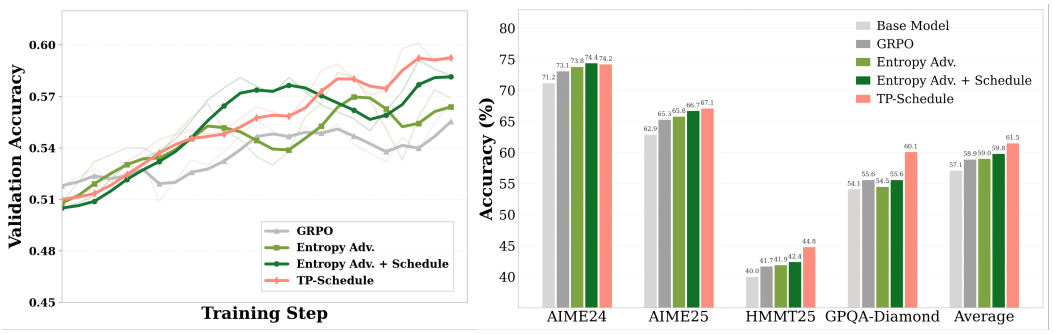
Not Only Where, But When: Temporal Scheduling for RLVR

Jinghao Zhang [1,2,4] **Ruilin Li** [2,3,4] **Feng Zhao** [1†] **Jiaqi Wang** [2,4†]
[1]MoE Key Lab of BIPC, University of Science and Technology of China
[2]Shanghai Innovation Institute [3]Wuhan University [4]JD.com
https://github.com/Jinghaoleven/RLVR-Schedule

Abstract

Reinforcement learning with verifiable rewards (RLVR) has become a core technique for post-training of Large Language Models (LLMs). While policy optimization is driven by all sampled tokens under a globally broadcast scalar reward, the heterogeneous policy behaviors exhibited along trajectories are largely overlooked without differentiation. Existing works address this by credit allocation, including token-level advantage reweighting, and selective token optimization, however, the allocation criterion are principally stagnant throughout training, limiting resilient policy evolution. In this work, we argue that *when* learning signals are scheduled can be as important as *where* they are allocated across tokens, and introduce the temporal dimension that scheduling the credit allocation criteria over the course of RLVR optimization. We find that prioritizing targeted tokens emphasized with specific policy behaviors, and gradually attenuating toward general optimization leads to more stable and efficient learning dynamics. Furthermore, we show that simple trajectory percentiles provide a natural perspective for distinguishing policy behaviors, and works effectively with temporal scheduling. Our analysis reveals that standard optimization substantially sacrifices policy entropy when simultaneously accommodating heterogeneous behaviors, whereas temporal scheduling yields healthier policy evolution dynamics. Experiments across mathematical and general reasoning benchmarks demonstrate consistent improvements, suggesting that temporal scheduling constitutes a promising optimization dimension.



(a) Validation accuracy in training **(b) Performance comparison on reasoning benchmarks**

Figure 1: Performance and training dynamics on Qwen3-4B. Temporal scheduling stabilizes training across existing credit allocation methods, including entropy-based proxy and proposed trajectory-percentile proxy, while consistently raising performance plateaus and outperforming vanilla GRPO and corresponding static credit allocation baselines on representative reasoning benchmarks.

†Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

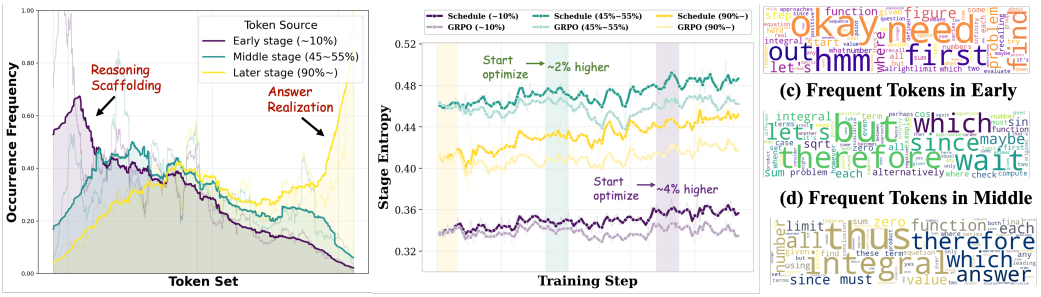
Reinforcement learning with verifiable rewards (RLVR) has emerged as a central paradigm for post-training of Large Language Models (LLMs), yielding substantial improvements in reasoning performance across a wide range of tasks [Guo et al., 2025, Comanici et al., 2025, Yang et al., 2025a, Team et al., 2026]. Unlike supervised fine-tuning, where dense next-token supervision provides explicit learning signals along the trajectory, RLVR typically broadcasts a single scalar reward across the full sequence, engaging all sampled tokens to participate equally in policy optimization. However, the heterogeneous policy behaviors exhibited along the trajectory are largely overlooked, such as reasoning scaffolding and answer convergence, and are optimized roughly without differentiation.

To encourage effective optimization, existing works address this by introducing credit allocation, supposed to identify where the learning signals are distributed within the sampled response. Representative approaches include Process Reward Models (PRMs) that provide step-level feedbacks [Zou et al., 2025, Zhao et al., 2025], while the heavy annotation cost and computation burden pose a significant scalability bottlenecks. More recent works instead focus on token-level signals derived from policy proxies for credit allocation, such as token entropy [Cheng et al., 2025], likelihood [Damani et al., 2025, Yang et al., 2025b], and KL divergence [Ma et al., 2026, Huang et al., 2026]. These methods typically identify specific policy behaviors, such as logic connection and reflection, with predefined allocation criteria, and reweight token advantages for preference in policy optimization, resulting in improved performance. Consistently, parallel studies further demonstrate that optimizing only a subset of critical tokens can outperform full-token optimization [Wang et al., 2025], reinforcing the view that policy gradients accumulated uniformly from all sampled tokens are suboptimal, and could be more effective through more targeted optimization criteria.

Despite progress, existing credit allocation methods are primarily restricted to stagnant allocation criteria throughout training, where once a token-level proxy is applied, the corresponding targeted policy behavior is continuously emphasized during the entire optimization. In view of this, the effectiveness of policy optimization is fundamentally determined by how well the adopted proxy in capturing the informative learning signals, while the behaviors outside the proxy identification could be suppressed for insufficient optimization. We thereby asking, *instead of persistently pursuing effective proxies for directing policy optimization, are there other potential optimization dimensions that have been underexplored.* From the perspective of training dynamics, we advocate that a single credit allocation criteria should not dominate the entire policy optimization process in practice, which is essentially different from their original design choice within individual sample.

Building on this principle, we revisit RLVR optimization from temporal dimension and argue that *when* learning signals are scheduled throughout training can be as important as *where* they are allocated across sampled tokens. Our key idea is to schedule the allocation criteria over the course of optimization, which prioritizing targeted tokens that emphasized by corresponding allocation criteria, and gradually attenuating toward general optimization. We show that temporally scheduling the allocation criteria complements existing credit allocation methods, including both advantage reweighting, and sparse token optimization. Furthermore, we identify that simple trajectory percentiles serve as an effective criteria for scheduling policy optimization, supported by its natural perspective in distinguishing heterogeneous policy behaviors. Specifically, we schedule from later tokens for trustworthy continuation, and incrementally incorporate earlier tokens for reliable reasoning scaffolding. In addition, we reveal that the policy entropy is largely sacrificed in standard RLVR optimization for accommodating heterogeneous policy behaviors, whereas temporal scheduling mitigate this effect with healthier optimization dynamics. Experiments across model scales and RLVR algorithms demonstrate consistent improvements, suggesting a promising dimension for RLVR optimization.

The contributions of this work can be summarized as follows: (1) We identify the temporal dimension beyond conventional stagnant credit allocation paradigm for RLVR optimization, and propose temporally scheduling the allocation criteria over the course of training, that prioritizing credit allocated tokens, and gradually attenuating toward general optimization. (2) Temporal scheduling complements existing credit allocation methods, including continuous advantage reweighting, and discrete sparse token optimization. We further identify simple trajectory percentiles as a natural perspective in distinguishing policy behaviors, which works effectively with temporal scheduling. (3) Experiments across models scales, RL algorithms, and credit allocation strategies show consistent improvement, and further analysis show that temporal scheduling substantially mitigates the sacrificed policy entropy induced by accommodating heterogeneous policy behaviors in RLVR optimization.



(a) Token Distribution across Trajectory (b) Train-Time Entropy Dynamics (c) Frequent Tokens in Early (d) Frequent Tokens in Middle (e) Frequent Tokens in Later

Figure 2: Policy behavior analysis and corresponding training dynamics. (a) Histogram of sampled token frequency across different trajectory positions. The x-axis represents the union of all sampled tokens, sorted by *TP-Score* (Eq. 5) to approximate the most likely position of each token along the trajectory. While different trajectory position exhibit substantially different policy behavior with respective token distribution. (b) Entropy dynamics of tokens at different trajectory percentiles. Tokens from different positions exhibit different entropy profiles, while temporal scheduling consistently preserves higher entropy throughout training. (c)-(e) Textual visualization of trajectory positions.

2 Preliminaries

Reinforcement learning with verifiable rewards (RLVR) has emerged as a common paradigm for post-training of large language models. Consider a language model parameterized by π_θ. Given a prompt x sampled from data distribution $\mathcal{D}$, the policy generates a response $y = (y_1, \ldots, y_T)$ of length T. A verifier then assigns a global scalar reward $r(x, y) \in \mathbb{R}$ that measures the quality of the sampled response. The RLVR objective is to maximize the expected reward

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[r(x, y)\right]. \tag{1}$$

Group Relative Policy Optimization (GRPO). GRPO [Shao et al., 2024] simplifies the Proximal Policy Optimization (PPO) [Schulman et al., 2017] by eliminating the critic model for advantage estimation. For each prompt, GRPO samples a group of response $\{\mathbf{y}_i\}_{i=1}^G$ and estimates the advantage of each response in a group-relative manner, and the clipped surrogate objective is preserved as PPO:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{\boldsymbol{y}_i\}_{i=1}^G\sim\pi_\theta(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\left(w_{i,t}(\theta)\widehat{A}_i,\ \text{clip}\left(w_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\widehat{A}_i\right)\right],$$

where $w_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(x_{i,t}|x, y_{i,<t})}$ is the importance sampling ratio between the current and old policy models, and is typically approximate 1 under on-policy optimization. The advantage $\widehat{A}_i$ is shared across all tokens within response $\boldsymbol{y}_i$, and the effective gradient of GRPO objective is derived as:

$$\nabla_\theta\mathcal{J}_{\text{GRPO}}(\theta) = \nabla_\theta\mathbb{E}_{x\sim\mathcal{D},\, \{y_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}w_{i,t}(\theta)\widehat{A}_{i,t}\right] \tag{2}$$

$$= \mathbb{E}_{x\sim\mathcal{D},\, \{y_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\widehat{A}_i\cdot\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})}\nabla_\theta\log\pi_\theta(y_{i,t}|x, y_{i,<t})\right], \tag{3}$$

which is typically accumulated as token-level average in driving policy optimization. While the weighted importance sampling ratio $w_{i,t}(\theta)$ primarily serves as a distributional correction term in off-policy setting, rather than indicating respective token-level importance.

Token-Level Policy Optimization. To encourage efficient policy optimization, recent methods focus on allocating globally distributed advantage $\widehat{A}$ over tokens for emphasizing token importance, as

$$\nabla_\theta\mathcal{J}_{\text{GRPO-credit}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{y_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\widehat{A}_i\cdot\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\psi(\mu_{i,t})g_{i,t}(\theta)\right], \tag{4}$$

where the $g_{i,t}(\theta) = w_{i,t}(\theta) \cdot \nabla_\theta \log \pi_\theta(y_{i,t}|x, y_{i,<t})$ denotes the per token policy gradient, $\mu_{i,t}$ denotes the derived policy proxy for token $y_{i,t}$, and ψ is a wrap function for proxy utilization. While the allocated $\widehat{A}_i \cdot \psi(\mu_{i,t})$ effectively determine the magnitude of each token in contributing to policy optimization. We mainly identify two types of credit allocation paradigms as following:

- Advantage Reweighting: The credit allocation factor $\mu_{i,t}$ can derive from diverse policy-derived proxies, such as token entropy [Cheng et al., 2025] and ψ typically denotes continuous function.
- Sparse Optimization: [Wang et al., 2025] only optimize a subset of critical tokens that selected by $\mu_{i,t}$, and ψ typically formed as indicator function for deriving binary mask.

Trajectory Percentile Score. To investigate whether the policy π_θ emerges distinct behaviors along the trajectory, we introduce the Trajectory Percentile Score (***TP-Score***), that characterizes the most likely position the token is sampled along the trajectory. Consider a set of predefined position ranges $\mathcal{P} = \{(p_i - \delta, p_i + \delta)\}_{i=1}^N$ along the trajectory, where $p_i \in (0, 1]$ denotes the center of i-th range and normalized by trajectory length, δ denotes the radius. We record the occurrence frequency of each sampled tokens y within each position ranges, denoted as $c_i(y)$, and the ***TP-Score*** is as

$$\mathbb{E}_{i\sim\tilde{c}_i(y)}[p_i] = \frac{\sum_i c_i(y) \cdot p_i}{\sum_i c_i(y)}, \tag{5}$$

where $\tilde{c}_i(y) = \frac{c_i(y)}{\sum_j c_j(y)}$ denotes the approximate probability of token y sampled at i-th position range, and smaller ***TP-Score*** correspond to earlier positions and larger values correspond to later.

3 Method

We first analyze the distribution of tokens sampled at different positions, showing that trajectory percentile serves as a natural perspective in distinguishing policy behaviors and revealing distinct training dynamics. To explore effective RLVR optimization, we introduce temporal dimension that schedules token optimization over the course of training, complementing existing stagnant credit allocation methods, including advantages reweighting and sparse optimization. Finally, we show that simple trajectory percentiles can effectively integrated with temporal scheduling.

3.1 Demystifying Policy Behavior Through Trajectory Percentile

Previous studies have shown that the policy exhibits distinct behaviors across sampled tokens, such as forking tokens for logical connection and following tokens for sentence completion [Wang et al., 2025], which have been effectively demystified through token entropy, and broadly adopted in subsequent works as proxies for credit allocation [Cheng et al., 2025]. In this work, we examine whether policy behaviors can be naturally distinguished along the trajectory. We use Qwen3-4B [Yang et al., 2025a] to generate rollouts on a challenging 78k subset curated from OpenMathReasoning [Moshkov et al., 2025] and DeepMath-103K [He et al., 2025], which have been processed through deduplication, resulting in 584M response tokens in total. We then introduce three representative trajectory position ranges, i.e., $\{p_i\}_{i=1}^N = \{0.05, 0.5, 0.95\}$ as described in Sec. 2, and the range radius δ is set by 0.05. We show the sampled token distribution at different position ranges in Figure 2 (a), where the x-axis represents the union of all sampled tokens within $\mathcal{P}$, sorted by the ***TP-Score*** to approximate the trajectory position where the tokens are most likely to be sampled, as detailed in Sec. 2.

Trajectory percentiles provide a natural perspective in distinguishing policy behavior. We observe substantial differences in sampled token distributions across trajectory percentiles, where the tokens sampled from early percentiles, i.e., $p_i = 0.05$, predominantly occur near the beginning of the trajectory, and rarely appear at later positions. While tokens sampled from middle and later percentiles exhibit similar predominant sampling behavior around the corresponding positions, suggesting that heterogeneous policy behaviors can be naturally distinguished along the trajectory.

Trajectory percentile anchors the policy entropy in orchestrating optimization. In Figure 2 (b), we show that the policy entropy diverge substantially across different trajectory percentiles, both at initialization and throughout training, suggesting that trajectory percentiles provide an effective anchor of policy entropy dynamics for accommodating optimization. In particular, early trajectory percentiles exhibit the lowest entropy, while the middle percentiles consistently maintain the highest

Algorithm 1 Temporal Scheduling for Credit Allocation

Inputs: Training data $\mathcal{D} = \{(x, y)\}$, policy π_θ, total training steps N, token proxy μ_t, credit allocation criteria $\psi(\cdot)$, schedule function $\mathcal{S}(\cdot)$, temporal range τ_{low}, τ_{high}.
Outputs: Optimized policy π_θ.
for step $= 1, \ldots, N$ **do**
  Determine the scheduled progress $\bar{\tau}$ from Eq. 6.
  Generate rollouts for batch data from $\mathcal{D}$, and estimate the credit allocation factor for each sampled token by $\psi(\mu_t)$.
  Schedule the allocation using Eq. 7 for advantage reweighting or Eq. 8 for sparse optimization.
  Optimize policy π_θ on batch data with scheduled advantage using Eq. 10.
end for
return policy π_θ.

entropy. We provide textual visualizations in Figure 2 (c)-(e), where distinct policy behaviors can be observed across trajectory. Early percentiles primarily correspond to reasoning scaffolding behavior, and tend to follow constrained patterns in the absence of sufficient context. While the middle percentiles are more associated with reflection and critical lighting behavior, exhibiting higher exploratory characteristics under intermediate context states. Later percentiles are mainly responsible for answer realization and summarization, exhibiting entropy falling in between.

3.2 Temporal Scheduling of Credit Allocation

RLVR optimizes the policy by broadcasting a scalar reward over the full response, providing reliable directional supervision while offering limited token-level differentiation in credit magnitude along the trajectory [Yang et al., 2026]. Existing works address this by introducing credit allocation strategies that typically leverage policy-derived proxies, such as token entropy, to reweight token-level advantages and reinforce specific policy behaviors, such as logical transitions and reflection. However, the underlying allocation criterion are principally stagnant throughout training, limiting resilient policy evolution and performance ceiling, as illustrated in Figure 1. Building upon this, we expose that temporally scheduling the allocation criterion over the course of training provides a promising dimension for policy optimization, where the scheduled credit allocation is formed as:

$$\widehat{f}_t(\tau) = \psi_{\mathcal{S}(\bar{\tau})}(\mu_t), \qquad \bar{\tau} = \frac{\mathrm{clip}(\tau,\, \tau_{\mathrm{low}},\, \tau_{\mathrm{high}}) - \tau_{\mathrm{low}}}{\tau_{\mathrm{high}} - \tau_{\mathrm{low}}} \in [0, 1], \tag{6}$$

where μ_t is a policy-derived proxy and ψ is a wrap function for proxy utilization, as described in Sec. 2. We denote the original credit allocation factor yielded from existing methods over token y_t by $\psi(\mu_t)$, and its scheduled counterpart by $\psi_{\mathcal{S}(\bar{\tau})}(\mu_t)$. $\tau \in [0, 1]$ denotes training progress normalized by the total number of training steps, and $0 \leq \tau_{\mathrm{low}} < \tau_{\mathrm{high}} \leq 1$ specifies the scheduling range. The schedule $\mathcal{S}(\bar{\tau}) \in [0, 1]$ is monotonically decreasing, and satisfies $\mathcal{S}(0) = 1$ to $\mathcal{S}(1) = 0$. We show that the scheduled credit $\psi_{\mathcal{S}(\bar{\tau})}$ starts from the most targeted allocation criterion and progressively attenuates toward general optimization over the course of training.

Integrating into Policy Optimization. The scheduled credit allocation factor $\psi_{\mathcal{S}(\bar{\tau})}(\mu_t)$ serves as a drop-in replacement for the original factor $\psi(\mu_t)$ in policy optimization, as in Eq. 4. We demonstrate the applicability of temporal scheduling for two representative credit allocation paradigms: *advantage reweighting* and *sparse optimization*. As described in Sec. 2, the former typically defines $\psi(\mu_t)$ as a continuous non-negative function that rescales the token-level advantage without altering its sign. While the latter instead uses an indicator function $\psi(\mu_t) = \mathbb{I}(\mu_t \geq \varepsilon)$ with threshold ε, yielding binary masks that selects a subset of tokens for optimization. Incorporating temporal scheduling into these two paradigms extends Eq 6 as

$$\text{Continuous}: \quad \widehat{f}_t(\tau) = 1 + \mathcal{S}(\bar{\tau})\big(\psi(\mu_t) - 1\big), \qquad \psi(\mu_t) \geq 0, \tag{7}$$

$$\text{Discrete}: \quad \widehat{f}_t(\tau) = \mathbb{I}(\mu_t \geq \mathcal{S}(\bar{\tau})\varepsilon)\,, \qquad \widehat{f}_t(\tau) \in \{0, 1\}. \tag{8}$$

For continuous scheduling, policy optimization starts from the most targeted allocation criterion $\psi(\mu_t)$ and gradually attenuates toward general optimization, corresponding to an uniform reweighting

factor of 1. For discrete scheduling, we instead schedule the selection threshold ε over the course of training, gradually relaxing the criterion and including more tokens into policy optimization.

3.3 Scheduling over Trajectory Percentiles

Instead of deriving the proxy μ_t from prevalent policy-dependent statistics such as token entropy to identify informative learning signals, we show that temporal scheduling works more effectively with simple trajectory percentile proxy. As analyzed in Sec 3.1, different trajectory percentiles exhibit substantially distinct sampled token distribution, providing a natural perspective in distinguishing heterogeneous policy behaviors, and evolving optimization throughout training. In particular, we prioritize later-percentile tokens that are closer to final answer realization, and gradually expand optimization towards earlier tokens that responsible for reasoning scaffolding. The percentile schedule over trajectory follows the discrete allocation scheduling in Eq. 8, and is formed as

$$\widehat{f}_{t(\tau)} = \mathbb{I}\left[\mu_t \geq \mathcal{S}(\bar{\tau}) \cdot 1\right], \quad \text{where } \mu_t = \frac{t}{T} \in (0, 1], \tag{9}$$

where μ_t is defined as trajectory percentile of token y_t, normalized by the response length T of trajectory y. $\varepsilon = 1$ denotes the maximal percentile threshold at initialization, and $\mathcal{S}(\bar{\tau}) \in [0, 1]$ monotonically decrease the percentile criterion to include earlier tokens into optimization. In case of GRPO, the effective policy gradients under temporal scheduling over trajectory percentile is given by

$$\nabla_\theta \mathcal{J}_{\text{GRPO-schedule}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{G}\sum_{i=1}^{G} \widehat{A}_i \cdot \frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \mathbb{I}\left[\frac{t}{|y_i|} \geq \mathcal{S}(\bar{\tau}) \cdot 1\right] g_{i,t}(\theta) \right], \tag{10}$$

where $g_{i,t}(\theta)$ denotes the token-level policy gradient, as described in Eq. 4. The proposed trajectory percentile scheduling manages policy improvements by accumulating policy gradients over targeted tokens and evolves optimization along trajectory, which initially focus on reliable downstream continuations and gradually calibrates earlier behaviors once the policy has been partially stabilized.

4 Experiments

4.1 Experimental Setup

Train data and benchmarks. We train our models on a challenging 30K dataset curated from OpenMathReasoning [Moshkov et al., 2025] and DeepMath-103K [He et al., 2025], and processed through deduplication and difficulty filtering. For each problem, we use DeepSeek-V3.2-Speciale [Liu et al., 2025] to synthesize three reasoning trajectories, and only those correctly verified at least one time are retained. We evaluate on in-distributional mathematical benchmarks: AIME24/25 [1], HMMT25 [Balunović et al., 2025], Minerva [Lewkowycz et al., 2022], and OlympiadBench [He et al., 2024], and out-of-distributional reasoning benchmarks, including Winogrande [Sakaguchi et al., 2021] for commonsense reasoning, GPQA-Diamond [Rein et al., 2023] for STEM reasoning, and MuSR [Sprague et al., 2023] for multistep soft reasoning. We use a rollout temperature of 0.6, and top-p sampling with p=0.95, under a maximum response length of 32768. We report the Avg@16 metric for AIME24/25 and HMMT25, and report the Avg@4 metric for other benchmarks.

Models and baselines. We perform our main experiments on Qwen3-4B and Qwen3-8B [Yang et al., 2025a]. We compare our method primarily against vanilla GRPO [Shao et al., 2024], and advanced token-level optimization baselines, including Entropy-Adv. [Cheng et al., 2025] that performs entropy-based advantage reweighting, and Forking-Tok. [Wang et al., 2025] that only optimizes a subset of forking tokens, selected by entropy proxy for policy gradients accumulation. Temporally scheduling the corresponding continuous and discrete credit allocation criteria are performed for comparison. TP-Schedule represents the trajectory percentile schedule on basic RL algorithms without other advanced credit allocation. We additionally validate our schedule on different RLVR algorithms including GSPO [Zheng et al., 2025], which underscores sequence-level optimization.

[1] https://huggingface.co/datasets/AI-MO/aimo-validation-aime

Table 1: Performance across mathematical and general reasoning benchmarks. **Bold** indicates the best results among the comparison. " ↪Schedule " represents applying the temporal schedule to corresponding credit allocation methods, while " TP " denotes the trajectory percentile.

Model	In-Distribution					Out-of-Distribution			
	AIME24	AIME25	HMMT25	Minerva	**Avg.**	WinoG*	GPQA*	MuSR	**Avg.**
Qwen3-4B	71.2	62.9	40.0	79.8	63.5	77.7	54.1	61.4	64.4
GRPO	73.1	65.3	41.7	80.9	65.3	76.8	55.6	60.9	64.4
TP-Schedule	**74.2**+1.1	**67.1**+1.8	**44.8**+3.1	**83.8**+2.9	**67.5**+2.2	**78.4**+1.6	**60.1**+4.5	**62.9**+2.0	**67.1**+2.7
GSPO	71.5	65.6	43.3	79.8	65.0	76.6	54.5	61.5	64.2
TP-Schedule	72.3+0.8	66.8+1.2	44.0+0.7	80.2+0.4	65.8+0.8	78.1+1.5	57.1+2.6	62.4+0.9	65.9+1.7
Entropy Adv.	73.8	65.8	41.9	80.1	65.4	77.8	54.5	61.7	64.6
↪ Schedule	**74.4**+0.6	**66.7**+0.9	42.4+0.5	80.1+0.0	65.9+0.5	**78.5**+0.7	55.6+1.1	62.2+0.5	65.4+0.8
Forking Tok.	71.5	64.8	41.5	81.3	64.8	77.6	55.6	61.5	64.9
↪ Schedule	74.2+2.7	65.8+1.0	**42.7**+1.2	**82.1**+0.9	**66.2**+1.4	78.2+0.6	**56.3**+0.7	**63.2**+1.7	**65.9**+1.0
Qwen3-8B	74.8	66.3	41.8	82.3	66.3	81.3	59.6	62.6	67.9
GRPO	75.6	68.6	42.7	83.6	68.0	82.0	62.7	63.9	69.5
TP-Schedule	**76.4**+0.8	**70.0**+1.4	**44.3**+1.6	**84.5**+0.9	**68.8**+0.8	**82.8**+0.8	**64.1**+1.4	**64.5**+0.6	**70.5**+1.0

Implementation details. We implement our methods based on the VERL [Sheng et al., 2024] framework. The maximum model length is set to 40960 tokens, with a maximum prompt length of 1024 and a maximum response length of 39936 to fully utilize the context window. We use the learning rate of 1×10^{-6} and the training batch size of 256. For each prompt, we sample 8 rollouts with a temperature of 1.2. Clipping thresholds are set to $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$, and we exclude both KL penalty and entropy regularization. We adopt the linear schedule as $\mathcal{S}(\cdot)$ in our main experiments, and apply temporal range $\tau_{low} = 0$ and $\tau_{high} = 0.8$, normalized by total training steps. Unless otherwise noted, we use the recommended implementation configuration and keep the training data, and evaluation settings consistent across methods for fair comparison.

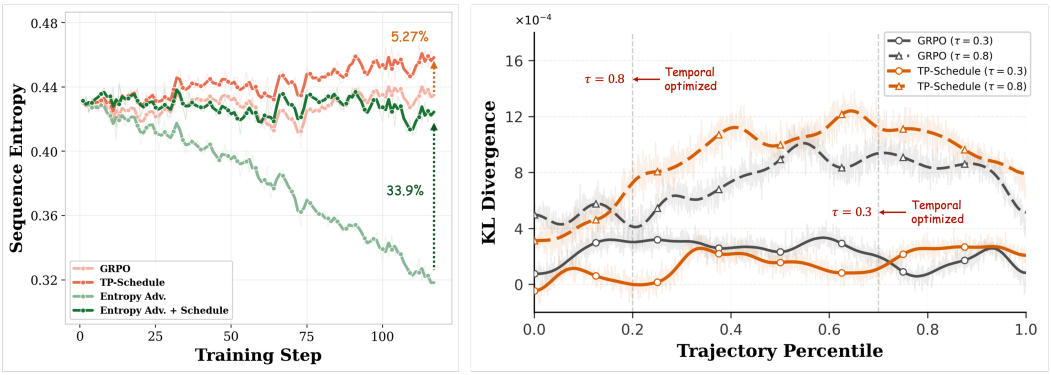
4.2 Main Results

Table 1 presents the evaluation results on in-distribution mathematical benchmarks and out-of-distribution multi-domain reasoning benchmarks, where the temporal scheduling show consistent improvement on both existing credit allocation methods, and proposed trajectory percentiles criteria. (1) On Qwen3-4B model, scheduling with trajectory percentile proxy over the course of training, i.e., TP-Schedule, outperforms the vanilla GRPO by 2.2% on mathematical and 2.7% on general reasoning benchmarks, respectively. While the improvements on Qwen3-8B further indicate that the trajectory percentile serves as a simple but effective criteria in working with temporal schedule, with overall 1% improvements over vanilla GRPO. (2) Additionally, for existing stagnant credit allocation methods, scheduling the corresponding allocation criteria, such as continuous advantage reweighting factor in Entropy Adv., and discrete threshold of selecting optimized tokens in Forking Tok., both show that temporal dimension provides a promising optimization direction for policy evolution. (3) We also validate the temporal scheduling on other RLVR algorithms, such as GSPO, where the performance show consistent improvement.

Applicability to multi-turn agentic tasks. To assess applicability beyond single-turn mathematical reasoning, we further evaluate Qwen3-4B on ALFWorld [Shridhar et al., 2020] and WebShop [Yao et al., 2022], which involve multi-turn interactive trajectories. We use a maximum of 20 interaction steps for ALFWorld and 15 steps for WebShop. The maximum response length is set to 4096 tokens per interaction. We apply temporal scheduling only to response tokens, starting from the final interaction step and progressively scheduling toward earlier tokens along the trajectory. We normalize token positions through total response length as trajectory percentile for scheduling. While temporal scheduling with trajectory percentile proxy shows consistent improvement, demonstrating its broader applicability and competitive on multi-turn interactive trajectories.

Table 2: Performance on multi-turn agentic tasks with Qwen3-4B. SR denotes success rate (%).

Method	ALFWorld		WebShop	
	SR (Seen)	SR (Unseen)	Score	SR
GRPO	80.3	73.5	66.8	48.2
TP-Schedule	82.9	74.8	69.5	51.0



(a) Train-Time Entropy Dynamics **(b) KL Divergence on Trajectory Percentiles**

Figure 3: Analysis of policy optimization under temporal scheduling. (a) Temporal scheduling consistently preserves higher policy entropy over full response under different credit allocation strategies. (b) Temporal scheduling over trajectory induces a structured shift in KL divergence from later to earlier percentiles over training, suggesting the non-uniformly optimized policy behaviors.

4.3 Temporal Scheduling Analysis

Beyond aggregate performance gains, we observe several characteristic phenomena that shed light on why temporal scheduling is effective in RLVR. By jointly analyzing training dynamics and trajectory-level behaviors, we identify two key factors that drive its improvements: the consistently reserved entropy compared to standard baselines, and the progressive alignment of policy behaviors.

Entropy preservation under temporal scheduling. We first examine the training-time entropy dynamics in Figure 3 (a), where the temporal scheduling and corresponding baselines are evaluated under different credit allocation strategies. Compared to standard GRPO, which optimized with all sampled tokens, temporal scheduling along trajectory percentiles consistently preserves higher policy entropy on full sequence. This effect is particularly pronounced when compared with entropy-based advantage reweighting, where entropy collapses rapidly as optimization aggressively concentrates on high entropy tokens, while temporally scheduling results more steady optimization and preserves policy entropy over 33.9%, and accompanied with higher performance in Table 1.

Temporal alignment of policy behaviors along the trajectory. We further analyze the token-level KL divergence normalized across trajectory percentiles in Figure 3 (b), which reveals how policy updates are distributed along the trajectory. We compute the KL divergence between the model checkpoints at $\bar{\tau} = 0.3$ and $\bar{\tau} = 0.8$ of the scheduled progress in training with the initial model, where temporal scheduling induces a structured and progressive shift in KL divergence. Early in training, larger deviations are concentrated on later trajectory percentiles compared to standard GRPO, and as training proceeds, the deviation gradually propagates toward earlier percentiles, suggesting that the policy behaviors are not updated uniformly, but instead optimized temporally for accommodation.

4.4 Ablations

To validate the reliability of temporal scheduling, we conduct ablations on Qwen3-4B model, including the sensitivity of schedule choice $\mathcal{S}(\cdot)$, the recommended temporal range to apply scheduling, and the allocation criteria worked with scheduling.

The choice of schedule $\mathcal{S}(\cdot)$ has minor effect on performance. We present the results of different scheduling functions applied to attenuate the credit allocation criteria in Table 3, and report Avg@16 performance on AIME25 and HMMT25, and Avg@4 performance on GPQA-Diamond, where the trajectory percentile is adopted as default allocation criteria. As described in Appendix A.2, the sigmoid features more optimization steps at the beginning and end of scheduling, while the gamma stresses more on early stage. We show that the temporal scheduling is insensitive to specific schedule choice, and show consistent improvement over basic GRPO, denoted as $\varnothing$. We adopt the linear schedule as the default setting.

Table 3: Ablations results of schedule choice.

$\mathcal{S}(\cdot)$	AIME25	HMMT25	GPQA-d
$\varnothing$	65.3	41.7	55.6
Sigmoid	66.7	43.5	58.5
Gamma	66.4	43.7	57.5
Linear	67.1	44.8	60.1

Temporal scheduling is more effective when extended to the later stages of training. In Table 4, we show results of different temporal range applied to scheduling the policy optimization. We remark that temporal scheduling starts from the strictest allocation criterion at $\tau_{low} = 0$, where only τ_{high} is ablated to determine the overall schedule range. The $\tau_{high} = 0$ reports the unscheduled GRPO baseline. We notice that extending scheduling to the later stages of training reveals the benefits of the temporal dimension, which allows more sufficient orchestration of policy optimization for accommodate distinct policy behaviors. We recommend $\tau_{high} \in [0.6, 0.8]$ as a performance sweet spot, where the allocation criteria over the course of training are well balanced, enabling effective policy optimization.

Table 4: Ablations results of applied temporal range.

τ_{high}	0.0	0.2	0.4	0.6	0.8	0.95
AIME25	65.3	65.4	65.4	66.7	67.1	65.6
HMMT25	41.7	39.8	40.8	43.3	44.8	42.6
GPQA-d	55.6	53.5	54.1	56.6	60.1	55.5

Percentile normalization improves robustness to response length variations. We show that TP-Schedule with percentile normalization accommodates trajectories with substantially different response lengths, and stably captures the coherent policy behaviors. Table 5 shows that when we replace normalized token positions with absolute token budget in TP-Schedule, the performance drops substantially, highlighting the effect of length normalization when scheduling token optimization with trajectory proxy. Moreover, we analyze answer convergence behavior as example to examine whether trajectory percentile captures stable policy behaviors across different response lengths. Specifically, we use the position of the `</think>` token as an indicator for answer convergence on trajectories with total response length across 4k, 16k, and 32k tokens, and observe that longer responses typically contain longer answer convergence segments in absolute token counts, while the normalized trajectory percentile remain stable and consistently capture the answer convergence at the final trajectory region, ranging from 0.912 to 0.963 across 4k to 32k response length. These observations support trajectory percentile as a more consistent proxy for aligning policy behavior compared to absolute token count.

Table 5: Effect of Percentile normalization.

	AIME25	HMMT25	GPQA-d
w/o Percentile	63.8	41.3	53.5
w/ Percentile	67.1	44.8	60.1

Table 6: Answer convergence across response lengths.

Metric	4k	16k	32k
Percentile	0.912 ± 0.025	0.947 ± 0.019	0.963 ± 0.011
Post-tokens	378.05 ± 127.95	955.81 ± 254.47	1204.4 ± 295.54

Trajectory percentile works effectively with temporal scheduling. In Figure 4, we show that temporal scheduling exhibits improved performance on reasonable credit allocation proxies, such as entropy and suffix, where the policy behaviors can be effectively accommodated for optimization, while degrading performance on others. In particular, randomly selecting a subset of tokens for scheduling policy optimization significantly corrupts the policy gradients, where the underlying policy behaviors are disorganized for temporal evolution. Additionally, trajectory percentile works effectively with temporal scheduling, supported by its natural perspective in distinguishing policy behaviors. However, the temporal order matters, while scheduling the optimization from earlier to later tokens (i.e., Prefix), critically depends on unreliable continuation, resulting in unstable optimization. The entropy proxy shows stable improvements when optimized from high entropy tokens, and gradually extends toward all sampled tokens, exhibiting improved performance compared to its stagnant counterpart (i.e., Forking Tok.). Detailed definitions are provided in Appendix A.3.

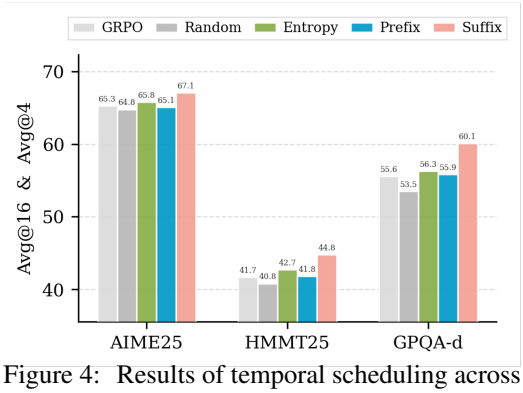


Figure 4: Results of temporal scheduling across different credit allocation proxies.

5 Related Work

Curriculum learning and staged optimization. The key distinction between temporal scheduling and existing curriculum approaches is the object being scheduled. Curriculum learning typically organizes training data from easier to harder, including reformulating data formats [Chen et al.,

2026], constructing verifiable subproblems [Jiang et al., 2026], or sampling prompts over task space [McKenzie et al., 2026]. Staged optimization instead schedule global training configurations across phases, including datasets, objectives, and hyperparameters [Hu et al., 2026]. The most related works are token selection, which typically rely on a static criterion to emphasize or mask selected tokens throughout training, including high-entropy token optimization [Wang et al., 2025], discarding spurious tokens through batch entropy for training stabilization [Liu et al., 2026], or restricting optimization to fixed top-k token space [Pang et al., 2026], where basically no object has been scheduled. Temporal scheduling instead schedules the token selection criterion throughout training, including continuous advantage reweighting criterion or discrete sparse token optimization criterion.

Token-selective optimization. Reinforcement learning with verification rewards faces a fundamental credit assignment challenge, motivating process-level feedback [Zou et al., 2025] and token-level advantage shaping through policy proxies such as entropy and future KL [Cheng et al., 2025, Ma et al., 2026]. Additionally, token-level analyses reveal that the effective learning signals are highly non-uniform, evidenced by divergence between the base and RL policies [Huang et al., 2026, Meng et al., 2026], token-ranking shifts [Huan et al., 2025, Chen et al., 2025], and cross-sampling performance [Meng et al., 2026]. Steering RLVR optimization with minority high-entropy tokens [Wang et al., 2025] or divergence-weighted token-level advantages [Meng et al., 2026, Yang et al., 2025b] induces superior performance over uniform full-token optimization. And recent works further extend token-level optimization through teacher-student divergence with on-policy distillation [Kim and Baek, 2026, Xu et al., 2026]. These methods primarily investigate where the learning signal lies. Although token scores and selected subsets may evolve with the policy, the underlying selection criterion remains unchanged, which consistently favoring the same type of proxy-defined behavior, and is principally different from temporal scheduling.

6 Conclusion

In this work, we revisit RLVR optimization from a temporal perspective, and show that prevalent credit allocation methods are principally suffered from stagnant allocation criteria throughout training, suggesting that *when* learning signals are scheduled can be as important as *where* they are allocated across tokens. Building upon this, we introduce temporal scheduling and identify trajectory percentiles as a simple yet effective proxy for evolving policy optimization. We further generalize the framework to common credit allocation paradigms, including continuous advantage reweighting, and discrete sparse token optimization. Temporal scheduling starts from targeted optimization induced by the corresponding allocation criterion and progressively relaxes toward general optimization over the course of training. We show that the sacrificed policy entropy is largely reserved by temporally scheduling the policy optimization. Experiments across model scales and RLVR algorithms demonstrate consistent improvements, highlighting temporal scheduling as a promising dimension for policy optimization. While current experiments identify a practical scheduling range, an important future direction is to leverage runtime feedback from policy optimization to adaptively determine the scheduling dynamics, thereby further strengthening the temporal curriculum effect.

Acknowledgments and Disclosure of Funding

This work was conducted during the first author's research internship at JD.com. We thank JD.com for providing computational resources and research support. We also acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC. No additional external funding was received for this work. The authors declare no other competing interests.

References

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2506.18896*, 2025.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, et al. Genprm: Scaling test-time compute of process reward models via generative reasoning. *arXiv preprint arXiv:2504.00891*, 2025.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint arXiv:2507.16806*, 2025.

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

Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. Fipo: Eliciting deep reasoning with future-kl influenced policy optimization. *arXiv preprint arXiv:2603.19835*, 2026.

Kexin Huang, Haoming Meng, Junkang Wu, Jinda Lu, Chiyu Ma, Ziqian Chen, Xue Wang, Bolin Ding, Jiancan Wu, Xiang Wang, et al. On the direction of rlvr updates for llm reasoning: Identification and exploitation. *arXiv preprint arXiv:2603.22117*, 2026.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

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

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

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *arXiv preprint arXiv:2310.16049*, 2023.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

Justin Chih-Yao Chen, Archiki Prasad, Zaid Khan, Joykirat Singh, Runchu Tian, Elias Stengel-Eskin, and Mohit Bansal. Cog-drift: Exploration on adaptively reformulated instances enables learning from hard reasoning problems. *arXiv preprint arXiv:2604.04767*, 2026.

Xitai Jiang, Zihan Tang, Wenze Lin, Yang Yue, Shenzhi Wang, and Gao Huang. From reasoning chains to verifiable subproblems: Curriculum reinforcement learning enables credit assignment for llm reasoning. *arXiv preprint arXiv:2605.22074*, 2026.

Darrien McKenzie, Nicklas Hansen, and Xiaolong Wang. Manifold bandits: Bayesian curriculum learning over the latent geometry of large language models. *arXiv preprint arXiv:2606.19750*, 2026.

Hanxu Hu, Yuxuan Wang, Maggie Huan, Jannis Vamvas, Yinya Huang, Zhijiang Guo, and Rico Sennrich. Dereason: A difficulty-aware curriculum improves decoupled sft-then-rl training for general reasoning. *arXiv preprint arXiv:2603.11193*, 2026.

Shiqi Liu, Zeyu He, Guojian Zhan, Letian Tao, Zhilong Zheng, Jiang Wu, Yinuo Wang, Yang Guan, Kehua Sheng, Bo Zhang, et al. Stapo: Stabilizing reinforcement learning for llms by silencing rare spurious tokens. *arXiv preprint arXiv:2602.15620*, 2026.

Jing-Cheng Pang, Liang Lu, Xian Tang, Kun Jiang, Sijie Wu, Kai Zhang, and Xubin Li. Reinforcement learning with promising tokens for large language models. *arXiv preprint arXiv:2602.03195*, 2026.

Haoming Meng, Kexin Huang, Shaohang Wei, Chiyu Ma, Shuo Yang, Xue Wang, Guoyin Wang, Bolin Ding, and Jingren Zhou. Sparse but critical: A token-level analysis of distributional shifts in rlvr fine-tuning of llms. *arXiv preprint arXiv:2603.22446*, 2026.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

Xingwu Chen, Tianle Li, and Difan Zou. Reshaping reasoning in llms: A theoretical analysis of rl training dynamics through pattern selection. *arXiv preprint arXiv:2506.04695*, 2025.

Minsang Kim and Seung Jun Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation. *arXiv preprint arXiv:2603.13260*, 2026.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026.

A Implementation Details

A.1 Training Configurations

We detail the training configurations and temporal scheduling hyperparameters employed throughout experiments in Table 7. We conduct experiments on 8 H200 GPUs, and uses AdamW optimizer with weight decay of 0.01. The learning rate follows a constant schedule. Gradient clipping is applied with maximum gradient norm set to 1.0. During training, we use 2048 sampled trajectories for policy optimization at each update step, and perform on-policy rollout generation throughout training.

For temporal scheduling, we adopt the linear schedule as the default setting in the main experiments, while additionally exploring sigmoid and gamma scheduling variants. We use trajectory percentile as the default credit allocation proxy and apply scheduling over the normalized training range $(\tau_{low}, \tau_{high}) = (0, 0.8)$, starting from the strictest allocation criterion for gradual attenuation.

Table 7: Training hyperparameters and temporal scheduling configurations used in our experiments, where underline denotes the default settings with variants for presentation

Hyperparameter	**Temporal Scheduling**
Shared Optimization Settings	
Base Model	Qwen3-4B / Qwen3-8B
Global Batch Size	256
Group Rollout Size	8
Optimizer	AdamW
Learning Rate	1×10^{-6}
LR Scheduler	Constant
Weight Decay	0.01
Gradient Clipping	1.0
Temperature	1.2
Max Prompt Length	1,024
Max Response Length	39,936
Policy Clip Ratio	[0.2, 0.28]
Temporal Scheduling Settings	
Scheduling Function $\mathcal{S}(\cdot)$	Linear / Sigmoid / Gamma
Sigmoid Schedule Center (β, k)	(0.5, 8)
Gamma Schedule Factor (γ)	2.0
Temporal Range (τ_{low}, τ_{high})	(0, 0.8)
Credit Allocation Criteria	Entropy / Trajectory Percentile

A.2 Scheduling Functions.

We explore the following scheduling functions $\mathcal{S}(\cdot)$ for progressively expanding optimization throughout training. All scheduling functions are monotonically decreasing with respect to the scheduled progress $\bar{\tau} \in [0, 1]$ defined in Eq. 6, and subject to $\mathcal{S}(\bar{\tau}) \in [0, 1]$. The linear and gamma schedules satisfy $\mathcal{S}(0) = 1$ and $\mathcal{S}(1) = 0$; the sigmoid schedule approximates these endpoints. Larger scheduling values correspond to stricter optimization selection during early training, while smaller values gradually expand optimization toward broader sampled tokens in later stages.

- LINEAR SCHEDULE: $\mathcal{S}(\bar{\tau}) = 1 - \bar{\tau}$. The optimization scope increases linearly along training progress.
- SIGMOID SCHEDULE: $\mathcal{S}_{\text{sigmoid}}(\bar{\tau}) = \frac{1}{1+\exp\left(k(\bar{\tau}-\beta)\right)}$. The optimization expansion remains conservative during early training and later training, and accelerates near the middle stage. We adopt $k = 8$ to control the sharpness of the transition, while $\beta = 0.5$ to specify the transition center along the training progress.

- GAMMA SCHEDULE: $\mathcal{S}(\bar{\tau}) = (1-\bar{\tau})^{\gamma}$, with $\gamma > 0$. Larger γ lowers the selection threshold more quickly and includes more tokens early in training, while smaller γ delays this expansion. We use $\gamma = 2$ in our experiments.

A.3 Credit Allocation Proxies

Temporal scheduling operates on token-level credit allocation proxies that determine the contribution of each sampled tokens for policy optimization. Given sampled trajectory $\mathbf{y} = \{y_1, \ldots, y_T\}$, each token y_t is associated with corresponding derived proxy μ_t, and we primarily focus on discrete case described in Sec 3.2, where the scheduled allocation factor is generally formulated as:

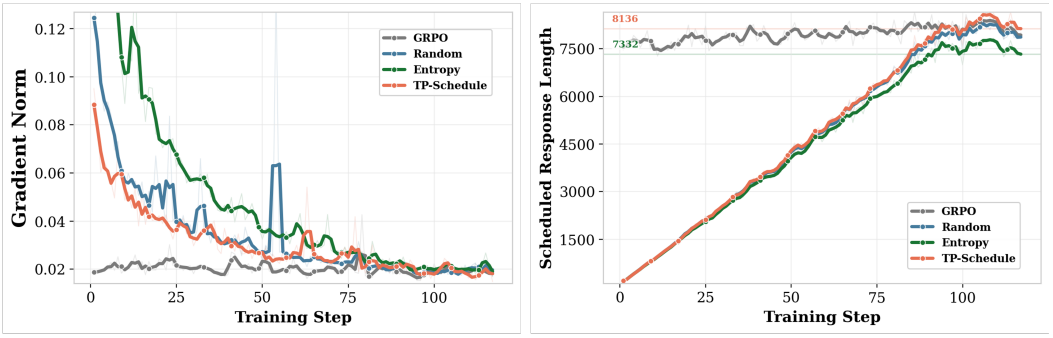
$$\widehat{f}_{t(\tau)} = \mathbb{I}\big[\mu_t \geq \mathcal{S}(\bar{\tau}) \cdot \epsilon\big],$$

where $\mathcal{S}(\bar{\tau})$ denotes the temporal scheduling function and ϵ controls the optimization boundary.

- TRAJECTORY PERCENTILE PROXY: $\mu_t = \frac{t}{T} \in (0, 1]$. Larger μ_t corresponds to tokens located closer to the end of the trajectory. The scheduled allocation factor is defined as $\widehat{f}_{t(\tau)} = \mathbb{I}[\mu_t \geq \mathcal{S}(\bar{\tau})]$ for suffix optimization, which progressively extends optimization from later toward earlier tokens throughout training, and $\widehat{f}_{t(\tau)} = \mathbb{I}[\mu_t \leq \mathcal{S}(1-\bar{\tau})]$ for prefix optimization, as described in Figure 4.
- ENTROPY-BASED PROXY: $\mu_t = \mathcal{H}(\pi_\theta(\cdot \mid y_{<t}))$. The scheduled allocation factor is defined as $\widehat{f}_{t(\tau)} = \mathbb{I}[\mu_t \geq \mathcal{S}(\bar{\tau}) \cdot \epsilon]$, where typically $\epsilon = 0.2$ without scheduling [Wang et al., 2025], and we set $\epsilon = 1$ for temporal scheduling. Optimization is initialized from most high-entropy tokens and progressively expanded toward broader sampled tokens during training.
- RANDOM PROXY: $\mu_t \sim \text{Bernoulli}(\bar{\tau})$, where the scheduled progress $\bar{\tau}$ controls the selected probability of sampled tokens. The scheduled allocation factor is defined as $\widehat{f}_{t(\tau)} = \mathbb{I}[\mu_t \geq \mathcal{S}(\bar{\tau})]$. Unlike structured scheduling criteria, random token selection does not preserve coherent policy behaviors along the trajectory and typically results in unstable optimization.

B Further Analysis on Temporal Scheduling

We further analyze the optimization dynamics induced by temporal scheduling through gradient norm and scheduled response length statistics, as shown in Figure 5.



(a) Train-Time Gradient Norm **(b) Scheduled Response Length**

Figure 5: Training dynamics under temporal scheduling. (a) Train-time gradient norm. Temporal scheduling consistently yields larger and more concentrated optimization signals compared to standard GRPO, while TP-Schedule exhibits the most stable optimization trajectory. Random scheduling introduces unstable gradient spikes, suggesting that incoherent token organization corrupts the accumulated policy gradients. (b) Scheduled response length during optimization. Temporal scheduling progressively expands the optimization from specified token subsets toward broader trajectory, where TP-Schedule achieves slightly higher response length compared to entropy-based schedule.

Gradient norm dynamics. We first examine the train-time gradient norm in Figure 5 (a). Compared to standard GRPO, temporal scheduling methods exhibit significantly larger gradient norms during the early training stage, indicating that optimization is concentrated on more informative token subsets

rather than uniformly averaged across all sampled tokens. Among different scheduling strategies, TP-Schedule shows the most stable optimization trajectory, where the gradient norm gradually decreases as optimization is progressively expanded toward broader trajectory positions. While entropy-based scheduling maintains relatively strong optimization signals during early training, with higher gradient norm compared to TP-Schedule, consistent with its focus on high-uncertainty policy behaviors. In contrast, random scheduling produces unstable optimization dynamics with significant gradient spikes during training, suggesting that randomly organizing token participation disrupts coherent policy behaviors and corrupts the accumulated policy gradients. Meanwhile, standard GRPO maintains consistently small gradient norms due to uniformly averaging optimization over all sampled tokens, potentially diluting informative learning signals.

Scheduled response length dynamics. We further analyze the scheduled response length in Figure 5 (b), which measures the average number of trajectory tokens participating in optimization throughout training, and performed under a linear schedule. All temporal scheduling methods progressively increase the optimization scope during training, gradually extending optimization from specified token subsets toward broader trajectory regions. Among them, TP-Schedule shows the largest final response length, indicating that trajectory percentile provides a coherent criterion for accommodating policy behaviors in optimization with less sacrificed policy entropy. Entropy scheduling shows comparatively slower expansion and shorter scheduled response lengths, suggesting that policy-entropy remain sacrificed during the early training. Interestingly, although temporal scheduling initially constrains optimization to partial trajectory regions, all scheduling variants naturally approach the response length of standard GRPO during later training stages, suggesting that temporal scheduling does not fundamentally alter the final response length distribution, but instead progressively reorganizes how policy behaviors are incorporated throughout optimization.

C Case Study

We further present qualitative case studies to analyze how temporal scheduling affects the reasoning behaviors learned during RLVR optimization.

Question and Ground Truth

Question: The parabola with equation $y = x^2 - 4$ is rotated $60°$ counterclockwise around the origin. The unique point in the fourth quadrant where the original parabola and its image intersect has y-coordinate $\frac{a-\sqrt{b}}{c}$, where a, b, and c are positive integers, and a and c are relatively prime. Find $a + b + c$.
Ground Truth: 62

Response from Baseline

To solve the problem, we are given a parabola defined by the equation:

$$y = x^2 - 4$$

and we are to rotate this parabola **60° counterclockwise** around the origin. The task is to find the y-coordinate of the unique point in the fourth quadrant where the original parabola and its rotated image intersect, expressed in the form:

$$\frac{a - \sqrt{b}}{c}$$

where a, b, c are positive integers, and a and c are relatively prime. Then, we are to find $a + b + c$.

Step 1: Rotation of a Point
A point (x, y) rotated counterclockwise by θ becomes:

$$X = x\cos\theta - y\sin\theta$$

$$Y = x\sin\theta + y\cos\theta$$

Response from Baseline

For $\theta = 60°$, we have:

$$\cos 60° = \frac{1}{2}, \qquad \sin 60° = \frac{\sqrt{3}}{2}$$

So the rotation becomes:

$$X = \frac{1}{2}x - \frac{\sqrt{3}}{2}y \qquad Y = \frac{\sqrt{3}}{2}x + \frac{1}{2}y$$

Step 2: Expressing the Rotated Parabola
Since the original parabola is $y = x^2 - 4$, any point (x, y) on the original parabola satisfies this. When this point is rotated, the resulting point (X, Y) lies on the rotated parabola. However, we are interested in the intersection of the original and rotated parabola, so we want to find a point (x, y) such that:

- $y = x^2 - 4$ (original parabola), and
- $(X, Y) = \left(\frac{1}{2}x - \frac{\sqrt{3}}{2}y, \frac{\sqrt{3}}{2}x + \frac{1}{2}y\right)$ lies on the original parabola, i.e., $Y = X^2 - 4$

So, we substitute $y = x^2 - 4$ into the rotated point equations and set $Y = X^2 - 4$. This gives an equation in terms of x, which we solve to find the x-coordinates of the intersection points.
After substitution and simplification, the resulting equation is a quartic in x:

$$3x^4 - 2\sqrt{3}x^3 - 25x^2 + 6\sqrt{3}x + 40 = 0$$

Annotation: The intersection condition is formulated with the wrong rotation direction. For the counterclockwise rotated parabola, the correct quartic should be $3x^4 + 2\sqrt{3}x^3 - 25x^2 - 6\sqrt{3}x + 40 = 0$, where the signs of the cubic and linear terms are reversed.

Step 3: Solving the Quartic Equation
By substituting $z = x\sqrt{3}$, the equation becomes:

$$z^4 - 2z^3 - 25z^2 + 18z + 120 = 0$$

This factors as:

$$(z^2 + z - 10)(z^2 - 3z - 12) = 0$$

Solving these quadratics gives:

$$z = \frac{-1 \pm \sqrt{41}}{2} \qquad z = \frac{3 \pm \sqrt{57}}{2}$$

Then, since $z = x\sqrt{3}$, we get:

$$x = \frac{-1 \pm \sqrt{41}}{2\sqrt{3}}, \qquad x = \frac{3 \pm \sqrt{57}}{2\sqrt{3}}$$

Step 4: Identify the Point in the Fourth Quadrant
We are looking for the unique point in the fourth quadrant, which requires:

- $x > 0$
- $y < 0$

Let's test the values:

- For
$$x = \frac{-1 + \sqrt{41}}{2\sqrt{3}} \approx \frac{-1 + 6.4}{3.464} \approx 1.56,$$
positive, and
$$y = x^2 - 4 \approx 2.43 - 4 = -1.57,$$
negative $\rightarrow$ this is a valid point in the fourth quadrant.

But when we compute the y-coordinate of this point, it turns out to be: $y = \frac{-3-\sqrt{41}}{6}$.

Response from Baseline

This is not in the form $\frac{a-\sqrt{b}}{c}$, as the numerator is negative and contains a negative sign.

However, if we consider the clockwise rotation instead, the correct solution emerges:

- The correct point is

$$x = \frac{-3+\sqrt{57}}{2\sqrt{3}}$$

Annotation: This solution is introduced by switching to a clockwise rotation, which is inconsistent with the original problem statement specifying a counterclockwise rotation.

- Then,

$$y = x^2 - 4 = \frac{3-\sqrt{57}}{2}$$

This is in the form $\frac{a-\sqrt{b}}{c}$, where:

- $a = 3$
- $b = 57$
- $c = 2$

These are all positive integers, and $\gcd(3, 2) = 1$, so the condition is satisfied.

Final Step: Compute $a + b + c$

$$a + b + c = 3 + 57 + 2 = \boxed{62}$$

Correct Answer with Incorrect Reasoning Process

Response from Temporal Scheduling

To solve the problem, we are given that the parabola $y = x^2 - 4$ is rotated $60°$ counterclockwise about the origin, and we are to find the unique point in the fourth quadrant where the original and rotated parabolas intersect. The y-coordinate of this point is given in the form

$$\frac{a-\sqrt{b}}{c},$$

and we are to find $a + b + c$, where a and c are coprime.

Step 1: Understand the Rotation
A rotation of a point (x, y) by $60°$ counterclockwise around the origin is given by the rotation matrix:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos 60° & -\sin 60° \\ \sin 60° & \cos 60° \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

So, the rotated coordinates (X, Y) of a point (x, y) are:

$$X = \frac{1}{2}x - \frac{\sqrt{3}}{2}y, \qquad Y = \frac{\sqrt{3}}{2}x + \frac{1}{2}y$$

We are to find the equation of the rotated parabola, which is the image of the original parabola under this rotation.

Step 2: Find the Equation of the Rotated Parabola
We solve the system:

$$X = \frac{1}{2}x - \frac{\sqrt{3}}{2}y \qquad Y = \frac{\sqrt{3}}{2}x + \frac{1}{2}y$$

Solving for x and y in terms of X and Y, we get:

$$x = \frac{X + \sqrt{3}Y}{2}, \qquad y = \frac{Y - \sqrt{3}X}{2}$$

Response from Temporal Scheduling

Since the original parabola is $y = x^2 - 4$, substitute these expressions into it:

$$\frac{Y - \sqrt{3}X}{2} = \left(\frac{X + \sqrt{3}Y}{2}\right)^2 - 4$$

Multiply both sides by 4:

$$2(Y - \sqrt{3}X) = (X + \sqrt{3}Y)^2 - 16$$

Expanding the right-hand side:

$$(X + \sqrt{3}Y)^2 = X^2 + 2\sqrt{3}XY + 3Y^2$$

So the equation becomes:

$$2Y - 2\sqrt{3}X = X^2 + 2\sqrt{3}XY + 3Y^2 - 16$$

Bring all terms to one side:

$$X^2 + 2\sqrt{3}XY + 3Y^2 + 2\sqrt{3}X - 2Y - 16 = 0$$

This is the equation of the rotated parabola in the original coordinate system.

Step 3: Find the Intersection with the Original Parabola
We are to find the points of intersection between the original parabola $y = x^2 - 4$ and the rotated one. Substituting $y = x^2 - 4$ into the rotated parabola equation:

$$x^2 + 2\sqrt{3}x(x^2 - 4) + 3(x^2 - 4)^2 + 2\sqrt{3}x - 2(x^2 - 4) - 16 = 0$$

Simplify each term:

- $2\sqrt{3}x(x^2 - 4) = 2\sqrt{3}x^3 - 8\sqrt{3}x$
- $3(x^2 - 4)^2 = 3(x^4 - 8x^2 + 16) = 3x^4 - 24x^2 + 48$
- $-2(x^2 - 4) = -2x^2 + 8$

Putting it all together:

$$x^2 + 2\sqrt{3}x^3 - 8\sqrt{3}x + 3x^4 - 24x^2 + 48 + 2\sqrt{3}x - 2x^2 + 8 - 16 = 0$$

Combine like terms:

- x^4: $3x^4$
- x^3: $2\sqrt{3}x^3$
- x^2: $x^2 - 24x^2 - 2x^2 = -25x^2$
- x: $-8\sqrt{3}x + 2\sqrt{3}x = -6\sqrt{3}x$
- Constants: $48 + 8 - 16 = 40$

So the equation becomes:

$$3x^4 + 2\sqrt{3}x^3 - 25x^2 - 6\sqrt{3}x + 40 = 0$$

Step 4: Solve the Quartic Equation
This is a quartic equation with irrational coefficients. We can factor it using substitution or by trial. Let's use the substitution

$$x = \frac{\sqrt{3}t}{2},$$

which simplifies the equation.
After substitution and factoring, we find that the equation factors as:

$$(3t^2 + 3t - 4)(9t^2 - 3t - 10) = 0$$

Response from Temporal Scheduling

Solving each quadratic:

- $3t^2 + 3t - 4 = 0 \Rightarrow t = \frac{-3 \pm \sqrt{57}}{6}$
- $9t^2 - 3t - 10 = 0 \Rightarrow t = \frac{1 \pm \sqrt{41}}{6}$

Now, recall that

$$x = \frac{\sqrt{3}t}{2},$$

so:

- $x = \frac{\sqrt{3}(-3+\sqrt{57})}{12} = \frac{-\sqrt{3}+\sqrt{19}}{2}$
- $x = \frac{\sqrt{3}(1+\sqrt{41})}{12}$
- etc.

Only the first solution gives a point in the fourth quadrant, where $x > 0$ and $y < 0$. Let's take:

$$x = \frac{-\sqrt{3} + \sqrt{19}}{2}$$

Then, the y-coordinate is:

$$y = x^2 - 4 = \left(\frac{-\sqrt{3} + \sqrt{19}}{2}\right)^2 - 4$$

Simplify:

$$x^2 = \frac{19 + 3 - 2\sqrt{57}}{4} = \frac{22 - 2\sqrt{57}}{4} = \frac{11 - \sqrt{57}}{2}$$

$$y = \frac{11 - \sqrt{57}}{2} - 4 = \frac{11 - \sqrt{57} - 8}{2} = \frac{3 - \sqrt{57}}{2}$$

Step 5: Final Answer
The y-coordinate is:

$$\frac{3 - \sqrt{57}}{2}$$

So,

$$a = 3, \qquad b = 57, \qquad c = 2,$$

and:

$$a + b + c = 3 + 57 + 2 = \boxed{62}$$

Although standard GRPO is able to improve final-answer accuracy through outcome-level supervision, we observe that its reasoning process often fails to establish reliable intermediate reasoning behaviors. In cases, the generated response arrives at the correct answer while containing inconsistent, incomplete, or logically incorrect intermediate derivations. While temporal scheduling is more able to produces coherent reasoning trajectories, suggesting that the process-level reasoning quality is also implicitly promoted during training. We believe this phenomenon arises from the progressive organization of policy behaviors throughout optimization.

- Standard GRPO uniformly optimizes all sampled tokens under globally broadcast rewards, which may simultaneously mix heterogeneous behaviors, encouraging shortcuts that directly improve outcome rewards without consistently refining intermediate reasoning steps.
- Temporal scheduling leverages the progressive optimization, avoiding simultaneously balancing heterogeneous policy behaviors during early training, allowing intermediate reasoning structures to emerge more consistently, suggesting a promising dimension to complement existing credit allocation methods for reasoning process optimization under sparse outcome supervision.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations and future work are discussed in Sec 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The implementation details and experimental settings are provided in sec 4 and Appendix A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data are publicly available at `https://github.com/Jinghaoleven/RLVR-Schedule`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental settings and data details are provided in sec 4.1 and Appendix A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The Avg@16 metric and Avg@4 metric are reported for main experiments in Table 1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computation resources is provided in Appendix A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research in the paper conform the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Any released models and data follow the original licenses and usage restrictions of the underlying base models and data source.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The assets used in this paper are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets introduced in the paper are well documented in Sec 4.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLM is used only for writing, editing, or formatting purposes and does *not* impact the core method development in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.