Cui et al., 2025), can be further amplified by *GRPO's zero-mean constraint* (the average of the advantages of the group $\bar{A} = 0$): when most completions receive positive rewards, the few with negative rewards must carry disproportionately large negative advantages to maintain balance. These penalties can suppress correct predictions and unintentionally amplify the likelihood of undesirable alternatives, raising entropy and introducing instability. Over time, this harms output quality and increases the risk of introducing structural and semantic errors that propagate through subsequent training steps. As the model drifts away from desirable behaviors, the reward signal becomes less meaningful. This phenomenon, which describes when high entropy can be harmful in GRPO, is illustrated in the top plot of Figure 1, which shows the average formatting reward during the training of LLaMA 8B (Patter-

[2] An average entropy close to $\ln 2$ indicates that the model is choosing between two equiprobable tokens at each step.

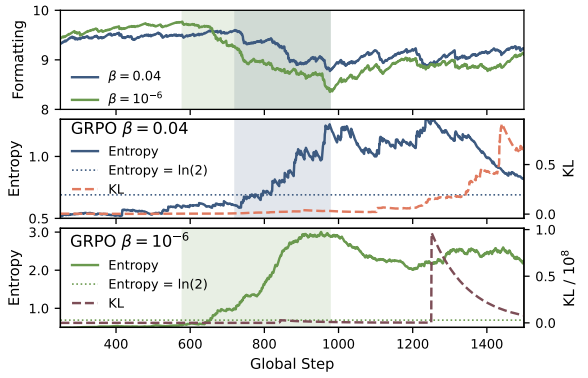


Figure 1: GRPO training of LLaMA 8B on GSM8K $G = 8$ comparing $\beta = 0.04$ and $\beta = 10^{-6}$. Plots show formatting reward (top), average entropy (middle), and KL divergence (bottom).

son et al., 2022) on GSM8K (Hendrycks et al., 2021a), using GRPO with $\beta = 0.04$ (as originally proposed in (Shao et al., 2024)) and $\beta = 10^{-6}$. At the beginning, the formatting reward increases more rapidly with $\beta = 10^{-6}$, while it remains relatively flat with $\beta = 0.04$. However, in both cases, the reward begins to drop below 9, after 580 steps for $\beta = 10^{-6}$ and around step 750 for $\beta = 0.04$.

KL Reacts Late to Avoid Collapse. Middle and bottom plots of Figure 1 show the average entropy $\langle H \rangle_i$ and average KL divergence, computed on the generated completions, for the two previous training runs of LLaMA 8B. Both KL divergence curves remain stable up to approximately step 1200, well after the degradation points, when the formatting rewards have already collapsed and entropy levels are high. Only around the KL peak, the reward curves show signs of stabilization. In contrast, the entropy curves begin to rise sharply as the formatting rewards start to decline, and continue to increase as the rewards deteriorate further. This suggests that the KL divergence acts as a delayed corrective signal, rising only after collapse. In contrast, entropy tracks the policy collapse in real time, increasing as the policy loses structure.

5 Method

This section presents GTPO, which mitigates gradient conflicts and prevents policy collapse.

5.1 Conflict-Aware Gradient Correction

As discussed in Section 4.1, shared tokens often suffer from gradient conflicts due to mixed advantages. To address this, we selectively mask gradient updates for these conflicting tokens. First, we

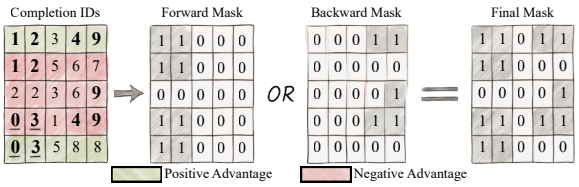


Figure 2: The numbers in the Completions represent token IDs, with mask values of 1 identifying conflict tokens and 0 indicating non-conflict ones.

define *conflict tokens* as tokens sharing the same token index (ID) that appear at the exact same position, either from the beginning or the end, across completions with positive and negative advantages (Figure 2). As discussed in Section 4.1, these tokens often receive conflicting gradient updates during training. Then, we mitigate such conflicts by correcting their gradient accordingly.

Conflict tokens definitions. Let $\{o_i\}_{i=1}^{G}$ be a group of completions. Each completion o_i contains a sequence of tokens $\{o_{i,t}\}_{t=1}^{|o_i|}$, and is associated with an advantage value $A^{(i)} \in \mathbb{R}$. We define G^- and G^+ as the sets of completions with $\mathcal{A} < 0$ and $\mathcal{A} > 0$, respectively.

Left-to-right alignment. A token $v \in \mathcal{V}$ is a *forward conflict token* at position p if it appears at the same position in at least one completion with positive advantage and in at least one with negative:

$$\exists\, i \in G^+ :\ o_{i,p} = v \quad \wedge \quad \exists\, j \in G^- :\ o_{j,p} = v.$$

Right-to-left alignment. A token v is defined as a *backward conflict token* at offset r if it occurs at the r-th position from the end in at least one completion with a positive advantage and in at least one with a negative advantage:

$$\exists\, i \in G^+ :\ o_{i,|o_i|-r} = v \quad \wedge \quad \exists\, j \in G^- :\ o_{j,|o_j|-r} = v.$$

Gradient Reweighting with conflict masks. We construct binary masks for tokens across completions to target positions potentially affected by gradient conflicts and thus correct their updates. An illustration of the following procedure is provided in Figure 2. We first define a *forward mask* $\mathcal{M}_i^{\text{fw}} \in \{0,1\}^{|o_i|}$ by scanning o_i from left to right, setting 1 over the first contiguous span of forward conflict tokens and 0 elsewhere. The *backward mask* $\mathcal{M}_i^{\text{bw}}$ is obtained by scanning from right to left, marking the first contiguous span of backward conflict tokens. A *final mask* $\mathcal{M}_i$ is then defined as: $\mathcal{M}_i = \mathcal{M}_i^{\text{fw}} \vee \mathcal{M}_i^{\text{bw}}$, highlighting only the

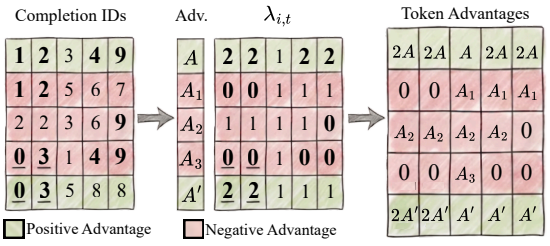


Figure 3: Creation of token advantages in GTPO

initial and final conflict regions in each completion o_i. After computing each $\mathcal{M}_i$, we correct inconsistent gradient updates over the selected conflict tokens, while leaving other token updates unchanged, through a token-level loss:

$$\mathcal{J}^* = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{\mathcal{A}_i}{|o_i|}\sum_{t=1}^{|o_i|} r_{i,t}\lambda_{i,t}\right] \tag{8}$$

where $\lambda_{i,t}$ controls the update of each token based on its conflict position and the sign of $\mathcal{A}_i$:

$$\lambda_{i,t} = \begin{cases} 1 & \text{if } \mathcal{M}_{i,t} = 0, \\ 0 & \text{if } \mathcal{M}_{i,t} = 1 \text{ and } \mathcal{A}_i < 0, \\ 2 & \text{if } \mathcal{M}_{i,t} = 1 \text{ and } \mathcal{A}_i > 0. \end{cases} \tag{9}$$

As shown in Fig. 3, the mask disables negative gradients on conflict tokens, reinforcing them only within positively rewarded completions. The total signal magnitude is preserved across the group: since $\sum_{i\in G^+}|\mathcal{A}_i| = \sum_{i\in G^-}|\mathcal{A}_i|$, doubling the signal for positive completions compensates for the removed negative updates, ensuring training stability and preventing semantic drift. Conversely, outer spans typically correspond to formatting tags, which is the primary source of conflict in GRPO (Section 4.1). From a computational perspective, this masking mechanism introduces minimal overhead, bounded by a spatial and temporal complexity of $\mathcal{O}(G^2L)$, where L is the maximum sequence length. Full derivations of the weighting scheme and formal complexity analyses are provided in Appendices 9.2 and 9.3, respectively, while empirical benefits are evaluated through ablation studies (Section 6.4).

5.2 Entropy Control

As discussed in Section 4.2, GRPO can lead to policy collapse, where standard KL term may react too slowly. To address this, we propose entropy-based regularization terms during training. These consist of two key parts: (i) a filtering mechanism to discard unstable completions, and (ii) a regularization term that penalizes high-entropy behavior.

Completion filter. Based on the policy-collapse analysis, high-entropy completions can jeopardize training by signaling structural uncertainty. Applying gradients to them risks accelerating collapse. To mitigate this, we filter out these unstable completions using a threshold of $\ln 2 \approx 0.69$, which mathematically represents a *"coin-flip"* uncertainty between two equiprobable tokens. [3] Let $\langle H\rangle_i$ denote the average token entropy of the i-th completion, and let $\langle H\rangle_{\text{ini}}$ represent its overall average prior to training over a small sample set. If $\langle H\rangle_{\text{ini}} < \ln 2$, we assume the model naturally produces low-entropy (stable) outputs, making it highly sensitive to derailed trajectories. In this case, we apply an entropy-based mask δ_i to filter out the advantage signals of completions exceeding the threshold:

$$\delta_i = \begin{cases} 1, & \text{if } \langle H\rangle_{ini} > \ln 2, \\ 0, & \text{if } \langle H\rangle_{ini} < \ln 2 \text{ and } \langle H\rangle_i > \ln 2, \\ 1, & \text{if } \langle H\rangle_{ini} < \ln 2 \text{ and } \langle H\rangle_i \le \ln 2. \end{cases} \tag{10}$$

Entropy Regularization. Inspired by PPO (Andrychowicz et al., 2021; Huang et al., 2022), we add a regularization term based on the average tokens entropy of each completion, $\langle H\rangle_i$, where γ balance the importance of this term in the final loss, as shown below. Note that, based on the internal characteristics of GRPO to implicitly increase entropy over time, we decided to minimize the term. This acts as a way to reduce the model entropy over time. The combination of these entropy-based strategies and the token-level loss defines the final GTPO objective, as:

$$\mathcal{J}_{\text{GTPO}} = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{\delta_i\cdot(\mathcal{A}_i - \gamma\cdot\langle H\rangle_i)}{|o_i|}\sum_{t=1}^{|o_i|} r_{i,t}\lambda_{i,t}\right] \tag{11}$$

Unlike vanilla-GRPO (Shao et al., 2024), the GTPO objective omits the KL divergence term and the reference model, enabling faster and lighter training. Although previous works (Hu et al., 2025; Liu et al., 2025) attempted removing this term from GRPO, our experiments (Section 6)

[3] We treat $\ln 2$ as a conservative hyperparameter balancing exploration and stability: higher thresholds allow more exploration but increase the risk of policy collapse, requiring expensive checkpoint rollbacks to recover from.

demonstrate that KL regularization often remains necessary for GRPO's stability.

6 Experiments

The experimental analysis is organized as follows: Section 6.1 compares GTPO with other methods across benchmarks; Section 6.2 examines the impact of conflict tokens; Section 6.3 investigates catastrophic forgetting utilizing the non-mathematical domains; and Section 6.4 reports ablation studies of the GTPO objective.

Concerning experimental setup, we conducted our experiments using LLaMA 8B and Qwen 2.5 (3B) (Yang et al., 2025a), finetuned on the GSM8K, MATH (Hendrycks et al., 2021b), and the R1 dataset released by DAPO[4]. These datasets were selected to present a progressive gradient of difficulty, from the easier GSM8K to the harder R1, designed to challenge the reasoning capabilities of the chosen models. For GSM8K and MATH, we assessed both in-distribution performance on their respective test splits and out-of-distribution (OOD) generalization on the AIME 2024, AIME 2025, and AMC 2023 benchmarks. Conversely, models trained on R1 were evaluated exclusively on OOD benchmarks. To provide comprehensive comparisons, we evaluated our proposed GTPO against SFT, DAPO, and GRPO baselines. SFT and GRPO underwent the full evaluation suite, whereas DAPO was assessed solely in OOD settings. For the GRPO style methods (Fontana et al., 2026), we explored generation sizes of $G = 8$ and $G = 12$, restricting R1 to $G = 8$ due to its larger scale. Additionally, GRPO was tested with KL penalty coefficients of $\beta \in \{0, 10^{-6}\}$, where $\beta = 10^{-6}$ yielded the best preliminary results. Regarding GTPO, we estimated the initial entropy, $\langle H \rangle_{\text{ini}}$, using the first 100 training samples and set $\gamma = 0.1$ to mitigate entropy suppression (see Figure 10a). To minimize computational overhead, we adopted a single-iteration update (Simoni et al., 2025; Fontana et al., 2026) where $\pi_\theta = \pi_{\theta_{\text{old}}}$. This approach renders clipping unnecessary, reducing the objective to Eq. 14. All experiments were executed on two A100 GPUs, a learning rate of 10^{-6} and a temperature of 1.0; further hyperparameters are detailed in Appendix 9.2.

[4]Models versions are meta-llama/Llama-3.1-8B-Instruct and Qwen/Qwen2.5-3B-Instruct, while DAPO and R1 training curves are provided in Appendix 10.4 and in Appendix 10.5.

6.1 Performance Evaluation

Training Dynamics of GTPO and GRPO. In Figure 4-top (the first two rows), we compare the training stability and performance of GTPO against GRPO. Formatting and accuracy rewards are reported as percentages, where the maximum value (100%) corresponds to a reward of 10 in both. On the GSM8K dataset with LLaMA, GTPO consistently outperforms GRPO across all training steps in both accuracy and formatting metrics. On the more challenging MATH dataset, GRPO (with $G = 12$ and $\beta = 0$) initially achieves slightly better accuracy around the midpoint of training. However, its performance drops sharply in the second half of training due to policy collapse, affecting both accuracy and formatting. In contrast, GTPO continues to improve steadily throughout training, avoiding collapse and maintaining stable performance. For Qwen 2.5, on both GSM8K and MATH, GTPO achieves comparable or improved accuracy relative to GRPO, with only a slight decrease in formatting performance (still above 97% in all runs). Note that, consistent with the analysis in Section 4.2, GRPO training curves for Qwen 2.5 are not affected by a policy collapse, as it shows high-entropy behavior. Overall, GTPO has more stable and reliable training compared to GRPO (see Fig. 11 in Appendix 10 for more details).

In-distribution Evaluation. The trained LLMs were evaluated on the test sets of GSM8K and MATH using the `pass@k` (Chen et al., 2021) and `maj@k` (Wang et al., 2023) metrics. The former measures whether at least one of the top-k completions yields a correct answer, while the latter assesses correctness via majority voting over the top-k completions. Note that, to ensure a fair comparison, we evaluated GRPO on LLaMA using the model checkpoint corresponding to the training step with the highest accuracy reward, rather than the one obtained after policy collapse. Few curves are omitted because, even at their best checkpoint, the corresponding models were too unreliable and achieved consistently low performance. Figure 4 bottom (the last two rows) shows that GTPO consistently outperforms GRPO in almost all settings for both `pass@k` and `maj@k`, as k varies from 1 to 32. This indicates that models trained with GTPO exhibit stronger self-consistency when answering questions (higher `maj@k`) and better cov-

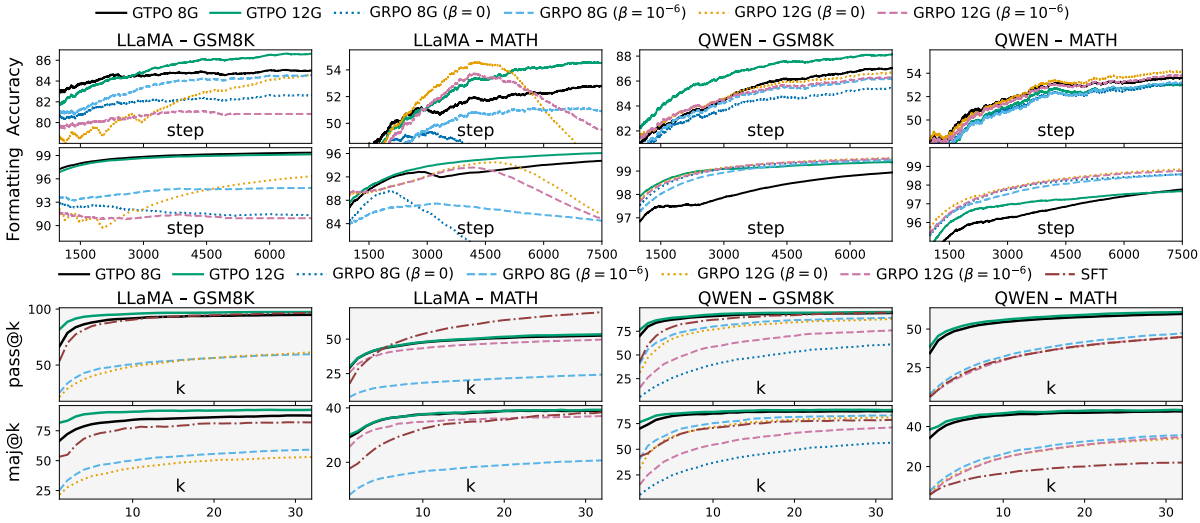


Figure 4: *Training accuracy and formatting rewards* (%) of GTPO and GRPO over training steps on MATH and GSM8K (top). **In-distribution evaluation** of models trained with GTPO, GRPO, and SFT on the corresponding test sets, using `pass@k` and `maj@k` (%) over k (bottom).

erage of the correct answer across multiple completions (higher `pass@k`). We also include testing comparisons with SFT, where GTPO consistently outperforms SFT in `maj@k` across all models and datasets, and achieves higher average performance in `pass@k`. Specifically, SFT surpasses GTPO only in `pass@k` for $k > 5$ on MATH with LLaMA, but not in terms of correctness with `maj@k`. Interestingly, in GTPO, larger values of G always lead to better performance, which is not the case for GRPO; in most cases, the latter benefits from a value of $\beta = 10^{-6}$ compared to $\beta = 0$.

Out-of-distribution Evaluation. We further evaluate the models on three out-of-distribution (OOD) benchmarks: AIME 2024, AIME 2025, and AMC 2023. We report `pass@`k for $k \in \{1, 8, 16, 32, 64\}$, alongside `maj@`$k$ for AMC (substituting $k = 4$ for $k = 1$, as `maj@1` equates to `pass@1`). To aggregate and simplify comparisons across these diverse scenarios, the final column, denoted *"Wins"*, reports the number of instances in which each method achieves the highest score. For GRPO and GTPO, we evaluate the variants utilizing the generation size (G) that maximized in-distribution `pass@`k on GSM8K and MATH. Conversely, for DAPO and models trained on the R1 dataset, we select the configurations that yielded the highest training reward. Overall, GTPO demonstrates the strongest OOD generalization across both architectures, returning 42/60 wins for LLaMA and 40/60 for Qwen. When trained on GSM8K and MATH, LLaMA-GTPO consistently outperforms SFT and either matches or surpasses both GRPO and DAPO across nearly all k values, with a minor exception on AIME 2025 ($k = 64$, MATH training) where SFT marginally leads. Within the Qwen family, GTPO excels heavily on the AMC benchmark. DAPO, however, emerges as a robust competitor on AIME, frequently matching or exceeding GTPO (e.g., AIME 2024 and 2025 following GSM8K training). While GRPO and DAPO occasionally rival GTPO at large sample sizes (e.g., AMC `pass@64` with MATH training), SFT remains the weakest baseline overall, showing competitive results only at large k. A different trend occurs with the R1 dataset. Its inherent complexity causes GRPO style methods to struggle; while SFT proves more resilient on this specific data, its absolute performance still falls short of GTPO trained on simpler datasets. As detailed in Section 6.3, any reliance on SFT incurs the severe penalty of catastrophic forgetting.

6.2 Understanding Gradient Conflicts

To understand how gradient conflicts affect training, we analyzed the logs from all training runs and averaged them. At each step, given G completions and their advantages, we computed the conflict mask $\mathcal{M}_i$, and the *Net Impact Signal* $\mathcal{S}_{\text{net}}$ for all conflict token positions $\mathcal{K}_t$ as $\mathcal{S}_{\text{net}}(t) = \frac{1}{|\mathcal{K}_t|} \sum_{(i,t) \in \mathcal{K}_t} \lambda^*_{i,t} \frac{A_i}{|o_i|}$, where A_i is the advantage of completion i, $|o_i|$ is its length, and

Model	Dataset	Method	AIME 2024 Pass@k (%) 1	8	16	32	64	AIME 2025 Pass@k (%) 1	8	16	32	64	AMC 2023 Pass@k (%) 1	8	16	32	64	AMC 2023 Maj@k (%) 4	8	16	32	64	Wins (/20)
LLAMA	GSM8K	SFT	0.0	0.0	0.0	13.3	20.0	0.0	0.0	0.0	0.0	6.7	12.5	37.5	42.5	57.5	72.5	10.0	15.0	15.0	20.0	17.5	0
		GRPO	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	00	0.0	2.5	2.5	5.0	5.0	5.0	2.5	2.5	5.0	5.0	5.0	0
		DAPO	0.0	6.7	6.7	6.7	13.3	0.0	0.0	0.0	0.0	6.7	2.5	22.5	25.0	35.0	55.0	2.5	15.0	20.0	17.5	25.0	0
		GTPO	0.0	**10.0**	**13.3**	**20.0**	**30.0**	0.0	**6.7**	**6.7**	**6.7**	**13.3**	**17.5**	**47.5**	**55.0**	**65.0**	**80.0**	**25.0**	**27.5**	**35.0**	**37.5**	**42.5**	**18**
	MATH	SFT	0.0	3.3	10.0	20.0	26.7	0.0	0.0	3.3	6.7	**16.7**	10.0	45.0	60.0	65.0	80.0	12.5	15.0	12.5	15.0	17.5	1
		GRPO	0.0	**13.3**	**16.7**	20.0	36.7	0.0	0.0	0.0	3.3	6.7	20.0	37.5	40.0	52.5	70.0	25.0	30.0	30.0	30.0	32.5	2
		DAPO	0.0	0.0	10.0	13.3	13.3	0.0	0.0	0.0	0.0	6.7	2.5	22.5	25.0	35.0	55.0	2.5	15.0	20.0	17.5	25.0	0
		GTPO	**3.3**	**13.3**	**16.7**	**36.7**	**43.3**	0.0	0.0	**10.0**	**10.0**	13.3	**22.5**	**52.5**	**67.5**	**75.0**	**82.5**	**37.5**	**37.5**	**37.5**	**37.5**	**37.5**	**17**
	R1	SFT	0.0	0.0	3.3	**13.3**	**16.7**	0.0	**3.3**	**3.3**	3.3	3.3	**7.5**	27.5	**37.5**	**52.5**	**57.5**	**7.5**	**17.5**	**25.0**	**30.0**	**30.0**	**13**
		GRPO	0.0	0.0	3.3	10.0	13.3	0.0	0.0	0.0	0.0	3.3	2.5	22.5	25.0	35.0	55.0	2.5	15.0	20.0	17.5	25.0	0
		DAPO	0.0	**3.3**	**6.7**	6.7	13.3	0.0	0.0	**3.3**	**6.7**	**6.7**	**7.5**	25.0	35.0	45.0	55.0	**7.5**	**17.5**	17.5	22.5	22.5	8
		GTPO	0.0	0.0	3.3	6.7	**16.7**	0.0	0.0	0.0	0.0	**6.7**	**7.5**	**32.5**	32.5	50.0	55.0	**7.5**	**17.5**	17.5	22.5	**30.0**	7
		Overall Wins (Overall Percentage)						SFT: 14 (23.3%)					GRPO: 2 (3.3%)					DAPO: 8 (20.0%)					**GTPO: 42 (70.0%)**
QWEN	GSM8K	SFT	0.0	0.0	0.0	0.0	6.7	0.0	0.0	0.0	6.7	6.7	0.0	10.0	17.5	52.5	62.5	0.0	2.5	7.5	7.5	7.5	0
		GRPO	0.0	3.3	6.7	16.7	20.0	0.0	3.3	6.7	10.0	20.0	7.5	40.0	52.5	67.5	77.5	17.5	12.5	25.0	30.0	42.5	0
		DAPO	0.0	10.0	**20.0**	**23.3**	**30.0**	**0.3**	**13.3**	**13.3**	13.3	13.3	12.5	52.5	67.5	80.0	85.0	12.5	27.5	35.0	30.0	37.5	6
		GTPO	**3.0**	**13.3**	16.7	16.7	26.7	0.0	6.7	10.0	**20.0**	**26.7**	**37.5**	**65.0**	**70.0**	**85.0**	**95.0**	**40.0**	**42.5**	**47.5**	**50.0**	**45.0**	**14**
	MATH	SFT	0.0	0.0	0.0	0.0	6.7	0.0	0.0	0.0	0.0	3.3	0.0	12.5	35.0	47.5	65.0	0.0	2.5	7.5	7.5	7.5	0
		GRPO	0.0	3.3	3.3	6.7	16.7	0.0	3.3	**10.0**	**13.3**	16.7	30.0	67.5	70.0	77.5	**87.5**	27.5	42.5	**42.5**	45.0	42.5	4
		DAPO	0.0	**10.0**	**20.0**	**23.3**	**30.0**	0.0	3.3	3.3	3.3	6.7	12.5	52.5	67.5	80.0	85.0	12.5	27.5	35.0	30.0	37.5	4
		GTPO	**3.3**	6.7	13.3	20.0	**30.0**	**3.3**	**10.0**	**10.0**	**13.3**	**23.3**	**40.0**	**75.0**	**80.0**	**82.5**	85.0	**52.5**	**47.5**	**42.5**	**50.0**	**50.0**	**16**
	R1	SFT	0.0	6.7	20.0	20.0	23.3	0.0	**6.7**	**6.7**	**10.0**	**10.0**	**12.5**	**52.5**	65.0	75.0	**85.0**	**12.5**	**35.0**	**40.0**	**40.0**	**47.5**	**12**
		GRPO	0.0	**10.0**	**20.0**	**23.3**	**30.0**	0.0	3.3	3.3	3.3	6.7	**12.5**	**52.5**	**67.5**	**80.0**	**85.0**	**12.5**	27.5	35.0	30.0	37.5	10
		DAPO	0.0	**10.0**	**20.0**	**23.3**	**30.0**	0.0	3.3	3.3	3.3	6.7	**12.5**	**52.5**	**67.5**	**80.0**	**85.0**	**12.5**	27.5	35.0	30.0	37.5	10
		GTPO	0.0	**10.0**	**20.0**	**23.3**	**30.0**	0.0	3.3	3.3	3.3	6.7	**12.5**	**52.5**	**67.5**	**80.0**	**85.0**	**12.5**	27.5	35.0	30.0	37.5	10
		Overall Wins (Overall Percentage)						SFT: 12 (20.0%)					GRPO: 14 (23.3%)					DAPO: 21 (35.0%)					**GTPO: 40 (66.7%)**

Table 1: Qwen and LLaMA 8B **out-of-distribution evaluation** using the best of GTPO, GRPO, and SFT on AIME 2024/25 and AMC, reporting `pass@k` (and `maj@k` for AMC), with $1 \leq k \leq 64$. Wins denotes the total count of top scores, shown in bold, achieved by each method in a given row.

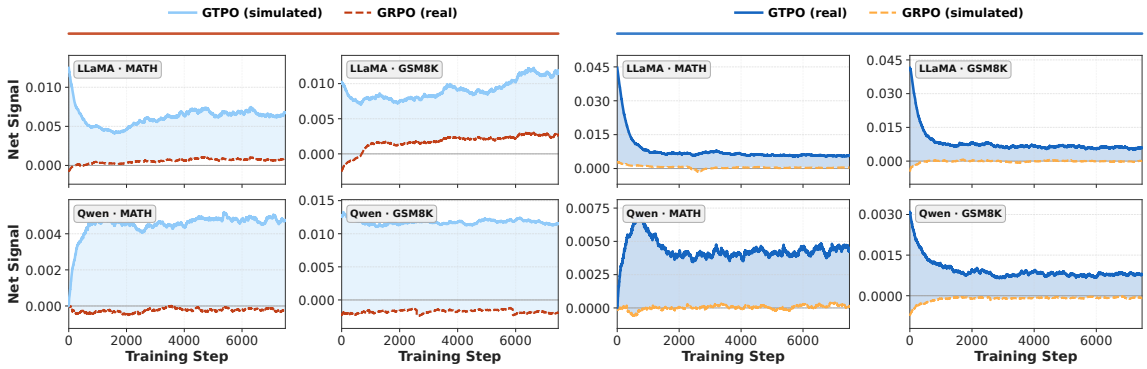


Figure 5: Net Impact Signal of GRPO and GTPO, comparing real and simulated updates on conflict tokens.

$\lambda^*_{i,t}$ is the algorithm-specific weighting factor. For standard GRPO, $\lambda^*_{i,t}$ is always 1. For GTPO, $\lambda^*_{i,t}$ dynamically changes to 0 (ignoring negative advantages) or 2 (doubling positive ones) on conflict tokens. $\mathcal{S}_{\text{net}}$ quantifies the effective learning signal; A near-zero or negative value indicates the erroneous penalization of shared tokens in successful responses. To make a fair comparison, Figure 5 uses a *what-if* scenario: the left panel takes a real GRPO training run and compares its actual signal to the simulated updates GTPO *would have* made (by applying its specific $\lambda^*_{i,t}$ values) on those exact same tokens. The right panel does the opposite, comparing a real GTPO run to simulated GRPO updates (by reverting $\lambda^*_{i,t}$ to 1). Across all settings, the net signal produced by GRPO is negligible, and even negative for Qwen, resulting in wasted optimisation updates on shared tokens. In contrast, GTPO consistently provides a positive learning signal, efficiently resolving these conflicts.

Figure 6 further supports this analysis by quantifying the number of conflict tokens for both LLaMA-3.1-8B and Qwen2.5-3B. Conflicts occur frequently during training, appearing 6–20 tokens per generation, with forward conflicts significantly outnumbering backward ones. Notably, the total number of conflict tokens is nearly identical for both GTPO and GRPO. This suggests that GTPO improves performance not by avoiding or increasing conflicts, but by effectively cor-

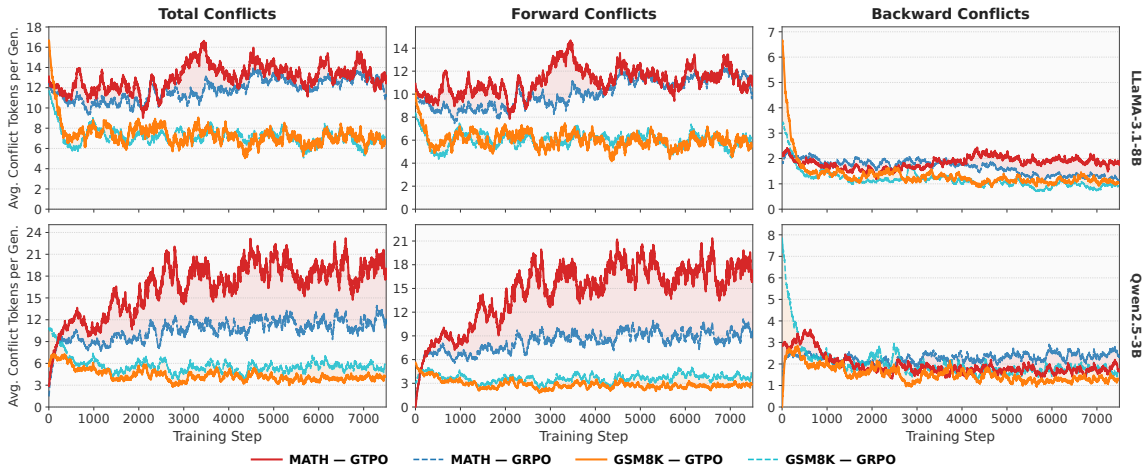


Figure 6: Evolution of Total, Forward, and Backward Conflict Tokens in GRPO and GTPO runs.

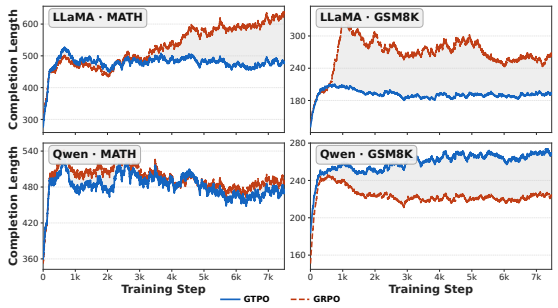


Figure 7: Evolution of completion lenghts.

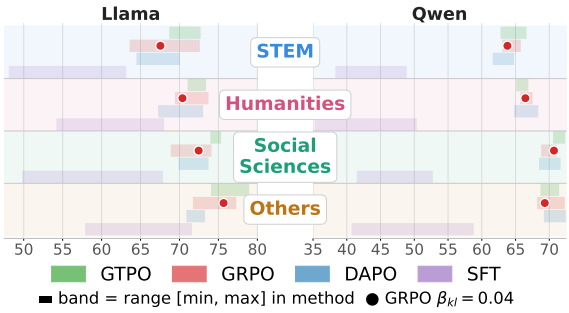


Figure 8: MMLU general knowledge performance across different training methods.

recting gradient updates when they occur. Figure 7 compares the completion lengths generated by GTPO and GRPO. Except for QWEN on GSM8K, GTPO generally produces shorter and more stable sequence lengths. This trend indicates that GTPO can enhance model capabilities without substantially increasing response lengths.

6.3 Preserving General Knowledge

To investigate whether training on the MATH and GSM8K datasets induces catastrophic forgetting of previously acquired knowledge, we evaluate the models on the MMLU benchmark (Hendrycks et al., 2020). We test unrelated domains by removing all math subjects and grouping the rest into four categories: STEM, Humanities, Social Sciences, and Others (see Appendix 10.6 for details). Figure 8 shows the average accuracy, with shaded bands indicating the variation across different training runs. The results show a clear difference between the training methods. GRPO, DAPO, and GTPO keep their general knowledge across all subjects. While GTPO scores slightly higher on average for LLaMA, the performance of all these methods is very similar. This tells us that simply using the RL framework is what protects the model from forgetting, rather than a specific feature of GRPO style methods (Fontana et al., 2026). In contrast, SFT suffers from severe *catastrophic forgetting*, performing much worse in every single category. Finally, we wanted to check if the KL divergence penalty (β) is the secret behind this stability. By evaluating LLaMA 8B and QWEN models trained on GSM8K and MATH (using $G = 8$ and a standard $\beta = 0.04$), we observe no discernible differences in knowledge retention, which indicates that KL-divergence regulation is not the primary mechanism mitigating catastrophic forgetting within GRPO.

6.4 Ablation Studies

Conflict-Aware gradient correction. Figure 9 shows the accuracy and formatting training curves of LLaMA on GSM8K. The model is trained using GRPO with KL β values set to 0, 0.04 (Shao et al., 2024), and 10^{-6}, while for GTPO, we use the full version (Eq. 11) and a variant with-

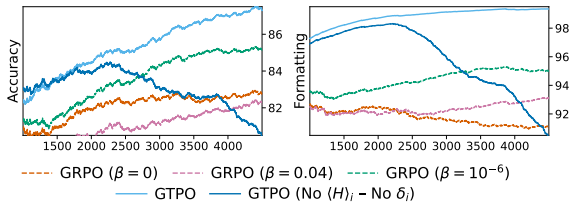


Figure 9: Accuracy and formatting of LLaMA on GSM8K. GRPO curves use different KL-β values, while *"No* $\langle H \rangle_i$ *– No* δ_i*"* for GTPO applies only the conflict-aware gradient correction.

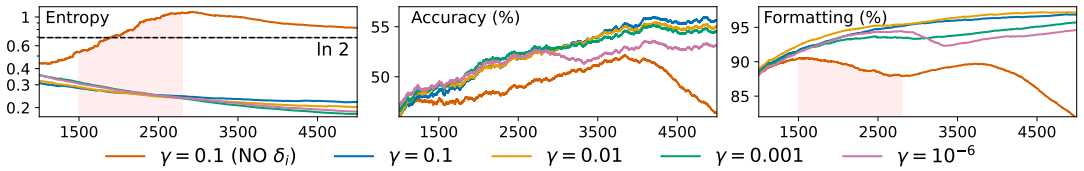
out entropy-based filtering and regularization (denoted as *"No* $\langle H \rangle_i$ *– No* δ_i*"* in the figure). This latter configuration isolates the effect of GTPO when relying solely on the *Conflict-Aware Gradient Correction* component (i.e., Eq. 8). As shown in the figure, GTPO outperforms GRPO in both accuracy and formatting during the first 2,500 steps. Beyond this point, GTPO with regularization and filtering continues to maintain better performance, whereas the variant without these components begins to degrade and eventually falls below GRPO. This behavior is expected, as the absence of regularization helps the model to prune high entropy completions. Most importantly, before the policy collapse, the use of gradient correction alone yields higher rewards than GRPO.

Impact of Entropy Regularization. Figure 10a shows the training curves for average completion entropy (left), accuracy (middle), and formatting (right) on LLaMA 8B trained on MATH. We compare several entropy regularization strengths ($\gamma = 0.1, 0.01, 0.001, 10^{-6}$) and a variant with $\gamma = 0.1$ but without entropy filtering ("*NO* δ_i"). Higher values of γ generally yield better accuracy and formatting, with $\gamma = 0.1$ performing best. When filtering is removed, however, both metrics collapse despite using the same γ, confirming that filtering high-entropy completions helps to reach training stability. The entropy curves further clarify this behavior. Without filtering, entropy continually rises above $\ln 2$, destabilizing formatting first and accuracy later. With filtering, entropy stays below $\ln 2$ and decreases smoothly. Moderate entropy (achieved with larger γ) supports exploration, producing more diverse and informative completions; excessively low entropy (e.g., with $\gamma = 0.001$) leads to overconfidence and early performance saturation. Finally, note that larger γ values tend to maintain slightly higher entropy. When a completion receives a negative advantage, entropy naturally increases (Eq. 8), and a stronger regularization term (Eq. 11) amplifies this effect by flattening the token distribution. However, this increase remains controlled and does not destabilize training; instead, it preserves a small amount of exploration, beneficial even in later stages.

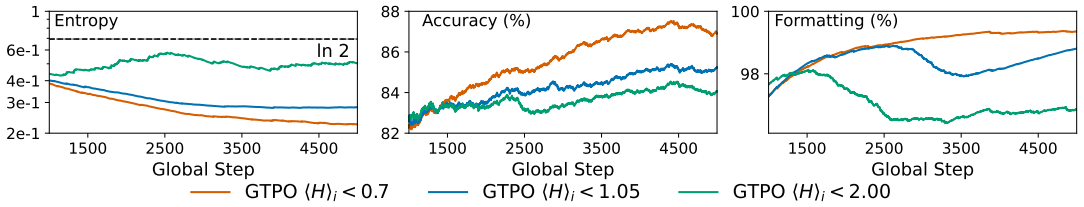
$\langle H \rangle_{\text{i}}$ Threshold in GTPO. Figure 10b analyzes the effect of different entropy thresholds $\langle H \rangle_{\text{i}}$ on GTPO training for LLaMA on GSM8K. We consider three filtering settings: 0.7, 1.05, and 2.0, which roughly correspond to output distributions over, respectively, 2, 3, and 4 equiprobable tokens (with no residual probability mass). With the stricter thresholds (0.7 and 1.05), the entropy plot shows that the average completion entropy remains low and stable, while the loose threshold (2.0) lets many high-entropy completions pass, causing entropy to drift toward the instability region around $\ln 2$ and destabilizing the policy. The accuracy curves mirror this behavior: tighter thresholds yield smoother improvements and higher final accuracy, whereas the loose threshold leads to stagnation. The formatting plot further confirms that insufficient filtering (as in the 2.0 setting) quickly degrades structural quality, whereas stricter thresholds preserve formatting and thus support both stability and output quality.

Effects of Negative-Only Entropy Filtering. Figure 10c compares GRPO against GTPO variants using Qwen on GSM8K dataset, analyzing different entropy filtering strategies. In *OnlyNeg* setting, the entropy threshold, tested at $\langle H \rangle_{\text{i}} < 2.00, < 1.05$ and < 0.7, is applied only to completions with negative advantages, meaning all positive completions are retained. Instead the standard approach for Qwen does not apply the completion filtering. The results show that *OnlyNeg* causes over-pruning, degrading performance: it drives entropy too low, restricting exploration, while failing to discard high-entropy positive completions that reinforce erroneous trajectories.

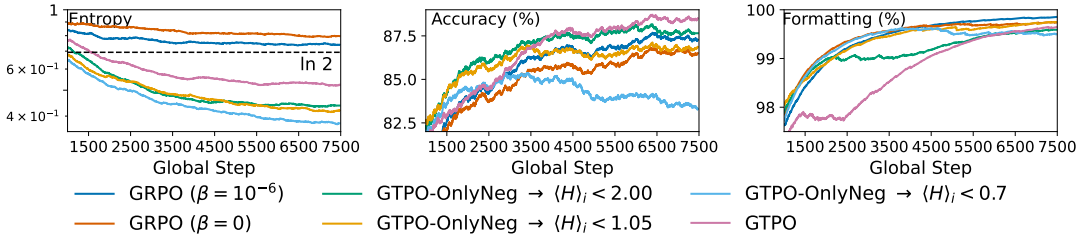
Effect of Regularization and Filtering on GRPO. Figure 10d compares LLaMA on GSM8K under three settings: (i) standard GRPO without KL ($\beta = 0$), (ii) GRPO with entropy filtering ($\langle H \rangle_i < 1.05$) and entropy regularization ($\gamma = 0.1$), and (iii) GTPO with the same entropy constraints. The entropy plot (left) shows that the baseline GRPO ($\beta = 0$) exhibits severe instability, characterized by entropy spikes followed by a sud-



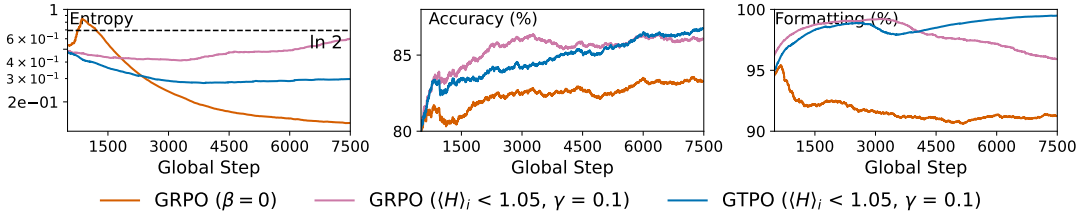
(a) LLAMA GTPO train on MATH with different γ: entropy, accuracy, and formatting. "NO δ_i" disables entropy filtering.



(b) Entropy, accuracy, and formatting in LLaMA trained with GTPO on GSM8K with different values of $\langle H \rangle_i$.



(c) QWEN on GSM8K: GRPO/GTPO with different $\langle H \rangle_i$ thresholds. *OnlyNeg* means $\langle H \rangle_i$ only for completions with $\mathcal{A}^-$.



(d) LLaMA trained on GSM8K with GRPO, and with GRPO/GTPO under the constraint $\langle H \rangle_i < 1.05$ and $\delta_i = 0.1$.

Figure 10: Comparison of entropy, accuracy, formatting under different training settings for GTPO and GRPO.

den collapse, which degrades both accuracy and structural formatting. Adding entropy regularization and filtering (roughly equivalent to bounding generation to three equiprobable tokens) significantly improves the situation. The entropy stays within a stable range, accuracy rises to reasonable levels, and formatting recovers. However, GTPO performs consistently better even under the same constraints. It maintains a lower and more stable entropy, achieves higher final accuracy, and preserves near-perfect formatting throughout training. In other words, while entropy control stabilizes GRPO, GTPO provides an additional layer of robustness and leads to the best performance.

7 Conclusion

In this work, we introduced Group-relative Trajectory-based Policy Optimization (GTPO), which addresses gradient conflicts and policy collapse in GRPO through conflict-aware updates and entropy control. We evaluated GTPO on mathematical reasoning tasks using LLaMA-8B and Qwen2.5-3B trained on GSM8K, MATH, and R1. Compared to SFT, GRPO, and DAPO, our method demonstrates improved training stability and stronger generalization across both in-distribution datasets (GSM8K, MATH) and challenging OOD benchmarks (AIME 2024/2025, AMC 2023). Additional analyses show that GTPO mitigates negative updates on conflict tokens, while MMLU results indicate that it preserves non-mathematical knowledge and mitigates catastrophic forgetting. Ablation studies further validate each component, showing that entropy regularization and completion filtering can independently improve GRPO stability. Future work will focus on establishing rigorous theoretical entropy bounds to guarantee effective exploration without compromising training stability. Additionally, we plan to investigate alternative value ranges for the token-update multiplier $\lambda_{i,t}$, exploring whether continuous or adaptive scaling can further optimize gradient control in conflict regions.

8 Acknowledgments

This paper has been supported by the TURING project, funded by the European Union's Horizon Europe research and innovation actions under grant agreement No 101215032.

References

Marcin Andrychowicz, Anton Raichuk, Piotr Stanczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. 2021. What matters for on-policy deep actor-critic methods? A large-scale study. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku — model card. Model card, Anthropic. Accessed: 2025-07-22.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. 2025. The entropy mechanism of reinforcement learning for reasoning language models.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Shibhansh Dohare, Qingfeng Lan, and A Rupam Mahmood. 2023. Overcoming policy collapse in deep reinforcement learning. In *Sixteenth European Workshop on Reinforcement Learning*.

Aleksandar Fontana, Marco Simoni, Giulio Rossolini, Andrea Saracino, and Paolo Mori. 2026. On the hidden objective biases of group-based reinforcement learning. *arXiv preprint arXiv:2601.05002*.

Saurabh Garg, Joshua Zhanson, Emilio Parisotto, Adarsh Prasad, J. Zico Kolter, Zachary C. Lipton, Sivaraman Balakrishnan, Ruslan Salakhutdinov, and Pradeep Ravikumar. 2021. On proximal policy optimization's heavy-tailed gradients. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 3610–3619. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Andre He, Daniel Fried, and Sean Welleck. 2025. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. *CoRR*, abs/2506.02355.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021a. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

Shengyi Huang, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang. 2022. The 37 implementation details of proximal policy optimization. In *ICLR Blog Track*. Https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/.

Lu Jia, Binglin Su, Du Xu, Yewei Wang, Jing Fang, and Jun Wang. 2024. Policy optimization algorithm with activation likelihood-ratio for multi-agent reinforcement learning. *Neural Process. Lett.*, 56(6):247.

Xuying Li, Zhuo Li, Yuji Kosuga, and Victor Bian. 2025. Optimizing safe and aligned language generation: A multi-objective GRPO approach. *CoRR*, abs/2503.21819.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1928–1937. JMLR.org.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller,

Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533.

Skander Moalla, Andrea Miele, Daniil Pyatko, Razvan Pascanu, and Caglar Gulcehre. 2024. No representation, no trust: Connecting representation, collapse, and trust issues in PPO. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. 2022. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org.

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

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

Marco Simoni, Aleksandar Fontana, Giulio Rossolini, and Andrea Saracino. 2025. Improving llm reasoning for vulnerability detection via group relative policy optimization. *arXiv preprint arXiv:2507.03051*.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024. "my answer is c": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 7407–7416. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yuhui Wang, Hao He, Xiaoyang Tan, and Yaozhong Gan. 2019. Trust region-guided proximal policy optimization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Decem-*

ber 8-14, 2019, Vancouver, BC, Canada, pages 624–634.

Haotian Xu, Zheng Yan, Junyu Xuan, Guangquan Zhang, and Jie Lu. 2023. Improving proximal policy optimization with alpha divergence. *Neurocomputing*, 534:94–105.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhihe Yang, Xufang Luo, Zilong Wang, Dongqi Han, Zhiyuan He, Dongsheng Li, and Yunjian Xu. 2025b. Do not let low-probability tokens over-dominate in RL for llms. *CoRR*, abs/2505.12929.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Appendix

9 Proofs and Repricability

9.1 Group-Relative Policy Optimization

The Role of KL Divergence on GRPO. We analyze the impact of the KL divergence term on the aggregated GRPO loss by transitioning from a token-level formulation to a trajectory-level one, when $r_{i,t} = 1$. Recall the normalized advantage function used in GRPO-like:

$$\hat{\mathcal{A}}_{i,t} = \hat{\mathcal{A}}_i = \frac{R_i - \bar{R}}{\mathrm{std}(R)}, \tag{12}$$

where R_i is the scalar reward associated with trajectory i, $\bar{R}$ is the average reward across all G trajectories in the batch, and $\mathrm{std}(R)$ denotes the standard deviation of the rewards.

Substituting Equation 12 into the GRPO objective from Equation 1, we obtain:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\frac{R_i - \bar{R}}{\mathrm{std}(R)} - \beta \cdot \mathrm{D}_{\mathrm{KL}}\left(\pi_\theta \| \pi_{\mathrm{ref}}\right)\Bigg] \tag{13}$$

We now turn our attention to the first term of the loss, which is defined as follows:

$$\tilde{\mathcal{J}}_{\mathrm{GRPO}}(\theta) := \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\frac{R_i - \bar{R}}{\mathrm{std}(R)}\right] \tag{14}$$

Since equation 12 is independent of t, it is possible to simplify $\tilde{\mathcal{J}}_{\mathrm{GRPO}}(\theta)$ as follows:

$$\begin{aligned}\tilde{\mathcal{J}}_{\mathrm{GRPO}}(\theta) &= \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\frac{R_i}{\mathrm{std}(R)} - \frac{\bar{R}}{\mathrm{std}(R)}\right)\right] \\ &= \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{\mathrm{std}(R)}\left(\frac{1}{G}\sum_{i=1}^{G}R_i - \bar{R}\right)\right] \\ &= \mathbb{E}_{q,\{o_i\}}[0].\end{aligned} \tag{15}$$

Thus, the overall GRPO objective reduces to:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\Big[-\beta \cdot \mathrm{D}_{\mathrm{KL}}\left(\pi_\theta \| \pi_{\mathrm{ref}}\right)\Big] \tag{16}$$

From this aggregated viewpoint, the loss is entirely governed by the KL divergence term. However, it is crucial to emphasize that a zero-valued aggregated reward term *does not imply that the gradient of the original GRPO loss is zero. The per-token advantages still guide parameter updates during optimization, ensuring meaningful learning even when the aggregated advantage cancels out.*

GRPO Gradient. Given the GRPO objective (see Equation 1), we aim to calculate the gradient.

$$\tilde{\mathcal{J}}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{\hat{\mathcal{A}}_i}{|o_i|}\sum_{t=1}^{|o_i|}\frac{\pi_\theta(o_{i,t} \mid s_{i,t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t} \mid s_{i,t})}\right] \tag{17}$$

To simplify the calculation, we keep $\hat{\mathcal{A}}_i$ in its unexpanded form (Eq. 12) and reduce the term $C_{i,t}$,(Eq. 2) to the ratio $r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|s_{i,t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t}|s_{i,t})}$. This substitution relies on the fact that the objective function's clipping mechanism eliminates gradient contributions from tokens where $|r_{i,t}(\theta)| > 1 + \epsilon$. Therefore, the summation is performed exclusively over the set of active (non clipped) tokens. Formalizing the excluded tokens adds notational burden with no impact on the resulting gradient analysis.

$$\begin{aligned}&\nabla_\theta \tilde{\mathcal{J}}_{\mathrm{GRPO}}(\theta) = \\ &\quad= \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{\hat{\mathcal{A}}_i}{|o_i|}\sum_{t=1}^{|o_i|}\nabla_\theta\left[\frac{\pi_\theta(o_{i,t} \mid s_{i,t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t} \mid s_{i,t})}\right]\right] \\ &\quad= \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{\hat{\mathcal{A}}_i}{|o_i|}\sum_{t=1}^{|o_i|}\frac{\pi_\theta(o_{i,t} \mid s_{i,t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t} \mid s_{i,t})}\right]\end{aligned}$$

We subsequently apply the log-derivative trick, a widely used technique in reinforcement learning, which reformulates the gradient as $\nabla_\theta \pi_\theta(x) = \pi_\theta(x)\nabla_\theta \log(\pi_\theta(x))$.

$$\begin{aligned}&\nabla_\theta \tilde{\mathcal{J}}_{\mathrm{GRPO}}(\theta) = \\ &\mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{\hat{\mathcal{A}}i}{|o_i|}\sum_{t=1}^{|o_i|}r_{i,t}(\theta)\cdot\nabla_\theta\left[\log(\pi_\theta(o_{i,t}|s_{i,t}))\right]\right]\end{aligned} \tag{18}$$

We can rewrite the complete gradient as:

$$\begin{aligned}&\nabla_\theta \mathcal{J}_{\mathrm{GRPO}}(\theta) = \\ &\mathbb{E}_{q,\{o_i\}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{\hat{\mathcal{A}}_i}{|o_i|}\sum_{t=1}^{|o_i|}r_{i,t}(\theta)\nabla_\theta\left[\log(\pi_\theta(o_{i,t}|s_{i,t}))\right] \\ &- \beta \cdot \nabla_\theta\left[\mathrm{D}_{\mathrm{KL}}\left(\pi_\theta \| \pi_{\mathrm{ref}}\right)\right]\Bigg]\end{aligned} \tag{19}$$

How gradients affect distribution over tokens. LLMs typically terminate with a softmax layer, which produces a probability distribution over the vocabulary. To simplify notation, we denote $(o_{i,t}|q, o_{i,<t})$ as (t'), yielding the following expression for the output distribution:

$$\pi_\theta(t') = \frac{e^{f_\theta^j}}{\sum_k^V e^{f_\theta^k}} \tag{20}$$

Here, f_θ^j denotes the chosen logit corresponding to position $o_{i,t}$, while $\sum_{k=1}^V f_\theta^k$ represents the aggregate logits over the entire vocabulary of size V, conditioned on the context $(q, o_{i,<t})$. Our objective is to compute the gradient in order to analyze its influence on the probability distribution of both the selected token and the unselected alternatives.

$$\begin{aligned}\nabla_\theta \log(\pi_\theta(t')) &= \nabla_\theta \log\left(\frac{e^{f_\theta^j}}{\sum_k^V e^{f_\theta^k}}\right) \\ &= \nabla_\theta \log\left(e^{f_\theta^j}\right) - \nabla_\theta \log\left(\sum_k^V e^{f_\theta^k}\right)\end{aligned} \tag{21}$$

The first term is just $\nabla_\theta e^{f_\theta^j}$; for the second, apply the chain rule:

$$\begin{aligned}\nabla_\theta \log\left(\sum_k^V e^{f_\theta^k}\right) &= \frac{1}{\sum_b^V e^{f_\theta^b}} \nabla_\theta \left(\sum_k^V e^{f_\theta^k}\right) \\ &= \sum_k^V \frac{e^{f_\theta^k}}{\sum_b^V e^{f_\theta^b}} \nabla_\theta f_\theta^k \\ &= \sum_k^V \pi_\theta^k(t') \nabla_\theta f_\theta^k\end{aligned} \tag{22}$$

Where $\pi_\theta^k(t')$ represents the probability of the possible token k given the context t', we can combine these as follows:

$$\begin{aligned}\nabla_\theta \log(\pi_\theta(t')) &= \nabla_\theta f_\theta^j - \sum_k^V \pi_\theta(k) \nabla_\theta f_\theta^k \\ &= \nabla_\theta f_\theta^k (1 - \pi_\theta(t')) - \sum_{k \neq t}^V \pi_\theta(k') \nabla_\theta f_\theta^k\end{aligned} \tag{23}$$

Substituting this into the GRPO gradient yields:

$$\begin{aligned}&\nabla_\theta \mathcal{J}_{\text{GRPO}}(\theta) = \\ &\mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^G \frac{\hat{\mathcal{A}}_i}{|o_i|}\sum_{t=1}^{|o_i|} r_{i,t}\left(\nabla_\theta f_\theta^k (1 - \pi_\theta(t')) \right.\right. \\ &\left.\left. - \sum_{k \neq t}^V \pi_\theta(k') \nabla_\theta f_\theta^k\right) - \beta \cdot \nabla_\theta \left[\mathrm{D}_{\mathrm{KL}}\left(\pi_\theta \| \pi_{\text{ref}}\right)\right]\right]\end{aligned} \tag{24}$$

9.2 Group-relative Trajectory-aware Policy Optimization: Objective

This section formalizes the *Conflict-Aware Gradient Correction* for GTPO, deriving the token-level loss and gradient. GTPO refines GRPO by weighting each token with a coefficient $\lambda_{i,t}$, determined by conflict regions and the advantage A_i. Furthermore, to mitigate instability, we incorporate an entropy-based filter δ_i. The resulting objective function is defined as:

$$\begin{aligned}\mathcal{J}^* &:= \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^G \frac{1}{|o_i|}\sum_{t \in o_i} r_{i,t}\lambda_{i,t}A_i\right] \\ &= \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^G \frac{1}{|o_i|}\left(\sum_{t \in c_i} r_{i,t}\lambda_{i,t}A_i + \sum_{t \in u_i} r_{i,t}\lambda_{i,t}A_i\right)\right]\end{aligned}$$

Where o_i denotes the set of tokens in each completion, c_i and u_i represent the subsets of conflicting and non-conflicting tokens, respectively. The notation $|\cdot|$ indicates the cardinality (number of elements) of each set and $\lambda_{i,t}$ is a parameter designed to account for conflict and non-conflict formulation. To motivate the definition of $\lambda_{i,t}$, we assume the experimental condition where $r_{i,t} = 1$. Furthermore, to ensure that the model retains the characteristics of GRPO for non-conflicting tokens, we define $\lambda_{i,t}$ as follows:

$$\lambda_{i,t} = \begin{cases}\lambda_i, & \text{if t} \in c_i \\ 1, & \text{if t} \in u_i\end{cases}$$

$$\begin{aligned}\mathcal{J}^* &= \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^G \frac{1}{|o_i|}\left(|c_i|\lambda_i A_i + (|o_i| - |c_i|)\, A_i\right)\right] \\ &= \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\left(\sum_{i=1}^G \frac{|c_i|}{|o_i|}\lambda_i A_i + \sum_{i=1}^G A_i - \sum_{i=1}^G \frac{|c_i|}{|o_i|}A_i\right)\right] \\ &= \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\left(\sum_{i=1}^G \frac{|c_i|}{|o_i|}A_i(\lambda_i - 1) + \sum_{i=1}^G A_i\right)\right]\end{aligned}$$

First of all, given the *GRPO's zero-mean constraint*, $\sum_i^G A_i = 0$, we split the sum into two parts: one considering only the positive completions (G^+), and the other considering only the negative ones (G^-).

$$\mathcal{J}^* = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\left(\sum_{i=1}^{G^+}\frac{|c_i|}{|o_i|}A_i(\lambda_i - 1) + \sum_{i=1}^{G^-}\frac{|c_i|}{|o_i|}A_i(\lambda_i - 1)\right)\right]$$

To prevent inadvertently penalizing tokens that contribute positively in other contexts, we refrain from assigning negative rewards to tokens involved in conflicting cases. These tokens frequently appear in completions that receive positive ratings and are not intrinsically responsible for the negative assessments.

$$\lambda_i = \begin{cases} \lambda, & \text{if i} \in G^+ \\ 0 & \text{if i} \in G^- \end{cases}$$

$$\mathcal{J}^* = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\left(\sum_{i=1}^{G^+}\frac{|c_i|}{|o_i|}|A_i|(\lambda - 1) + \sum_{i=1}^{G^-}\frac{|c_i|}{|o_i|}|A_i|\right)\right]$$

It is evident that by setting $\lambda = 2$, we ensure equal contribution from each completion, regardless of whether it is positive or negative. The resulting equation for the aggregated loss is:

$$\mathcal{J}^* = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{|c_i|}{|o_i|}|A_i|\right] \tag{25}$$

Instead the resulting equation for the token level loss, considering $r_{i,t} \neq 1$, is:

$$\mathcal{J}^* = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=0}^{o_i} r_{i,t}\lambda_{i,t}A_i\right] \tag{26}$$

where:

$$\lambda_{i,t} = \begin{cases} 1, & \text{if } i \in \{G^-, G^+\} \text{ and } t \in u_i \\ 0, & \text{if } i \in G^- \text{ and } t \in c_i \\ 2, & \text{if } i \in G^+ \text{ and } t \in c_i \end{cases} \tag{27}$$

GTPO Gradient: To calculate the gradient of our proposed loss, we start from equation 26, then we apply the same tricks used to compute the GRPO loss (eq. 17).

$$\nabla_\theta \mathcal{J}^* = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G^+}\frac{\tilde{A}_i}{|o_i|}\left(2\sum_{t\in c_i} g_{i,t} + \sum_{t\in u_i} g_{i,t}\right) + \frac{1}{G}\sum_{i=1}^{G^-}\frac{\tilde{A}_i}{|o_i|}\sum_{t\in u_i} g_{i,t}\right] \tag{28}$$

where: $g_{i,t} = r_{i,t}\nabla_\theta\left[\log\left(\pi_\theta(o_{i,t} \mid q, o_{i,<t})\right)\right]$

In addition, following our objective, the adjusted advantage for trajectory i becomes:

$$\tilde{A}_i = A_i - \gamma\langle H\rangle_i, \tag{29}$$

where $\langle H\rangle_i$ represents the average policy entropy over all tokens of trajectory i, and γ controls the strength of the entropy correction.

Reward Settings. The reward used in this paper is composed of two components: the first one refers to the format of the model's answer, and the second one refers to the accuracy of the model's answer. The first component assesses formatting: the model receives a score of 10 if it includes all predefined special formatting tokens specified in the prompt (see box 9.2). If only a subset of the reasoning or answer tokens are present, the model is awarded 1 point; otherwise, it scores 0. The second component evaluates accuracy: the model obtains a score of 10 if the correct target answer is correctly enclosed within the answer tokens; otherwise, it receives 0. The final reward is the sum of these two components and is provided to guide the learning algorithms.

9.2 PROMPT

You are a helpful assistant for solving math problems.
Given a question, first think step by step between `<reasoning>` and `</reasoning>`.
Then, give the final answer between `<answer>` and `</answer>`.

Training Settings. Listing 1 reports the main training parameters used for GRPO, GTPO and SFT, for both LLaMA and Qwen models. To implement GTPO we used Unsloth (Daniel Han and team, 2023).

```
model_name: "meta-llama/meta-Llama-3.1-8
    B-Instruct" / "Qwen/Qwen2.5-3B-
    Instruct"
max_seq_length: 5500
max_prompt_length: 4000
lora_rank: 128 / 64 (Qwen)
load_in_4bit: true
gpu_memory_utilization: 0.4
target_modules:
  - "q_proj"
  - "k_proj"
  - "v_proj"
  - "o_proj"
  - "gate_proj"
```

```
  - "up_proj"
  - "down_proj"
random_seed: 3407
warmup_ratio: 0.005
learning_rate: 1e-6
adam_beta1: 0.999999 / 0.9 (GRPO, DAPO &
     SFT)
adam_beta2: 0.999999 / 0.95 (GRPO, DAPO
    & SFT)
weight_decay: 0.1
beta: 0.0/ 0.04/ 10e-6 (GRPO) / "None" (
    GTPO, DAPO & SFT)
lr_scheduler_type: "cosine"
optimizer: "paged_adamw_8bit"
logging_steps: 1
per_device_train_batch_size: 1
gradient_accumulation_steps: 1
num_generations: "8/12" / "None" (SFT) /
     "8" (R1 Dataset)
num_train_epochs: 1
num_iterations: 1
save_steps: 500
max_grad_norm: 0.1
report_to: ["wandb"]
```

Listing 1: Settings for GTPO, GRPO and SFT.

Algorithm 1: Forward Pass

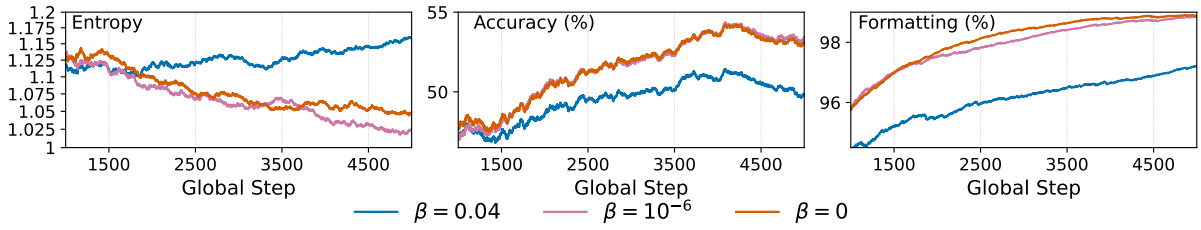
Input: $\mathbf{X} \in \mathbb{R}^{G \times L}$ (Token IDs), $\mathbf{A} \in \mathbb{R}^{G}$ (Advantages)
Output: $\mathbf{M}_{fwd} \in \{0,1\}^{G \times L}$ (Forward Conflict Mask)

1: Extract active sequences $\mathbf{X}_{act}$ where $\mathbf{A}_i \neq 0$ *% Shape:* $[G_{act}, L]$
2: Extract positive sequences $\mathbf{X}_{pos}$ where $\mathbf{A}_i > 0$ *% Shape:* $[G_{pos}, L]$
3: Extract negative sequences $\mathbf{X}_{neg}$ where $\mathbf{A}_i < 0$ *% Shape:* $[G_{neg}, L]$
4: **Broadcast Equality:**
5: $\mathbf{E}_{pos} \leftarrow \mathbf{X}_{act}.\text{unsq}(1) == \mathbf{X}_{pos}.\text{unsq}(0)$ *% Shape:* $[G_{act}, G_{pos}, L]$
6: $\mathbf{E}_{neg} \leftarrow \mathbf{X}_{act}.\text{unsq}(1) == \mathbf{X}_{neg}.\text{unsq}(0)$ *% Shape:* $[G_{act}, G_{neg}, L]$
7: **Strict Trajectory (Cumulative Product):**
8: $\mathbf{S}_{pos} \leftarrow \text{cumprod}(\mathbf{E}_{pos}, \text{dim} = 2)$
9: $\mathbf{S}_{neg} \leftarrow \text{cumprod}(\mathbf{E}_{neg}, \text{dim} = 2)$
10: **Reduce Trajectories:**
11: $\mathbf{I}_{pos} \leftarrow \text{any}(\mathbf{S}_{pos}, \text{dim} = 1)$ *% Shape:* $[G_{act}, L]$
12: $\mathbf{I}_{neg} \leftarrow \text{any}(\mathbf{S}_{neg}, \text{dim} = 1)$ *% Shape:* $[G_{act}, L]$
13: **Final Forward Mask:**
14: $\mathbf{M}_{fwd} \leftarrow \mathbf{I}_{pos} \wedge \mathbf{I}_{neg}$

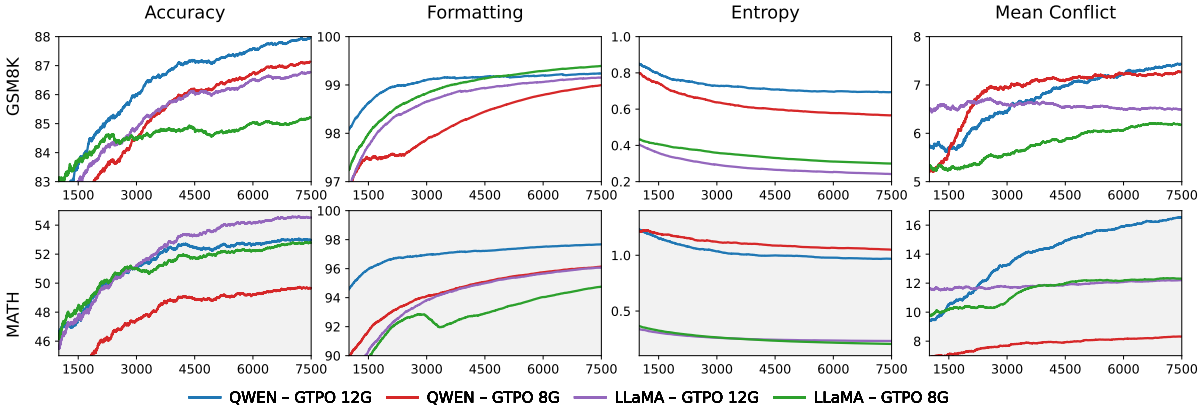
9.3 Complexity of Conflict Masking

In this section, we detail the computational complexity of the dual-pass conflict masking algorithm. To ensure minimal computational overhead during training, we implement a highly optimized, vectorized approach based on tensor broadcasting. Algorithm 1 outlines the core operations of our strict trajectory matching for the forward pass. For brevity, the backward pass is omitted from the pseudocode, as it operates under the exact same logic by simply flipping the token sequences along the sequence length dimension prior to matching. The computational cost is dominated by the tensor broadcasting operations (Lines 5-6) and the cumulative products (Lines 8-9). The algorithm first filters the input batch to isolate G_{act} valid sequences, where $G_{act} \leq G$. It then expands dimensions to perform parallel equality checks between the G_{act} active sequences and the G_{pos} (or G_{neg}) target sequences. This broadcasting creates a temporary boolean volume of shape $[G_{act}, G_{pos}, L]$. Consequently, both the spatial and temporal complexity for a single pass are strictly bounded by $\mathcal{O}(G_{act} \cdot G_{pos} \cdot L)$. Because $G_{act}, G_{pos}, G_{neg} \leq G$, the worst-case upper bound for the forward pass is $\mathcal{O}(G^2 L)$. Since the dual-pass algorithm simply executes this logic twice (once forward, and once backward on reversed tensors), the total complexity is bounded by $\mathcal{O}(2 \cdot G^2 L)$, which asymptotically simplifies back to $\mathcal{O}(G^2 L)$.

Given that Reinforcement Learning fine-tuning typically employs small generation group sizes ($G \in [4, 64]$), the $\mathcal{O}(G^2 L)$ overhead is negligible compared to the standard forward and backward passes of the language model itself. By confining the comparison space strictly within the intra-batch generated trajectories, the algorithm ensures stable and efficient training dynamics with minimal memory footprint.



(a)



(b)

Figure 11: Ablation of GTPO for QWEN and LLaMA on MATH and GSM8K.(a) Average entropy (left), accuracy (center), and formatting rate (right) in QWEN trained with GRPO on MATH with different values of β of KL; (b) Impact of GTPO group size ($G = 8$ vs. $G = 12$) on GSM8K (top) and MATH (bottom) for QWEN and LLaMA. Metrics over training steps: accuracy, formatting reward, entropy, and mean conflict (left to right).

10 Complementary results

10.1 Impact of KL β-coefficient on GRPO.

Figure 11a presents an ablation study on the effect of the KL divergence coefficient β during GRPO training on the MATH dataset using the Qwen model. The three plots respectively show: (left) the average entropy, (center) the accuracy rate, and (right) the formatting rate across training steps.

In the leftmost plot, we observe that with $\beta = 0.04$ (blue line), the entropy steadily increases, indicating that the model becomes progressively less confident in its predictions. Conversely, smaller or null values of β (orange for $\beta = 0$ and pink for $\beta = 10^{-6}$) result in a more stable or decreasing entropy, reflecting more deterministic behavior during generation.

The central plot highlights how lower values of β lead to better accuracy. In particular, $\beta = 0$ achieves the highest accuracy throughout training, while $\beta = 0.04$ results in slower improvement and lower final performance. This suggests that a strong KL regularization term may overly constrain the policy and hinder learning.

Finally, the rightmost plot shows the formatting reward. Both $\beta = 0$ and $\beta = 10^{-6}$ achieve high formatting scores (above 98%) by the end of training, whereas $\beta = 0.04$ lags behind. This confirms that a weaker KL constraint facilitates the preservation of structural consistency and formatting in model completions. Overall, the figure demonstrates that reducing or removing the KL divergence term in GRPO ($\beta \to 0$) leads to better performance in terms of both accuracy and formatting, while avoiding the entropy inflation observed with larger β values.

10.2 Sensitivity of GRPO to Adam Momentum.

Figure 12 compares the impact of different Adam optimizer hyperparameter configurations, specifically the momentum coefficients α_1 and α_2, on the training dynamics of GRPO with generation size $G = 8$. The two plots report the evolution of accuracy (left) and formatting score (right) over the global training steps. The experiments are conducted using two models, Qwen and LLaMA, each evaluated under two Adam settings:$\alpha_1 = 0.999999$, $\alpha_2 = 0.999999$ (used in GTPO (Dohare et al., 2023));$\alpha_1 = 0.9$, $\alpha_2 = 0.95$ (the

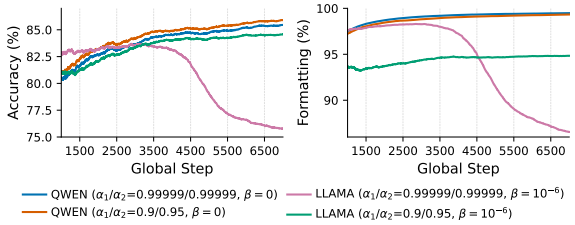


Figure 12: GRPO-G=8, comparison between different adam α_1 and α_2 values.

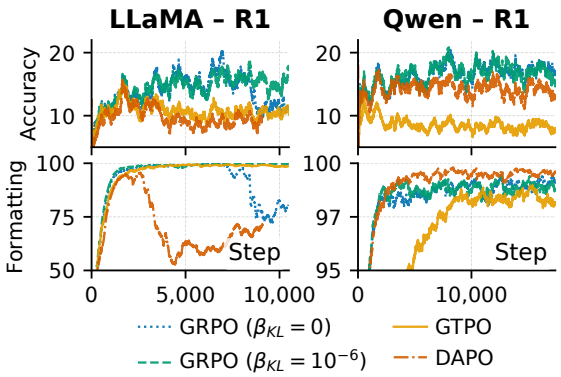


Figure 13: *Accuracy and formatting rewards* (%) of GTPO, GRPO and DAPO over training steps on R1.

one used by original GRPO (Shao et al., 2024)). For Qwen (blue and orange curves), the optimizer settings do not significantly impact training stability: both configurations lead to consistent improvements in accuracy and formatting over time, with the GRPO originals ($\alpha_1 = 0.9$, $\alpha_2 = 0.95$) slightly outperforming in final accuracy. In contrast, for LLaMA (pink and green curves), the choice of α_1/α_2 has a substantial effect. Using the GTPO ones ($\alpha_1 = 0.99999$, $\alpha_2 = 0.999999$) causes a sharp degradation in both accuracy and formatting after approximately 3500 steps, symptomatic of policy collapse. On the other hand, the GRPO default configuration results in more stable formatting behavior and mitigates collapse, although final formatting remains below 95%. These results suggest that GRPO training with LLaMA is highly sensitive to Adam hyperparameters, and that careful tuning of α_1 and α_2 is crucial to maintain training stability and prevent collapse, particularly when using smaller values of β.

10.3 Effect of GTPO Group Size on Training Dynamics

Figure 11b compares GTPO with two group sizes (G=8 vs. G=12) on GSM8K and MATH for QWEN and LLaMA, reporting *accuracy*, *formatting*, *entropy*, and *mean conflict*. On GSM8K, larger groups help both models: QWEN 12G achieves the best final accuracy, followed by LLaMA 12G, while both 8G settings converge more slowly and to lower plateaus. All runs exceed 98% formatting, but larger groups stabilize formatting earlier, with QWEN 12G reaching the highest levels. LLaMA maintains lower entropy than QWEN; increasing group size slightly raises entropy for LLaMA but not for QWEN. Mean token-level conflict increases with group size (12G $>$ 8G), and QWEN shows higher conflict than LLaMA, consistent with its higher entropy and trajectory diversity; under GTPO, this diversity remains beneficial and does not harm formatting. On MATH, the advantage of larger groups persists. LLaMA 12G achieves the highest accuracy, with LLaMA 8G close behind; QWEN benefits from 12G but remains below LLaMA. QWEN attains the highest formatting scores, especially with 12G, reaching near-perfect formatting early in training, while LLaMA also improves with 12G but requires more steps to reach comparable levels. As on GSM8K, LLaMA stays lower-entropy than QWEN, and mean conflict grows with group size, most noticeably for QWEN 12G, whereas LLaMA shows more moderate conflict and weaker sensitivity to group size.

10.4 R1 training

Figure 13 illustrates the training trajectories on the R1 dataset. To conserve resources, LLaMA training was halted at around 10,000 steps, as QWEN's convergence behavior suggested further improvements were unlikely. All techniques exhibit a shared limitation: the accuracy reward plateaus near 20%. We attribute this to sparse learning signals; generations often format correctly but fail the actual task (correctness =0), yielding zero gradients that stall learning. Consequently, while GTPO does not attain the highest peak reward, it excels in late-stage stability. Unlike GRPO and DAPO, which eventually suffer from reward collapse on LLaMA, GTPO maintains a stable trajectory. This robustness translates into competitive generalization. As Table 1 shows, GTPO's lower peak reward does not degrade out-of-distribution (OOD) performance. For QWEN, shared gradient stagnation homogenizes OOD results across all methods. For LLaMA, GTPO matches or exceeds baselines

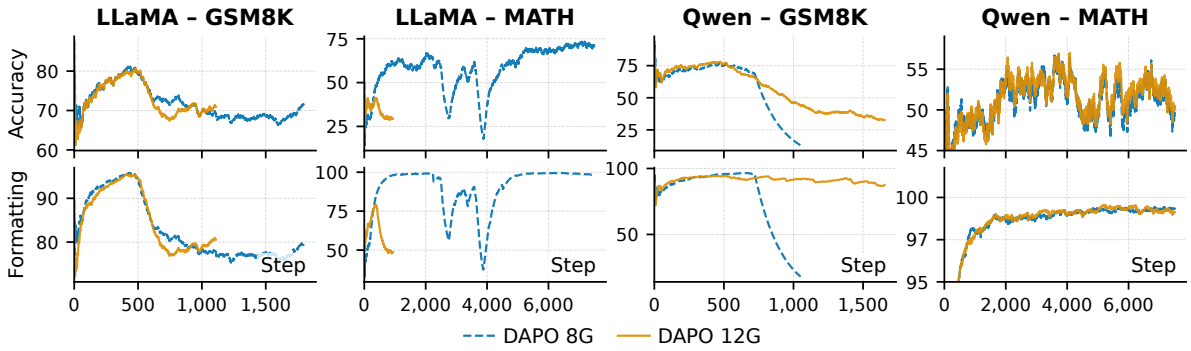


Figure 14: *Accuracy and formatting rewards* (%) of DAPO over training steps on GSM8K and MATH.

in pass@k and maj@k for $k = 64$, despite slight underperformance at lower k values.

10.5 DAPO training

Figure 14 illustrates DAPO's training dynamics on GSM8K and MATH for LLaMA and QWEN. While DAPO largely mirrors GRPO's vulnerabilities, its lack of explicit regularization on o_i introduces unique instabilities. Crucially, this omission causes DAPO to suffer reward collapse on QWEN, a degradation not seen in GRPO. Dataset trajectories further underscore this volatility. On GSM8K, rapid collapse forced early termination for both models. Conversely, LLaMA on MATH (G=12) was trained to completion to test recovery; although it eventually rebounded from an initial collapse, such drastic fluctuations preclude reliable training. Only QWEN on MATH maintained a stable, collapse-free trajectory. Consequently, for out-of-distribution (OOD) evaluations, we selected the last checkpoint prior to the peak reward.

10.6 MMLU further analysis

Figure 15 details the average MMLU accuracy across the LLaMA and Qwen models, aggregated by training technique (SFT, DAPO, GRPO, GRPO with $\beta = 0.04$, and GTPO). The heatmap visualizes performance on individual subjects categorized into four macro-areas: STEM, Humanities, Social Sciences, and Others. Crucially, the results demonstrate that GRPO style methods yield comparable performance to one another, consistently outperforming the SFT baseline across domains.

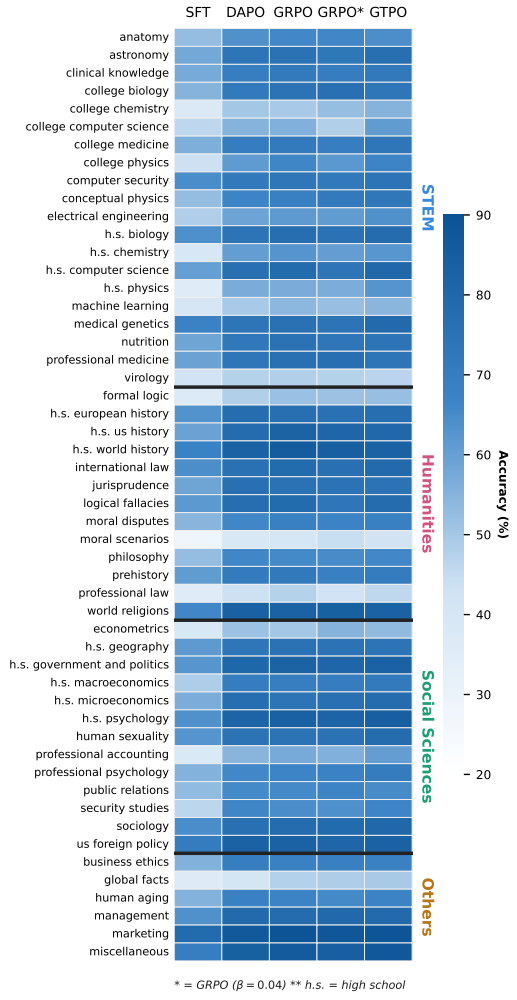


Figure 15: Heatmap of the average accuracy on the MMLU benchmark across LLaMA and Qwen, aggregated by training technique.