ATPO: Adaptive Token Aggregation for Policy Optimization

Anonymous ACL submission

Abstract

Group Relative Policy Optimization (GRPO) has become a central paradigm for training large language models to perform long-horizon reasoning. Existing GRPO-style methods implicitly impose fixed token-level aggregation rules. In this work, we propose Adaptive Token Aggregation for Policy Optimization (ATPO), a unified framework that reframes GRPO, DAPO, and related variants as instances of a common token preference optimization objective. Instead of relying on manually designed heuristics, ATPO introduces a learnable token aggregation mechanism that allows the model to adaptively determine how responses of different lengths should contribute to policy updates. A single interpretable parameter explicitly controls the direction of token preference, enabling a smooth trade-off between favoring longer or shorter responses. Extensive experiments across multiple model families and scales (Qwen2.5, Llama 3, and Mathstral) demonstrate that ATPO consistently outperforms GRPO on various mathematical reasoning benchmarks as well as out-of-domain evaluation on GPQA without additional supervision or computational overhead.

1 Introduction

Large language models (LLMs) with o1-like systems (OpenAI et al., 2024; Wu et al., 2024; Guo et al., 2025) have been widely applied to complex reasoning tasks such as mathematical problem solving. A common approach to training long chain-of-thought reasoning is Reinforcement Learning with Human Feedback (RLHF) (Xu et al., 2025), where models explore trajectories and are optimized toward preferred outcomes. The most popular optimization method in this paradigm is Proximal Policy Optimization (PPO) (Schulman et al., 2017). However, PPO training is often unstable and computationally expensive (Zheng et al., 2023; Wu et al., 2023). To address these limitations, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) removes the need for separate value model and reward model and rely on a rule-based verifier as the training signal. This design has proven highly effective in domains requiring verifiable reasoning, spanning mathematical problem solving (Guo et al., 2025), code programming (Liu and Zhang, 2025), search agent (Jin et al., 2025) and even multimodal reasoning (Huang et al., 2025a).

Despite these advances, modern policy and preference optimization methods for post-training almost always exhibit a strong *length bias*, the tendency for models to generate unnecessarily long responses even when concise answers would suffice. For example, Singhal et al. (2024a) showed that much of the reward gains in RLHF stem from response length increases rather than substantive improvements in quality. Similarly, Hu et al. (2025a) found that reward models and preference datasets themselves often favor longer responses, which leads trained policies to exploit this bias by producing verbose outputs. This issue also persists in GRPO, since the advantage is applied uniformly across all tokens of a response (Shao et al., 2024). Recent variants address this bias by modifying how token rewards are aggregated. DAPO (Yu et al., 2025) enforces uniform token-level aggregation so that all tokens across responses share equal weight, while Liu et al. (2025a) aggregates token-level gradients without variance normalization. Although these strategies partially mitigate length bias, they remain heuristic and lack the flexibility to adapt across tasks. These observations raise a critical question: ***Can we let the model itself decide its token preference?*** In other words, rather than hard-coding whether long or short replies should be favored, can the optimization process learn a tunable, context-aware preference over response length?

To this end, we propose a unified framework ***A**daptive **T**oken Aggregation for **P**olicy **O**ptimization (ATPO)* with a learnable weighting

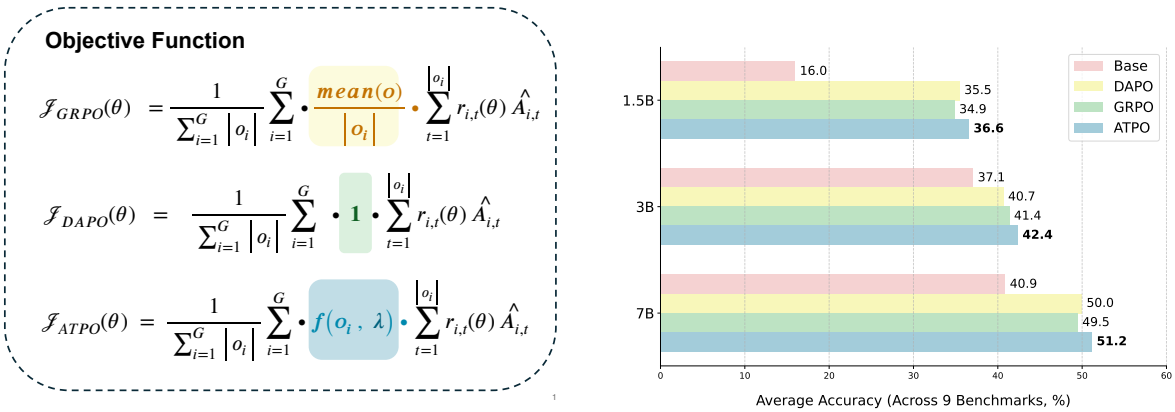


Figure 1: The key distinction among GRPO, DAPO, and ATPO lies in their token-aggregation schemes (highlighted in the shaded region). Across *Qwen2.5* models of multiple sizes (1.5B, 3B, and 7B) and average of 8 math benchmarks and 1 OOD benchmark, ATPO consistently outperforms others.

scheme. Unlike heuristics based methods (Shao et al., 2024; Yu et al., 2025; Liu et al., 2025a), our method introduces a parametric weighting function with a tunable parameter λ that adapts dynamically to the distribution of response lengths. This design enables the model to determine its own token aggregation in a context-aware manner. We summarize the main difference in Figure 1, and highlight how existing methods arise as special cases of our formulation. Empirically, we demonstrate that our method consistently achieves stronger performance across models of different scales (e.g., 1.5B, 3B, and 7B) and families (Llama 3 and Qwen) on eight in-domain mathematical reasoning benchmarks, as well as out-of-domain (OOD) tasks such as GPQA (Rein et al., 2024). Our main contributions are summarized as follows:

- We propose a unified formulation that encompasses GRPO, DAPO, and Dr. GRPO as special instances of token-level preference optimization. This approach clarifies the implicit assumptions each method makes regarding length aggregation.
- We introduce a learnable preference framework that adaptively reweights token contributions by response length distributions, enabling the model to learn its own preference.
- We instantiate ATPO within group-based policy optimization via a single learnable parameter λ, and empirically demonstrate consistent improvements in reasoning performance, reduced verbosity, and enhanced training stability across multiple model scales, architectures.

2 Related Work

Reinforcement Learning with Verifiable Rewards (RLVR) Reasoning-centric post-training for LLMs increasingly relies on *verifiable* signals (e.g., math/code correctness) as scalable supervision, reducing dependence on expensive human preference labels in RLHF (Ouyang et al., 2022; Shao et al., 2024; Le et al., 2022; Liu et al., 2023). Under this RLVR paradigm, policy optimization has been shown to substantially improve reasoning on both mathematical and programming benchmarks (Feng et al., 2025b; Liu et al., 2025b). A representative approach is *Group Relative Policy Optimization* (GRPO), which replaces critic-based token-level advantage estimation in PPO (Schulman et al., 2017) with group-wise normalization over multiple sampled responses per query, enabling critic-free updates and improved sample efficiency (Shao et al., 2024; Guo et al., 2025; Mroueh et al., 2025).

Building on GRPO, subsequent work can be broadly grouped into: *objective/ratio design* variants that modify importance-ratio aggregation or clipping (Zhao et al., 2025b; Zheng et al., 2025); *reward shaping and regularization* mechanisms that augment outcome rewards with additional structured signals (e.g., consistency) or constrain updates to mitigate degeneration (Chen et al., 2025; Jang et al., 2024); and *credit assignment / variance reduction* techniques for long-horizon or nonstationary rewards, including hierarchical grouping and adaptive baselines (Feng et al., 2025a; Wang et al., 2025). Together, these extensions refine stability, variance, and scalability of RLVR-style train-

ing while retaining the core group-based, critic-free spirit of GRPO.

Token-weighting and Normalization Biases While GRPO removes the critic, it can introduce biases in how token-level losses are aggregated: if token advantages are treated uniformly, longer responses may accumulate larger total gradients regardless of quality; conversely, aggressive length-based normalization can over-reward brevity when correct yet under-penalize verbose errors. Length-related preference is not unique to RLVR. In PPO-style RLHF, systematic verbosity/length bias has been repeatedly observed in preference data and reward modeling (Singhal et al., 2024b; Zhao et al., 2025a). Several works further show that these artifacts can persist through post-training and motivate post-hoc diagnosis or calibration of reward signals (Huang et al., 2025b; Bu et al., 2025). Complementarily, disentangling reward factors has been proposed to reduce reward hacking behaviors that can be entangled with length (Chen et al., 2024). Similar issues also arise in direct preference optimization (DPO). DPO can be viewed as optimizing an implicit reward under a KL-structured objective (Rafailov et al., 2023), and follow-up analyses show that improvements can correlate with response length unless length is explicitly controlled or desensitized (Park et al., 2024; Liu et al., 2024). Related variants adjust the objective/regularization to curb biased length reliance while maintaining the simplicity of direct optimization (Lu et al., 2024; Meng et al., 2024).

These observations motivate methods that explicitly reweight or normalize token/sequence contributions. *Dynamic sAmpling Policy Optimization* (DAPO) (Yu et al., 2025) introduces length-aware weighting to balance short and long responses during updates. *Dr. GRPO* (Liu et al., 2025a) improves robustness to reward noise by reducing the impact of outlier responses. Concurrently, Token Hidden Reward (Deng et al., 2025) reweights per-token updates in GRPO via a "hidden reward," exposing a knob p that steers exploitation versus exploration.

3 Methodology

3.1 Preliminary

Reinforcement Learning with Verifiable Rewards (RLVR) views text generation with an LLM as a sequential decision process. At step t, the state is the prompt x together with the partial output $o_{1:t-1}$, and the action is to produce the next token $o_t \sim \pi_\theta(\cdot \mid x, o_{1:t-1})$. The policy π_θ denotes the current model we are optimizing, while $\pi_{\theta_{\text{old}}}$ is the frozen policy from the previous iteration that is used to sample trajectories for stable training. The reward function $r(x, o)$ is defined by deterministic rules (e.g., correctness of an answer or validity of code), ensuring that the feedback signal is verifiable rather than subjective. The learning objective combines two ingredients: a policy gradient loss that increases the likelihood of outputs with higher advantage values, and a KL penalty that keeps the updated policy not far from a fixed reference distribution $\pi_{\theta_{\text{ref}}}$. The resulting objective is:

$$\mathcal{J}_{RLVR}(\theta) = E_{x\sim\mathcal{D},,o\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[r_{i,t}(\theta)A(x,o)\right] + \beta\mathbb{E}_{x\sim\mathcal{D}}\left[\mathrm{KL}\big(\pi_{\theta_{\text{ref}}}(\cdot|x)||\pi_\theta(\cdot|x)\big)\right] \tag{1}$$

where $A(x, o)$ is the advantage function that measures how much better a completion o performs. The KL term is optional, with a coefficient β that controls the trade-off between maximizing the verifiable reward and staying close to the reference model. r is the importance sampling ratio comparing the new and old policy:

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid x, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid x, o_{i,<t})} \tag{2}$$

GRPO was proposed as a lightweight alternative to PPO that avoids learning an additional value function while still providing a stable training signal (Shao et al., 2024). The key idea is to measure the quality of each sampled output relative to a group of alternatives drawn from the same policy. Rather than fitting a value baseline, GRPO defines the advantage of o_i by subtracting the average reward of the group and scaling by its standard deviation:

$$\hat{A}_{i,t} = \frac{R(x, o_i) - \text{mean}(\{R(x, o_j)\}_{j=1}^G)}{\text{std}(\{R(x, o_j)\}_{j=1}^G)} \tag{3}$$

This group-normalized advantage is then assigned uniformly to all tokens in the response, leading to the clipped surrogate loss

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big\{r_{i,t}(\theta)\,\hat{A}_{i,t},\ \text{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\} \tag{4}$$

where G denotes the number of rollouts (i.e., sampled responses per prompt), $|o_i|$ is the length of the i-th response.

DAPO A central feature of DAPO is that the policy gradient is applied directly at the token level rather than treating the entire response as a single unit. This design ensures that every token within a sampled output o_i contributes proportionally to the gradient update. This fine-grained token-level optimization improves stability and provides more informative feedback for training LLMs. Formally, the DAPO objective is defined as

$$\mathcal{J}_{\text{DAPO}}(\theta) = \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\Big\{r_{i,t}(\theta)\,\hat{A}_{i,t},\ \text{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon'\big)\,\hat{A}_{i,t}\Big\} \tag{5}$$

Dr. GRPO (Liu et al., 2025a) was proposed that each response is normalized by the group's standard deviation, and the policy loss is further averaged across response length:

$$\mathcal{J}_{\text{Dr.GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\Big\{r_{i,t}(\theta)\,\hat{A}_{i,t},\ \text{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\} \tag{6}$$

3.2 Unified Token Preference

The policy optimization objectives of GRPO, DAPO, and Dr. GRPO can be expressed under a unified framework that we call *Unified Token Preference*. The key observation is that all three methods share the same token-level clipped surrogate structure, but differ in the weighting function $f(o_i)$ applied to each sampled output o_i (details shown in Appendix D). Formally, the unified objective can be written as

$$\mathcal{J}(\theta) = \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G} f(o_i)\sum_{t=1}^{|o_i|}\min\Big\{r_{i,t}(\theta)\,\hat{A}_{i,t},\ \text{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\} \tag{7}$$

The weighting function $f(o_i)$ specifies how each output contributes to the token-level aggregation:

$$f_{\text{GRPO}}(o_i) = \frac{\mu}{|o_i|} \tag{8}$$

$$f_{\text{DAPO}}(o_i) = 1 \tag{9}$$

$$f_{\text{Dr.GRPO}}(o_i) = \mu \tag{10}$$

where μ denotes the average response length across the group, i.e., $\mu = \frac{1}{G}\sum_{i=1}^{G}|o_i|$.

This formulation highlights that GRPO effectively downweights long responses, DAPO treats all responses equally at the token level, and Dr. GRPO applies a uniform length-based scaling across responses. By casting them into the same token preference framework, we can analyze their differences in terms of how they balance output length and reward signal during optimization.

3.3 Learnable Preference

Previous designs do not allow the optimization process to flexibly balance between long and short responses across different prompts. To overcome this limitation, we propose a *learnable preference* that treats response length as a stochastic variable and derives adaptive weights through a normalized transformation. Given G responses $\{o_i\}_{i=1}^{G}$, we first compute their mean and variance

$$\mu = \text{mean}(o) \tag{11}$$

$$\sigma = \text{std}(o) \tag{12}$$

And then standardize the length:

$$z_i = \frac{o_i - \mu}{\sigma} \tag{13}$$

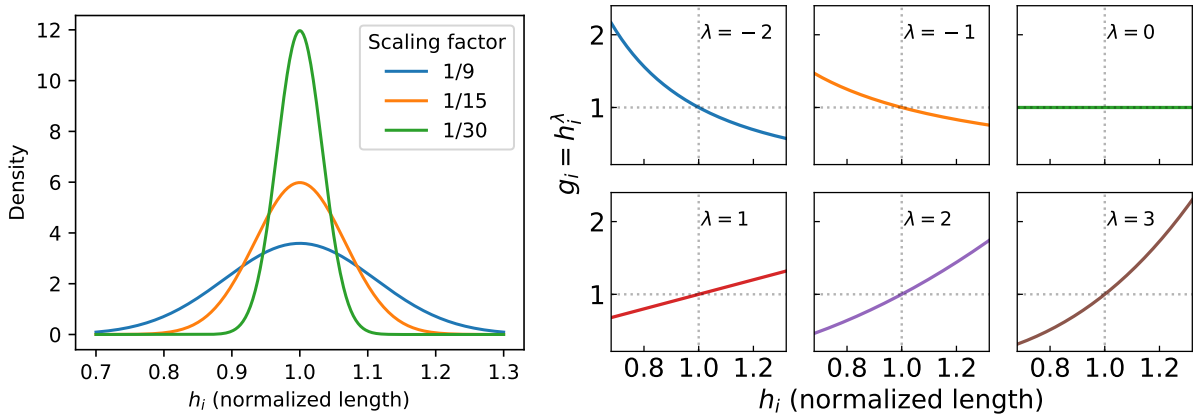
$$h_i = 1 + r\,z_i \tag{14}$$

$r > 0$ is a scaling factor. This construction ensures that h_i is centered around 1, with deviations controlled by r. Illustratively, about 99.7% of $z_i = \frac{o_i-\mu}{\sigma}$ lie within $[-3, 3]$, implies that $h_i \in [1-3r,\ 1+3r]$ for almost all responses. As shown in Figure 2a, this corresponds to $h_i \in [0.9, 1.1]$ for $r = \frac{1}{30}$ (conservative), $h_i \in [0.8, 1.2]$ for $r = \frac{1}{15}$ (moderate), and $h_i \in [\frac{2}{3}, \frac{4}{3}]$ for $r = \frac{1}{9}$ (aggressive).

We then apply an exponent λ to adjust the relative contribution:

$$g_i = h_i^{\lambda} \tag{15}$$

On the role of λ The exponent λ in $g_i = h_i^{\lambda}$ can be interpreted as a **tunable** token preference. Figure 2b illustrates this effect. When $\lambda = 0$, all $g_i = 1$, meaning that every response is treated equally regardless of its normalized length h_i. When $\lambda > 0$, g_i grows with h_i, which emphasizes longer responses and downweights shorter ones. Conversely, when $\lambda < 0$, g_i decreases with h_i, assigning more weight to shorter responses while suppressing long ones. Overall, λ sets the *direction* of token preference, and scaling factor r sets its *strength*, offering an interpretable knob to trade off



(a) Scaling effect

(b) Lambda effect

Figure 2: Illustration of our learnable preference design. (a) shows how the scaling factor $1/k$ influences the distribution of normalized lengths h. Larger scaling values produce a wider spread, magnifying differences between long and short responses, smaller values compress the distribution around 1, reducing sensitivity to length deviations. (b) shows how the exponent λ adjusts the direction of token preference through $g_i = h_i^\lambda$. When $\lambda = 0$, all responses are weighted equally; positive λ favors longer responses, negative λ emphasizes shorter ones.

verbosity vs. brevity without distorting the expectation across methods.

The scale of g can vary significantly across groups and across different λ values. To stabilize training and make weights comparable, we normalize g with a softmax:

$$f(o_i) = G \cdot \frac{\exp(g_i)}{\sum_j \exp(g_j)} \tag{16}$$

The overall objective function is:

$$\begin{aligned}\mathcal{J}(\theta) = &\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\left(G \cdot \frac{\exp\left(h_i^\lambda\right)}{\sum_{j=1}^{G}\exp\left(h_j^\lambda\right)}\right) \\ &\times \sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\,\hat{A}_{i,t}, \\ &\mathrm{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big)\end{aligned} \tag{17}$$

The detailed derivation of the gradient with respect to λ is provided in Appendix E.

3.4 Analysis

Group-relative token preference ATPO models token preference relative to the distribution of response lengths within each rollout group, rather than using absolute response length. This choice avoids prompt-dependent bias, as response length scales can vary substantially across tasks and training stages. By standardizing lengths within each group, ATPO focuses on relative deviations from the group mean, leading to stable and context-adaptive preference learning. This formulation is consistent with prior observations that length bias in RLHF and GRPO-style methods is inherently comparative rather than absolute (Singhal et al., 2024c; Hu et al., 2025b).

Softmax normalization Softmax normalization ensures that token weights depend only on relative differences among responses within a group. Without normalization, exponentiated preferences could produce highly imbalanced weights and destabilize optimization. Softmax prevents such degenerate updates while preserving comparability across groups, in line with normalization principles used in GRPO and related objectives (Shao et al., 2024). Rescaling by G further maintains consistency with prior formulations.

Learnable preference parameter The preference parameter λ is directly optimized and admits a clear semantic interpretation. It controls the direction of token preference, determining whether longer or shorter responses are emphasized during training. Unlike heuristic length penalties or fixed aggregation rules (Yu et al., 2025; Liu et al., 2025b), λ evolves smoothly and converges to stable values across model scales, indicating that preference learning is well-behaved.

Relation to length bias mitigation ATPO addresses verbosity bias at the optimization level by learning how token-level updates are aggregated,

rather than relying on explicit length regularization or post-hoc calibration (Singhal et al., 2024c; Park et al., 2024). This allows ATPO to reduce verbosity while preserving reasoning performance.

4 Experimental Setup

Training Datasets We follow the SimpleRL Zoo setting (Zeng et al., 2025), adopting GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) training datasets. In line with SimpleRL Zoo's setup, we emphasize that the difficulty of training data is crucial for the success of zero RL, where training may start directly from the base models in a way that accords with the base models' natural ability to explore and discover rewardable behaviors. The data is partitioned into three difficulty levels: Easy (GSM8K and MATH level 1), Medium (MATH levels 1–4), and Hard (MATH levels 3–5). We use the hard level with $8,523$ problems as our training set, and *qwen-boxed* as our chat template, with an example in Appendix A.

Evaluation For in-domain evaluation, we follow SimpleRL (Zeng et al., 2025), using 8 widely used math reasoning benchmarks: GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), Gaokao (Zhang et al., 2023), OlympiadBench (He et al., 2024a), and College Math (Innovations, 2024), AIME24 (LI et al., 2024), and AMC23 (He et al., 2024b), relying on the Qwen Math codebase that was also used in Zeng et al. (2025). For AMC and AIME, we report Average@32: for each problem, we generate 32 independent rollouts (stochastic samples) and compute the mean Pass@1 accuracy across these 32 attempts; the final score is the average of these per-item means over the dataset. For the remaining six benchmarks, we report the standard Pass@1 using a single prediction per problem. The statistics of math benchmarks are listed in Appendix B. For OOD evaluation, we test on GPQA (Rein et al., 2024).

Model Selection We evaluate our method on a diverse set of LLMs such as *Qwen2.5* (Qwen et al., 2025), *Llama-3* (Grattafiori et al., 2024), and *Mathstral* (Jiang et al., 2023).

Detail Implementation Our training is based on the VeRL (Sheng et al., 2025) and use a rule-based reward, adding a format penalty on the basic correctness (ground truth) that assigns $+1$ for correct answers, -0.5 for incorrect answers with correct boxed formats, and -1 for incorrect answers with incorrect formats. We fine-tune the base models using 4× NVIDIA A800 GPUs for the 1.5B/3B models (19 hours) and 8× NVIDIA A800 GPUs for the 7B/8B model (43 hours), with a batch size of 1024, mini batch size of 256, micro batch size of 8 and a max response length of 2048. For λ updating, we use stochastic gradient descent with a relatively large learning rate $1e-1$ and no weight decay.

Baselines We compare our method with vanilla GRPO (Shao et al., 2024), DAPO (Yu et al., 2025) and Dr.GRPO (Liu et al., 2025a).

5 Results

5.1 Main Results

Table 1 reports the performance of various models across multiple tasks.

In-Domain Evaluation As expected, the base models without RL are extremely weak. This highlights the necessity of reinforcement learning in stimulating the reasoning potential of these models. Across model scales, our method consistently outperforms both GRPO and DAPO. On *Qwen2.5-1.5B*, it attains an average score of 36.6, exceeding DAPO (+1.1%) and GRPO (+1.7%). On the *Qwen2.5-3B* base model, our method further improves the average to 42.4, showing consistent gains over both DAPO (+1.7%) and GRPO (+1%). For the larger *Qwen2.5-7B* model, the base model already exhibits stronger reasoning ability; the advantage of our method remains sufficient, with average results of 51.2 compared to 49.5 on GRPO (+1.7%) and 50.0 on DAPO (+1.2%). We observe a similar pattern on *Llama3-8B*: our method attains an average of 23.3, improving over DAPO (21.0, +2.3%) and GRPO (21.3, +2.0%). The improvements are most pronounced on widely used core benchmarks such as GSM8K (70.3 vs. 64.4/64.1) and MATH500 (29.6 vs. 26.4/26.0). We also evaluate the math-specialized *Mathstral* family and observe the same overall trend: our method remains competitive and delivers consistent gains on the averaged in-domain score. Overall, these results confirm that our method consistently improves reasoning performance across scales.

Out-of-Domain Evaluation We further evaluate generalization on the OOD benchmark GPQA; results are reported in Table 1 under the OOD/GPQA column. Overall, our method also generalizes well to this OOD benchmark and improves

Model	Method	Math								OOD	Avg
		GSM8K	MATH500	Minerva	Gaokao	Olympiad	College	AIME (Avg@32)	AMC (Avg@32)	GPQA	
Qwen2.5 1.5B	Base	56.4	29.2	7.8	16.9	7.4	11.6	0.4	15.0	6.1	16.0
	DAPO	75.7	58.0	20.2	50.4	20.6	33.0	3.3	31.5	26.8	35.5
	Dr.GRPO	76.0	59.0	21.0	47.8	**21.5**	34.4	1.6	28.1	**27.3**	35.2
	GRPO	75.1	57.0	**21.7**	49.6	20.3	32.4	**4.2**	26.9	**27.3**	34.9
	ATPO	**77.3**	**59.2**	21.0	**50.6**	21.3	**35.0**	3.4	**34.7**	26.8	**36.6**
Qwen2.5 3B	Base	81.6	58.8	25.0	47.0	24.1	32.8	5.5	33.8	25.0	37.1
	DAPO	85.1	65.4	31.2	53.0	27.6	36.6	6.2	35.9	25.8	40.7
	Dr.GRPO	85.2	63.6	30.1	53.8	26.2	37.4	7.7	39.1	28.8	41.3
	GRPO	**85.7**	64.0	31.2	53.0	**27.9**	37.0	**7.7**	36.5	29.8	41.4
	ATPO	85.3	**67.4**	**32.7**	**54.0**	27.7	**37.2**	7.3	**38.7**	**30.8**	**42.4**
Qwen2.5 7B	Base	87.8	63.6	26.8	53.5	29.9	34.6	8.0	36.6	27.3	40.9
	DAPO	91.0	**77.8**	36.4	63.9	37.9	40.4	14.2	54.1	**34.8**	50.0
	Dr.GRPO	91.1	77.4	38.6	64.7	37.0	40.2	16.8	55.5	29.8	50.1
	GRPO	89.7	77.2	**40.1**	62.6	35.6	38.6	**17.2**	52.6	30.8	49.5
	ATPO	**92.0**	75.8	**40.1**	**66.2**	**38.1**	**41.4**	15.1	**58.8**	32.8	**51.2**
Llama3 8B	Base	43.2	14.3	5.8	9.3	2.1	7.9	0.0	2.1	19.7	11.6
	DAPO	64.4	26.4	**16.2**	24.9	6.5	17.4	0.0	8.1	25.3	21.0
	Dr.GRPO	67.2	25.2	13.6	28.8	6.2	20.0	0.0	13.4	25.8	22.2
	GRPO	64.1	26.0	14.0	26.2	7.3	**20.0**	0.1	10.7	23.7	21.3
	ATPO	**70.3**	**29.6**	12.5	**28.6**	**8.4**	17.2	**0.2**	**12.2**	**30.3**	**23.3**
Mathstral 7B	Base	85.2	53.4	23.9	47.0	18.7	29.4	2.0	23	28.3	34.5
	DAPO	88.9	66.0	**31.6**	**57.9**	29.0	**36.6**	5.0	36.9	34.3	42.9
	Dr.GRPO	90.1	65.4	32.0	58.2	29.5	35.2	6.6	36.6	34.8	43.2
	GRPO	**90.3**	66.0	30.5	53.8	28.6	35.6	6.2	34.7	31.8	41.9
	ATPO	89.5	**66.4**	31.2	56.4	**31.1**	33.8	**9.3**	**42.1**	**37.4**	**44.1**

Table 1: Performance on math reasoning and OOD benchmarks using Qwen2.5, Llama 3, and Mathstral models.

accuracy compared to GRPO, DAPO, and the base model. On *Qwen2.5-3B*, our method achieves 30.8 on GPQA, higher than both DAPO and GRPO. On *Llama3-8B*, our method attains 30.3, exceeding 25.3 for DAPO and 23.7 for GRPO.

5.2 Other Study

Analysis across the benchmarks We further examine training dynamics on the *Llama3-8B* base model by plotting accuracy versus training step across benchmarks (Figure 3). Overall, our method shows steady gains and achieves the strongest final accuracy on the majority of benchmarks, our method shows clear late-stage gains and the strongest final performance: GSM8K reaches 70.3 (vs. 64.4 for DAPO and 64.1 for GRPO), and MATH500 reaches 29.6 (vs. 26.4 and 26.0). Similar trends appear on Gaokao and the contest-style evaluations. Notably, the advantage is most visible after step 80, where GRPO can be competitive early but tends to plateau earlier, while our method continues to improve toward the end of training. For completeness, we include the corresponding training curves for *Qwen2.5-1.5B/3B/7B* in Appendix C, where we observe similar dynamics and our method typically achieves the strongest final performance.

Effect of Scaling factor We report an additional ablation on the scaling factor for in-domain evaluation on the *Qwen2.5-7B* model in Table 2 and *Llama3-8B* in Appendix F. As shown in Figure 2a, larger scaling factors produce a wider spread, and the performance improves as the scaling factor increases until it stabilizes. We observe that the performance essentially converges around a scaling factor of $1/9$, with only small fluctuations in its neighborhood, so we fix the scaling factor to $1/9$ in all other experiments.

Output Length Figure 4 reports the evolution of average response length for *Qwen2.5-3B* during training. We observe that GRPO consistently produces the longest responses, reflecting the well-known verbosity bias induced by uniform token-level advantage assignment. DAPO partially mitigates this effect by reweighting token updates, yet still exhibits relatively long and unstable responses over training. In contrast, ATPO leads to a steady and sustained reduction in response length as training progresses. Here the adaptivity comes from a single global parameter λ, learned jointly with the policy: starting from 0, λ consistently moves to a small negative value, mildly favoring

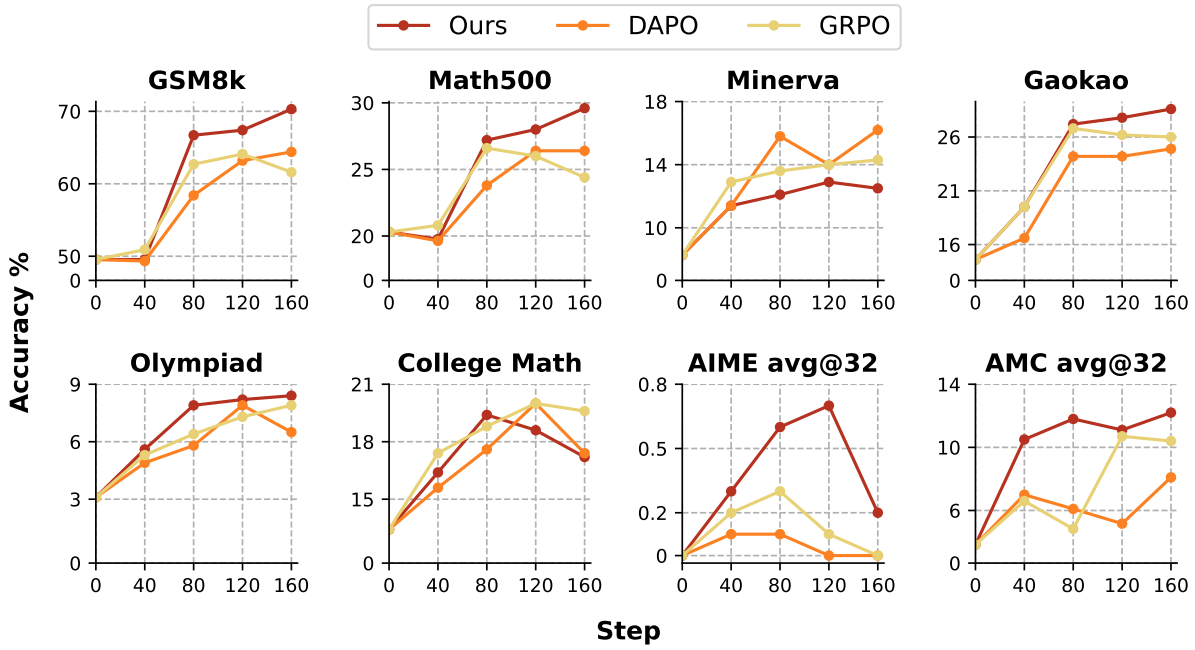


Figure 3: Training curves for the *Llama3-8B* base model across benchmarks. Each panel shows accuracy (%) versus training step for three methods (*Ours*, *DAPO*, and *GRPO*). Steps are {0, 40, 80, 120, 160}.

Scaling	GSM8K	MATH500	Minerva	Gaokao	Olympiad	College	AIME	AMC	Average
1/3	91.1	76.4	39.7	62.1	37.6	40.8	14.0	60.2	52.7
1/9	92.0	75.8	40.1	66.2	38.1	41.4	15.1	58.8	53.4
1/15	91.2	75.4	38.2	61.3	38.2	40.4	16.4	53.4	51.8

Table 2: Effect of the scaling factor on in-domain performance for the *Qwen2.5-7B* model.

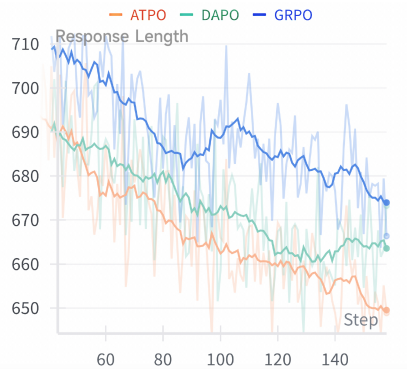


Figure 4: Response length comparison across different policy optimization methods. The lighter lines show the raw training curves, and the darker lines shows the running-average curves.

shorter responses. This learned global preference replaces fixed aggregation heuristics and remains stable throughout training.

Other Study Appendix G presents a qualitative showcase that compares the responses generated by different methods, providing an intuitive illustration of their behavioral differences. Appendix H reports the final values of the learnable parameter λ that different models converge to.

6 Conclusions

In this work, we explored the feasibility of moving beyond manually designed heuristics for token aggregation and instead allowing the model to learn its own token-level preference. We proposed a unified framework that reformulates existing methods, and introduced a tunable parameter λ that explicitly encodes token preference. When $\lambda = 0$, the model is length-neutral; when $\lambda > 0$, the optimization favors longer responses; and when $\lambda < 0$, it favors shorter ones. This learnable preference provides a flexible mechanism to balance different response lengths across tasks, reducing the reliance on fixed aggregation strategies. Empirical results on multiple mathematical reasoning benchmarks demonstrate that our method consistently improves over existing GRPO variants without introducing additional data or computational cost.

7 Limitations

While ATPO provides a principled mechanism for learning token-level preferences in group-based policy optimization, several limitations remain. First, ATPO primarily addresses biases arising from how token-level updates are aggregated, rather than directly modeling the semantic correctness or logical validity of individual reasoning steps. As a result, the method should be viewed as a bias-correction mechanism at the optimization level, complementary to advances in reward design or verifier quality, rather than a replacement for them. Second, although ATPO effectively mitigates length-related verbosity bias, response length captures only one aspect of generation behavior. Other dimensions, such as stylistic variation, factual hallucination, or safety-related behaviors, are not explicitly modeled by the current formulation and may require additional preference signals or constraints.

References

Yuyan Bu, Liangyu Huo, Yi Jing, and Qing Yang. 2025. Beyond excess and deficiency: Adaptive length bias mitigation in reward models for RLHF. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3091–3098, Albuquerque, New Mexico. Association for Computational Linguistics.

Lichang Chen, Chen Zhu, Jiuhai Chen, Davit Soselia, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. ODIN: Disentangled reward mitigates hacking in RLHF. In *Forty-first International Conference on Machine Learning*.

Yi Chen, Yuying Ge, Rui Wang, Yixiao Ge, Junhao Cheng, Ying Shan, and Xihui Liu. 2025. Grpo-care: Consistency-aware reinforcement learning for multimodal reasoning. *arXiv preprint arXiv:2506.16141*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Wenlong Deng, Yi Ren, Danica J Sutherland, Christos Thrampoulidis, and Xiaoxiao Li. 2025. Token hidden reward: Steering exploration-exploitation in grpo training. In *2nd AI for Math Workshop@ ICML 2025*.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025a. Group-in-group policy optimization for LLM agent training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhangyin Feng, Qianglong Chen, Ning Lu, Yongqian Li, Siqi Cheng, Shuangmu Peng, Duyu Tang, Shengcai Liu, and Zhirui Zhang. 2025b. Is PRM necessary? problem-solving RL implicitly induces PRM capability in LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024a. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024b. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. 2025a. Explaining length bias in llm-based preference evaluations. *Preprint*, arXiv:2407.01085.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. 2025b. Explaining length bias in LLM-based preference evaluations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6763–6794, Suzhou, China. Association for Computational Linguistics.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025a. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *Preprint*, arXiv:2503.06749.

Zeyu Huang, Zihan Qiu, Zili Wang, Edoardo Ponti, and Ivan Titov. 2025b. Post-hoc reward calibration: A case study on length bias. In *The Thirteenth International Conference on Learning Representations*.

Feynman Innovations. 2024. Maths-college (revision 9eaa32a).

Youngsoo Jang, Geon-Hyeong Kim, Byoungjip Kim, Yu Jin Kim, Honglak Lee, and Moontae Lee. 2024. Degeneration-free policy optimization: Rl fine-tuning for language models without degeneration. In *Forty-first International Conference on Machine Learning*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. 2022. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numina-math. [https://github.com/project-numina/aimo-progress-prize](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).

Jiate Liu, Yiqin Zhu, Kaiwen Xiao, QIANG FU, Xiao Han, Yang Wei, and Deheng Ye. 2023. RLTF: Reinforcement learning from unit test feedback. *Transactions on Machine Learning Research*.

Jiawei Liu and Lingming Zhang. 2025. Code-r1: Reproducing r1 for code with reliable rewards. https://github.com/ganler/code-r1. GitHub repository, accessed 2025-10-07.

Wei Liu, Yang Bai, Chengcheng Han, Rongxiang Weng, Jun Xu, Xuezhi Cao, Jingang Wang, and Xunliang Cai. 2024. Length desensitization in direct preference optimization. *arXiv preprint arXiv:2409.06411*.

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

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. 2024. Eliminating biased length reliance of direct preference optimization via down-sampled KL divergence. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1047–1067, Miami, Florida, USA. Association for Computational Linguistics.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Greenewald, Jiri Navratil, Jerret Ross, and Jesus Rios. 2025. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. *arXiv preprint arXiv:2505.22257*.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4998–5017, Bangkok, Thailand. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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 *Thirty-seventh Conference on Neural Information Processing Systems*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

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, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297, New York, NY, USA. Association for Computing Machinery.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024a. A long way to go: Investigating length correlations in rlhf. *Preprint*, arXiv:2310.03716.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024b. A long way to go: Investigating length correlations in RLHF. In *First Conference on Language Modeling*.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024c. A long way to go: Investigating length correlations in RLHF. In *First Conference on Language Modeling*.

Hu Wang, Congbo Ma, Ian Reid, and Mohammad Yaqub. 2025. Kalman filter enhanced grpo for reinforcement learning-based language model reasoning. *arXiv preprint arXiv:2505.07527*.

Siwei Wu, Zhongyuan Peng, Xinrun Du, Tuney Zheng, Minghao Liu, Jialong Wu, Jiachen Ma, Yizhi Li, Jian Yang, Wangchunshu Zhou, Qunshu Lin, Junbo Zhao, Zhaoxiang Zhang, Wenhao Huang, Ge Zhang, Chenghua Lin, and J. H. Liu. 2024. A comparative study on reasoning patterns of openai's o1 model. *Preprint*, arXiv:2410.13639.

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2023. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment. *arXiv preprint arXiv:2310.00212*.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *Preprint*, arXiv:2501.09686.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint*.

Kangwen Zhao, Jianfeng Cai, Jinhua Zhu, Ruopei Sun, Dongyun Xue, Wengang Zhou, Li Li, and Houqiang Li. 2025a. Bias fitting to mitigate length bias of reward model in rlhf. *arXiv preprint arXiv:2505.12843*.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, and 1 others. 2025b. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, and 8 others. 2023. Secrets of rlhf in large language models part i: Ppo. *Preprint*, arXiv:2307.04964.

A Chat Template

In this section, we present the chat template used during training. The template follows the format

of role-tagged messages, with system instructions, user input, and assistant output.

Chat Template

```
<|im_start|>system
You are a helpful assistant.
<|im_end|>

<|im_start|>user
{input}
Please reason step by step,
and put your final answer within
\boxed{{}}.
<|im_end|>

<|im_start|>assistant
```

B Benchmark Statistics

The detailed statistics of the nine benchmarks is shown in Table 4, including gsm8k, math500, minerva, gaokao, olympiad, college math, aime24, amc23, and gpqa (diamond).

C Training curves for models across benchmarks

See Figure 5, Figure 6, Figure 7.

D Explanation of the unified form

According to section 3.2, the unified form is

$$\mathcal{J}(\theta) = \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G} f(o_i) \sum_{t=1}^{|o_i|} \min\Big\{ r_{i,t}(\theta)\,\hat{A}_{i,t}, \operatorname{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\}$$

GRPO. As shown in Eq. (4), we have the GRPO objective function as follows:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \min\Big\{ r_{i,t}(\theta)\,\hat{A}_{i,t}, \operatorname{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\}$$

Let

$$\mu = \frac{|o_1| + |o_2| + \ldots + |o_G|}{G} = \frac{1}{G}\sum_{i=1}^{G}|o_i|$$

be the average output length in the group. Multiplying and dividing by μ gives

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) &= \frac{1}{G\cdot\mu}\sum_{i=1}^{G}\frac{\mu}{|o_i|}\sum_{t=1}^{|o_i|} \min\Big\{ r_{i,t}(\theta)\,\hat{A}_{i,t}, \\ &\qquad \operatorname{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\} \\ &= \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\frac{\mu}{|o_i|}\sum_{t=1}^{|o_i|} \min\Big\{ r_{i,t}(\theta) \\ &\qquad \hat{A}_{i,t}, \operatorname{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\}\end{aligned}$$

, where $G\cdot\mu = G\cdot\big(\frac{1}{G}\sum_{i=1}^{G}|o_i|\big) = \sum_{i=1}^{G}|o_i|$.

DAPO. As shown in Eq. (5), we have the DAPO objective function as follows:

$$\mathcal{J}_{\text{DAPO}}(\theta) = \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \min\Big\{ r_{i,t}(\theta)\,\hat{A}_{i,t}, \operatorname{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon'\big)\,\hat{A}_{i,t}\Big\}$$

This already matches the unified form with a constant weighting function:

$$\mathcal{J}_{\text{DAPO}}(\theta) = \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G} 1 \sum_{t=1}^{|o_i|} \min\Big\{ r_{i,t}(\theta) \hat{A}_{i,t}, \operatorname{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\}$$

Dr. GRPO. As shown in Eq. (6), we have the Dr. GRPO objective function as follows:

$$\mathcal{J}_{\text{Dr.GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \min\Big\{ r_{i,t}(\theta) \hat{A}_{i,t}, \operatorname{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\}$$

Using the same μ as above and inserting μ/μ, we obtain

$$\begin{aligned}\mathcal{J}_{\text{Dr.GRPO}}(\theta) &= \frac{1}{G\cdot\mu}\sum_{i=1}^{G}\mu\sum_{t=1}^{|o_i|} \min\Big\{ r_{i,t}(\theta) \\ &\qquad \hat{A}_{i,t}, \operatorname{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\} \\ &= \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\mu\sum_{t=1}^{|o_i|} \min\Big\{ r_{i,t}(\theta) \\ &\qquad \hat{A}_{i,t}, \operatorname{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big\}\end{aligned}$$

Question set	Description
GSM8K	High quality linguistically diverse grade school math word problems (1319)
MATH500	A representative subset from the MATH benchmark, covering challenging competition-style problems across algebra, geometry, probability, and number theory (500)
Minerva	A curated slice of the Minerva benchmark focusing on advanced STEM reasoning problems sourced from undergraduate-level math and science exams (272)
Gaokao	Math questions from Chinese college entrance examinations, requiring multi-step symbolic reasoning (385)
Olympiad	OlympiadBench problems, emphasizing high-school Olympiad-level creative problem solving (675)
College Math	Random subsamples of college-level math exam questions, selected with `random.seed(42)` for reproducibility (500)
AIME24	Problems from the 2024 American Invitational Mathematics Examination, designed for top high-school students (30)
AMC23	Problems from the 2023 American Mathematics Competition, targeting high-school competition-level reasoning (40)
GPQA (Diamond)	A high-quality subset of the Graduate-Level Google-Proof Q&A benchmark, a collection of challenging multiple-choice questions in biology, physics, and chemistry (198)

Table 4: Different benchmarks spanning diverse difficulty levels and coverage, from grade-school word problems to Olympiad- and college-level mathematics, as well as general-domain questions.

E Derivation of the proposed objective function

According to section 3.3, the objective is

$$\begin{aligned}\mathcal{J} &= \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G} f(o_i)\,L_i \\ &= \frac{G}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G} s_i\,L_i\end{aligned}$$

where

$$L_i = \sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\,\hat{A}_{i,t},\ \mathrm{clip}\big(r_{i,t}(\theta),\ 1-\epsilon,\ 1+\epsilon\big)\,\hat{A}_{i,t}\Big)$$

Here, we show the detailed gradient derivation with respect to λ.

First, we want to calculate $\frac{\partial g_i}{\partial \lambda}$, where $g_i = h_i^{\lambda}$. We can substitute h_i^{λ} as $e^{\ln h_i^{\lambda}}$, and

$$\frac{\partial(e^{\ln h_i^{\lambda}})}{\partial \lambda} = \frac{\partial(e^{\lambda\cdot\ln h_i})}{\partial \lambda} = e^{\ln h_i^{\lambda}}\cdot \ln h_i$$

Hence,

$$\boxed{\frac{\partial g_i}{\partial \lambda} = \frac{\partial}{\partial \lambda}\big(h_i^{\lambda}\big) = h_i^{\lambda}\cdot \ln h_i} \tag{18}$$

Then, we will take softmax derivative $\frac{\partial s_i}{\partial \lambda}$, where $s_i = \frac{e^{g_i}}{\sum_{j=1}^{G} e^{g_j}}$.

Let

$$s_i = \frac{e^{g_i}}{Z}, \qquad Z = \sum_{j=1}^{G} e^{g_j}.$$

Then, by the quotient rule,

$$\begin{aligned}\frac{\partial s_i}{\partial \lambda} &= \frac{\partial}{\partial \lambda}\left(\frac{e^{g_i}}{Z}\right) \\ &= \frac{e^{g_i}\,\frac{\partial g_i}{\partial \lambda}\,Z \;-\; e^{g_i}\,\frac{\partial Z}{\partial \lambda}}{Z^2}.\end{aligned}$$

Using $s_i = \frac{e^{g_i}}{Z}$, we rewrite

$$\frac{\partial s_i}{\partial \lambda} = s_i\left(\frac{\partial g_i}{\partial \lambda} \;-\; \frac{1}{Z}\,\frac{\partial Z}{\partial \lambda}\right).$$

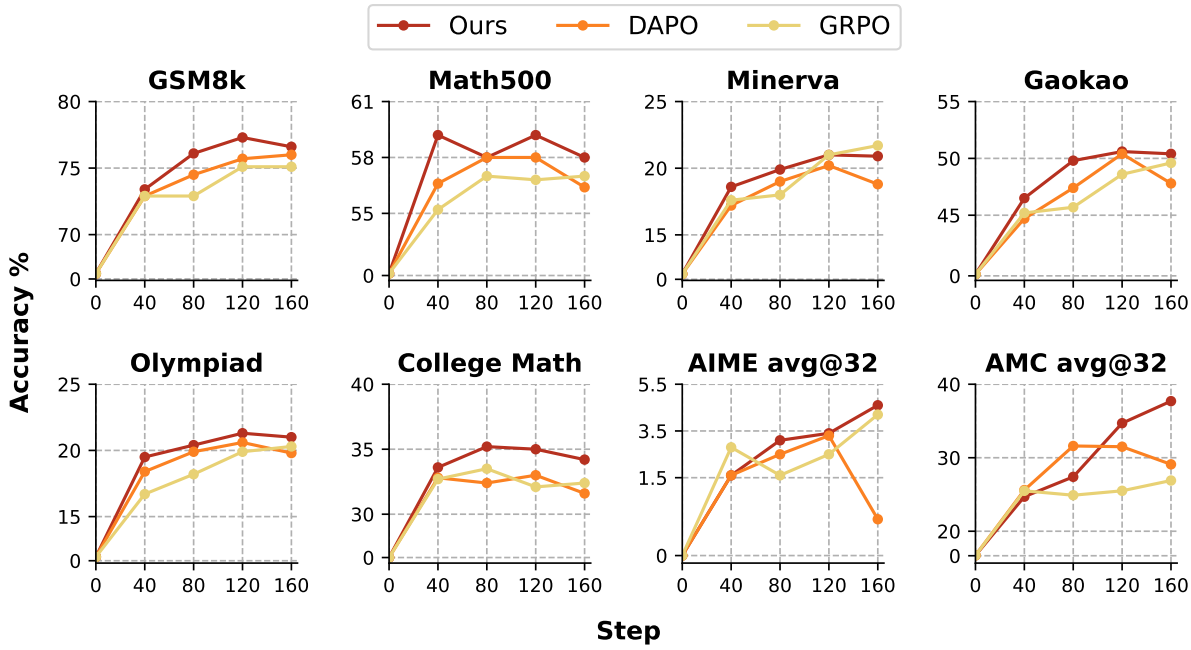


Figure 5: Training curves for the *Qwen2.5-1.5B* base model across benchmarks. Each panel shows accuracy (%) versus training step for three methods (*ATPO*, *DAPO*, and *GRPO*). Steps are {0, 40, 80, 120, 160}.

Moreover,

$$\begin{aligned}\frac{1}{Z}\frac{\partial Z}{\partial \lambda} &= \frac{1}{Z}\sum_{j=1}^{G}\frac{\partial e^{g_j}}{\partial g_j}\frac{\partial g_j}{\partial \lambda}\\ &= \frac{1}{Z}\sum_{j=1}^{G} e^{g_j}\frac{\partial g_j}{\partial \lambda}\\ &= \sum_{j=1}^{G}\frac{e^{g_j}}{Z}\frac{\partial g_j}{\partial \lambda}\\ &= \sum_{j=1}^{G} s_j\frac{\partial g_j}{\partial \lambda}.\end{aligned}$$

Substituting back gives the desired result:

$$\boxed{\frac{\partial s_i}{\partial \lambda} = s_i\left(\frac{\partial g_i}{\partial \lambda} - \sum_{j=1}^{G} s_j\frac{\partial g_j}{\partial \lambda}\right)} \tag{19}$$

Since $f(o_i) = G\,s_i$,

$$\frac{\partial f(o_i)}{\partial s_i} = G \tag{20}$$

Therefore, according to the Chain Rule,

$$\begin{aligned}&\frac{\partial \mathcal{J}}{\partial \lambda}\\ &= \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G} L_i\frac{\partial f(o_i)}{\partial s_i}\cdot\frac{\partial s_i}{\partial \lambda}\\ &= \frac{G}{\sum_{i=1}^{G}|o_i|}\left[\sum_{i=1}^{G} L_i\cdot s_i\Big(\frac{\partial g_i}{\partial \lambda} - \sum_{j=1}^{G} s_j\frac{\partial g_j}{\partial \lambda}\Big)\right]\end{aligned} \tag{21}$$

Plugging in $\frac{\partial g_i}{\partial \lambda} = h_i^{\lambda}\ln h_i$ yields

$$\begin{aligned}&\frac{\partial \mathcal{J}}{\partial \lambda}\\ &= \frac{G}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G} L_i\cdot s_i\Big(h_i^{\lambda}\ln h_i - \sum_{j=1}^{G} s_j h_j^{\lambda}\ln h_j\Big)\end{aligned} \tag{22}$$

F Effect of Scaling Factor

Table 5 reports the effect of different scaling factors for *Llama3-8B*. The trend closely matches the in-domain results for *Qwen2.5-7B* in the main text: performance improves as the scaling factor increases, then plateaus once the scaling factor reaches roughly $1/9$. Beyond this point, the scores only exhibit small fluctuations around the optimum.

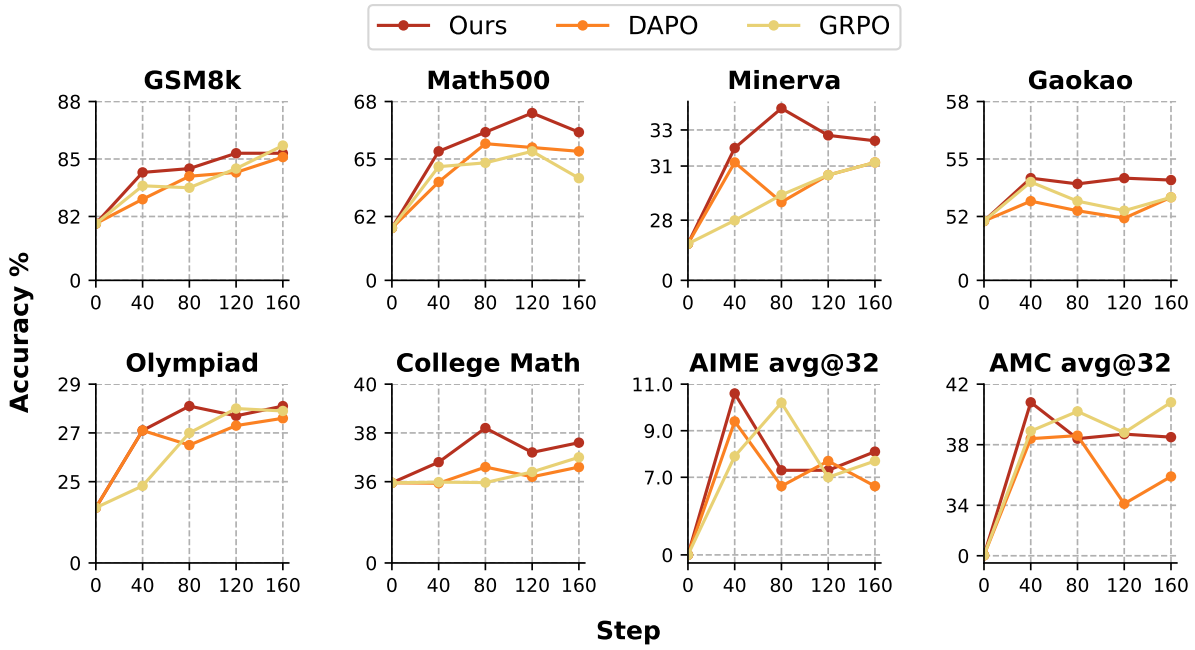


Figure 6: Training curves for the *Qwen2.5-3B* base model across benchmarks. Each panel shows accuracy (%) versus training step for three methods (*Ours*, *DAPO*, and *GRPO*). Steps are {0, 40, 80, 120, 160}.

Scaling	GSM8K	MATH500	Minerva	Gaokao	Olympiad	College	AIME	AMC	Average
1/3	66.9	24.6	14.0	27.8	6.7	17.6	4.0	15.9	22.2
1/9	70.3	29.6	12.5	28.6	8.4	17.2	0.2	12.2	22.4
1/15	61.9	26.0	15.8	27.0	7.6	20.4	0.0	9.7	21.1

Table 5: Effect of the scaling factor on in-domain performance for the *Llama3-8B* model.

Model	λ
Qwen2.5-1.5B	-0.059
Qwen2.5-3B	-0.041
Qwen2.5-7B	-0.064
Llama3-8B	-0.186
Mathstral-7B	-0.229

Table 6: Final λ value for each model in our experiments.

This corroborates our choice to fix the scaling factor to $1/9$ for all other models rather than tuning it separately per backbone.

G Sample Output of Mathstral-7B

Table 7 and Table 8 present two representative scope-sensitive arithmetic word problems, where the models must correctly track how quantifiers (e.g., "every weekday", "twice as old") and time expressions (e.g., "in 3 years") apply in context. For the water-consumption problem, the question requires correctly distinguishing John's weekday routine from his weekend routine (where dinner water is replaced by soda), which can easily lead to incorrect counting if the weekend scope is misapplied. For the age problem, the model must correctly compose multiple temporal modifiers ("will turn 12 in 3 years", "had a son when he was 19", "born when John was 32") to recover John's current age and then the youngest son's age in 3 years. We report the raw outputs generated by Mathstral-7B under GRPO, DAPO, and our ATPO, and annotate the key reasoning spans: incorrect reasoning is highlighted in red, while the corresponding correct reasoning is highlighted in blue. As shown, GRPO and DAPO both make scope errors on the weekend counting in the first problem and on the temporal relations in the second, whereas ATPO produces the correct breakdown and final answer in both cases.

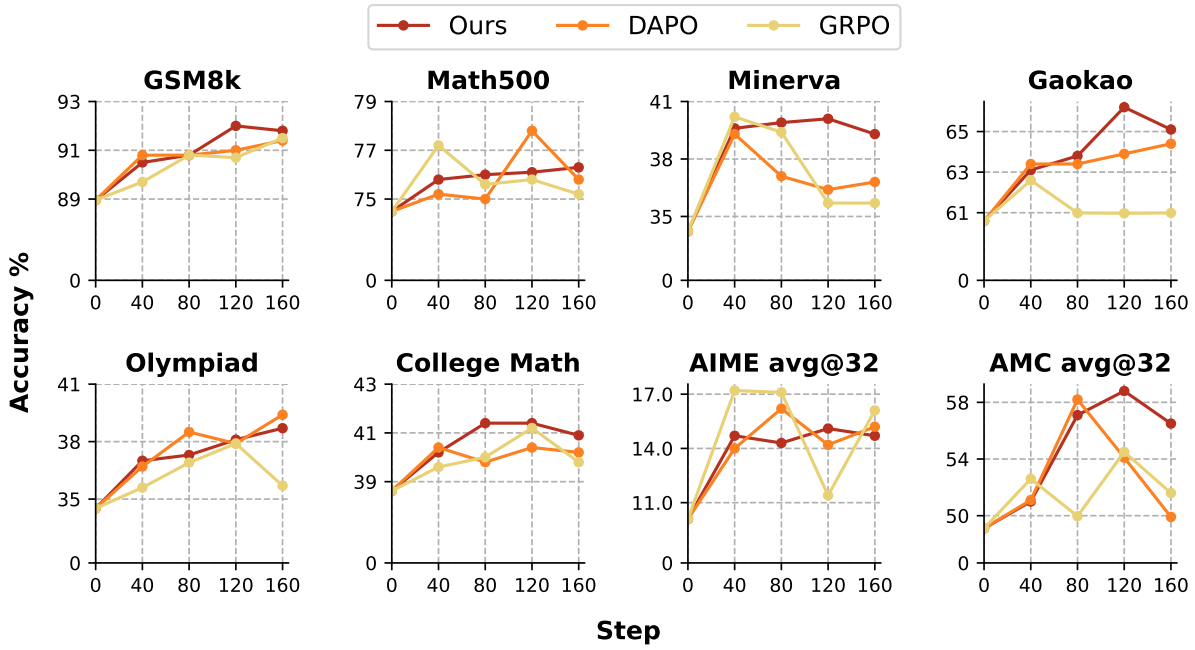


Figure 7: Training curves for the *Qwen2.5-7B* base model across benchmarks. Each panel shows accuracy (%) versus training step for three methods (*Ours*, *DAPO*, and *GRPO*). Steps are {0, 40, 80, 120, 160}.

H Final λ values for different models

Table 6 summarizes the final values of the learnable parameter λ for each backbone model. All runs initialize λ at 0 and update it jointly with the policy parameters throughout training. Across architectures, λ consistently converges to a small negative value, with larger models (Llama3-8B and Mathstral-7B) exhibiting slightly larger magnitudes. The absolute values remain modest (between approximately -0.23 and -0.04), indicating that ATPO does not rely on extreme settings of λ to obtain improvements. Instead, the method learns a relatively stable preference that is broadly consistent across different model sizes and architectures.

I AI Assistants In Research Or Writing

Our use of LLMs was limited to grammar and style editing; we did not employ them for research functions (e.g., idea generation). We supplied the original manuscript to OpenAI GPT-5 and asked it to improve clarity and tone to make our work a more professional research paper. All generated text was subsequently checked for factual correctness and fidelity to the source.

Question: John likes to have a glass of water with breakfast, lunch and dinner. Finally, he has one before he goes to bed as well. John does this every weekday, but on the weekends he likes to relax and have a soda with dinner instead. How many glasses of water does John drink in a week?

Method	**Model output**
GRPO	Let's calculate the number of glasses of water John drinks during the weekdays first. Since he drinks water with each meal (breakfast, lunch, and dinner) and before bed, that's a total of 4 glasses of water per day. There are 5 weekdays in a week. So, the total number of glasses of water John drinks during the weekdays is: 4 glasses/day * 5 days = 20 glasses. On the weekends, John doesn't drink water with breakfast, lunch, or dinner. He only has a glass of water before bed on both days. So, the total number of glasses of water John drinks during the weekends is: 1 glass/day * 2 days = 2 glasses. Adding the glasses of water from the weekdays and the weekends together, we get the total number of glasses of water John drinks in a week: 20 glasses (weekdays) + 2 glasses (weekends) = 22 glasses. So, John drinks a total of $\boxed{22}$ glasses of water in a week.
DAPO	To calculate the number of glasses of water John drinks in a week, we need to consider his habits on weekdays and weekends separately. On weekdays (Monday to Friday), John drinks water with each of his meals (breakfast, lunch, dinner) and before bed. That's a total of 4 glasses of water per day. Since there are 5 weekdays, the total number of glasses of water he drinks in a week on weekdays is: 4 glasses/day times 5 days = 20 glasses. On weekends (Saturday and Sunday), John does not drink water with breakfast, lunch, or dinner. Instead, he has a soda with dinner. Therefore, he only drinks water before bed on both days, which is 1 glass per day. The total number of glasses of water he drinks in a week on weekends is: 1 glass/day times 2 days = 2 glasses. Adding the glasses of water from weekdays and weekends together gives us the total number of glasses of water John drinks in a week:20 glasses (weekdays) + 2 glasses (weekends) = 22 glasses. Therefore, John drinks a total of $\boxed{22}$ glasses of water in a week.
ATPO (Ours)	To calculate the total number of glasses of water John drinks in a week, we follow these steps: 1. Calculate the number of glasses of water John drinks each weekday: He drinks water with breakfast, lunch, dinner, and before bed, which totals to 4 glasses per day. 2. Since there are 5 weekdays, the total number of glasses of water he drinks during the weekdays is: 4 glasses/day * 5 days = 20 glasses. 3. On the weekends, John does not drink water with dinner. Since there are 2 days in a weekend, and he drinks water with breakfast, lunch, and before bed, he drinks: 3 glasses/day * 2 days = 6 glasses. 4. Total: $20 + 6 = 26$. Therefore, John drinks a total of $\boxed{26}$ glasses of water in a week.

Table 7: Comparison of GRPO, DAPO, and ATPO (ours) on a word problem about weekly water consumption, using outputs generated by Mathstral-7B. The question requires scope-sensitive reasoning over weekday vs. weekend quantifiers. Incorrect spans are highlighted in red, and the corresponding correct reasoning under ATPO is highlighted in blue.

Question: John had a son James when he was 19. James is now twice as old as his sister Dora, who will turn 12 in 3 years. How old will John's youngest son, who was born when John was 32, be in 3 years?	
Method	**Model output**
GRPO	First, let's determine Dora's current age. If Dora will turn 12 in 3 years, she is currently 12 - 3 = 9 years old. Next, we know that James is twice as old as Dora. So, James' current age is 2 * 9 = 18 years old. John had James when he was 19, so the current year is 19 + 18 = 37 years since John had James. John's youngest son was born when John was 32. So, John's youngest son was born in 37 - 19 = 18 years ago. Therefore, John's youngest son is currently 18 years old. In 3 years, John's youngest son will be 18 + 3 = 21 years old. So, the age of John's youngest son in 3 years will be $\boxed{21}$ years old.
DAPO	Let's break down the information given step by step: 1. James is now twice as old as his sister Dora. 2. Dora will turn 12 in 3 years, which means Dora is currently 12 - 3 = 9 years old. 3. Since James is twice as old as Dora, James is currently 2 * 9 = 18 years old. 4. John had a son James when he was 19, so the difference in age between John and James is 18 - 19 = -1 years (James is actually 1 year younger than John). 5. To find out how old John is now, we need to add the age difference between James and Dora to Dora's current age and then add 1 year (since James is 1 year younger than John): 9 (Dora's age) + 1 (age difference between James and Dora) + 1 (James is 1 year younger than John) = 11 years. So, John is currently 19 (his age when James was born) + 11 = 30 years old. 6. John's youngest son was born when John was 32, so the youngest son is currently 30 (John's current age) - 32 (John's age when the son was born) = -2 years old. However, since we cannot have a negative age, this means the youngest son was actually born when John was 32, and he is currently 0 years old (just born). 7. In 3 years, John's youngest son will be 0 (current age) + 3 = $\boxed{3}$ years old.
ATPO (Ours)	Let's break down the information given: 1. James is now twice as old as Dora. 2. Dora will turn 12 in 3 years, which means Dora is currently 12 - 3 = 9 years old. 3. Since James is twice as old as Dora, James is 2 * 9 = 18 years old. 4. John had a son James when he was 19, so the current year is 19 + 18 = 37 years. 5. John's youngest son was born when John was 32. John turned 32 when he was 32 - 19 = 13 years older than James's birth year. 6. Therefore, the youngest son was born in 19 + 13 = 32 years from James's birth year. 7. Since the current year is 37 years from James's birth year, the youngest son was born in 37 - 32 = 5 years ago. 8. In 3 years, the youngest son will be 5 + 3 = $\boxed{8}$ years old.

Table 8: Comparison of GRPO, DAPO, and ATPO (ours) on a word problem about ages, using outputs generated by Mathstral-7B. The question requires scope-sensitive reasoning about temporal modifiers (e.g., "when John was 32"). Incorrect spans are highlighted in red, and the corresponding correct reasoning under ATPO is highlighted in blue.