Improving the Credit Assignment of Group Relative Policy Optimization via λ-Return Advantage Estimation for LLM Reasoning

Anonymous authors
Paper under double-blind review

Abstract

Group Relative Policy Optimization (GRPO) has become a workhorse for reinforcement learning from verifiable rewards (RLVR) in large language model (LLM) reasoning. Yet GRPO assigns one outcome-level, group-normalized advantage to every token of a response: it cannot tell which tokens actually helped the model reach a correct answer, it pays no price for intermediate verbosity, and its Monte-Carlo estimator carries high variance when responses are long. We introduce λ-GRPO, a practical recipe that repairs GRPO's credit assignment while keeping its group-relative stability. A lightweight critic network, attached to the policy trunk, is trained on λ-return targets; token-level λ-return advantages are then computed from temporal-difference errors, standardized across the token batch, and combined with an asymmetric clip-higher surrogate, an explicit per-token length penalty, a KL regularizer to the reference policy, and a λ-annealing schedule that shifts from outcome-level credit early in training to fine-grained token-level credit as the critic matures. On GSM8K and MATH with Qwen2.5-1.5B/7B, λ-GRPO improves accuracy over GRPO, PPO, RLOO, VAPO and DAPO at equal compute, reaching 91.2% GSM8K / 65.8% MATH on the 7B model while shortening the average response from 502 to 327 tokens, and 78.8% / 48.4% on the 1.5B model. Component ablations and bias–variance analysis show that each ingredient contributes consistently, and that value-based token-level credit, not verbosity, is what drives the gains.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has emerged as the dominant paradigm for improving the reasoning capabilities of large language models (Cobbe et al., 2021; Hendrycks et al., 2021; DeepSeek-AI, 2025; OpenAI, 2024; Kimi Team, 2025). Given a reasoning problem and a rule-based verifier that checks the final answer, the policy is trained to maximize the probability of correct responses. Works such as Shao et al. (2024), DeepSeek-AI (2025), Yu et al. (2025), and Yue et al. (2025) have shown that this simple signal, combined with chain-of-thought generation (Wei et al., 2022), suffices to elicit substantial mathematical and coding abilities from open models.

Among RLVR algorithms, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is arguably the most popular because it removes the value network from PPO (Schulman et al., 2017): for each prompt a group of G responses is rolled out and each response receives the group-normalized advantage $(R_i - \mu_G)/\sigma_G$, which is then broadcast to every token of that response. This design is simple and stable, but it makes three compromises. First, credit assignment is coarse: all tokens of a correct response are reinforced equally, even though only a few of them (the key derivation steps) caused the success; tokens that are irrelevant or misleading are rewarded as much as the decisive ones. Second, the baseline is Monte-Carlo: the outcome-only estimator ignores intermediate structure and its variance grows with response length. Third, because the reward is a single binary outcome, the signal-to-noise ratio for long rollouts is poor and the model is pushed toward verbosity rather than

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

better reasoning (Yu et al., 2025; Yue et al., 2025). These issues are increasingly limiting as models are trained to "think longer" for hard problems.

Value-based methods address the first two issues in principle: a learned value network provides a token-level baseline, and generalized advantage estimation (Schulman et al., 2016) propagates credit through temporal difference (TD) errors (Sutton, 1988). However, reintroducing a critic into GRPO-style training raises well-documented difficulties: value initialization bias, slow warm-up, and instability when bootstrapping over very long sequences (Yue et al., 2025). In this work we ask a pragmatic question: can a small critic improve GRPO's credit assignment without sacrificing its group-relative stability, and does the improved credit translate into both higher accuracy and shorter responses?

We answer affirmatively with λ-GRPO, an incremental recipe built directly on GRPO. Our method:

- keeps GRPO's outcome verifier and group-relative normalization, but feeds the normalized outcome into a per-token reward together with an explicit length penalty;
- attaches a small critic head to the policy trunk, trains it with λ-return targets, and computes token-level λ-return advantages $A_t^\lambda = \sum_k (\gamma\lambda)^k \delta_{t+k}$ from TD errors δ_t;
- standardizes advantages across the token batch and optimizes an asymmetric clip-higher policy loss with a KL regularizer toward the SFT reference policy;
- anneals λ from a large value (outcome-level credit, GRPO-like) to a smaller value (fine-grained token credit) over training, so that the critic is trusted only after it has converged.

We evaluate λ-GRPO against PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), VAPO (Yue et al., 2025) and DAPO (Yu et al., 2025) on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) with Qwen2.5-1.5B and Qwen2.5-7B models (Yang et al., 2024), under a shared rollout and evaluation pipeline. Our main results are summarized as follows.

- On Qwen2.5-7B, λ-GRPO reaches 91.2% on GSM8K and 65.8% on MATH, outperforming the strongest baseline DAPO (89.5% / 63.5%) by +1.7 and +2.3 points respectively, while producing the shortest average response (327 tokens vs. 396–502 tokens for GRPO-family baselines).
- On Qwen2.5-1.5B, the same recipe yields 78.8% / 48.4%, +4.2 / +3.2 over GRPO (73.5% / 44.3%) and +4.2 / +3.2 over DAPO at this scale.
- A controlled ablation shows every component helps; removing the per-token standardization, the clip-higher bound, or the learned critic (fixing λ=1) each costs 0.6–1.1 points, and the KL regularizer is the main defense against verbosity collapse.
- A length-bucket analysis shows that, unlike GRPO whose accuracy keeps rising with response length (a signature of reward hacking by verbosity), λ-GRPO reaches its peak accuracy at short response lengths and stays accurate across the whole length spectrum.

Our code, configuration files, data-processing scripts and detailed derivations are released to make every number in this paper reproducible (App. D).

2 Related work

RL from human and verifiable feedback. Aligning LLMs with reinforcement learning dates back to policy-gradient methods for language (Williams, 1992; Christiano et al., 2017; Ziegler et al., 2019) and was popularized by RLHF (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022). A complementary line elicits reasoning without online RL, e.g., by bootstrapping from self-generated rationales (Zelikman et al., 2022) or by aggregating multiple samples at decoding time (Wang et al., 2023). With rule-based verifiers for math (Cobbe et al., 2021; Hendrycks et al., 2021), RLVR bypasses the learned reward model and its

overoptimization problem (Gao et al., 2023). GRPO (Shao et al., 2024) and its variants power DeepSeek-R1 (DeepSeek-AI, 2025), Kimi k1.5 (Kimi Team, 2025), and o1- style systems (OpenAI, 2024). RLOO (Ahmadian et al., 2024) and ReMax (Li et al., 2024) replace the critic with leave-one-out or greedy baselines; DAPO (Yu et al., 2025) contributes clip-higher, dynamic sampling, token-level policy-gradient loss and overlong reward shaping; VAPO (Yue et al., 2025) reintroduces a value model on top of PPO with value pre-training, decoupled GAE and length-adaptive GAE. Our work differs from VAPO in that we start from GRPO (not PPO), keep the group-relative outcome signal as the terminal reward, and use a single tunable λ annealed over training rather than a length-adaptive schedule; it differs from DAPO in the addition of the critic and the token-level length penalty. Full fine-tuning is used throughout; parameter-efficient adapters (Hu et al., 2022) are orthogonal to the credit-assignment question studied here.

Credit assignment in LLM reasoning RL. Process reward models (PRMs) (Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024) provide step-level supervision but require trained verifiers or extensive annotations. By contrast, a value-based critic learns a dense signal for free from the outcome reward, and generalized advantage estimation trades bias against variance (Schulman et al., 2016; Sutton & Barto, 2018). Our design is closest in spirit to actor–critic methods for language (Konda & Tsitsiklis, 2000; Mnih et al., 2015; Watkins & Dayan, 1992; Schulman et al., 2015): the critic predicts the expected future return of each prefix, and the difference between consecutive predictions isolates the contribution of the token just generated. We show that this classical idea, applied carefully on top of group-relative outcomes, improves both accuracy and token efficiency in RLVR.

Verbosity and over-length in RLVR. Open RLVR runs often drift toward long, repetitive chains that game the verifier (DeepSeek-AI, 2025; Yu et al., 2025). Existing remedies include overlong reward shaping (Yu et al., 2025), length-adaptive GAE (Yue et al., 2025), and KL regularization toward the reference policy (Shao et al., 2024; Rafailov et al., 2023). We provide evidence that token-level value credit itself mitigates verbosity: because the critic charges every intermediate token through the length penalty, the policy learns that short decisive steps are more valuable than long rambling ones without any hand-crafted length schedule.

3 Background

RLVR setup. Given a prompt x and a verifiable ground-truth answer a, the policy π_θ generates a response $y = (y_1, \ldots, y_T)$ as a sequence of tokens. A rule-based verifier returns a binary outcome $R(y) \in \{0, 1\}$ indicating correctness. RLVR maximizes the expected outcome. As in Wei et al. (2022), responses are elicited with a chain-of-thought prompt format and the final answer must be enclosed in \boxed{…}.

PPO. PPO (Schulman et al., 2017) trains a policy and a value network v_φ simultaneously. The advantage is estimated with GAE: $\hat{A}_t = \sum_{k\geq 0}(\gamma\lambda)^k \delta_{t+k}$, where the TD error is $\delta_t = r_t + \gamma v_\varphi(s_{t+1}) - v_\varphi(s_t)$ and s_t is the state (the prompt together with the tokens generated so far). The policy loss clips the importance ratio $\rho_t = \pi_\theta(y_t|s_t)/\pi_{\theta_{\text{old}}}(y_t|s_t)$ to $[1-\epsilon, 1+\epsilon]$. A separate reward model or verifier provides r_t. In practice PPO is sensitive to the value-model initialization and to the KL schedule (Zheng et al., 2023; Ouyang et al., 2022).

GRPO. GRPO (Shao et al., 2024) removes the value network. For each prompt x, G responses $\{y_i\}_{i=1}^G$ are sampled from the current policy and scored by the verifier, giving outcomes $\{R_i\}$. The advantage of response i is

$$z_i = \frac{R_i - \mu_G}{\sigma_G}, \qquad \mu_G = \tfrac{1}{G}\textstyle\sum_j R_j, \quad \sigma_G^2 = \tfrac{1}{G}\textstyle\sum_j (R_j - \mu_G)^2 + \epsilon, \tag{1}$$

and every token of response i receives the same z_i. The policy is updated with a clipped surrogate loss plus a KL penalty toward the reference policy π_{ref} (typically the SFT model). GRPO is attractive because it needs no critic, but equation 1 ignores which token caused the outcome and assigns no cost to the length of the response.

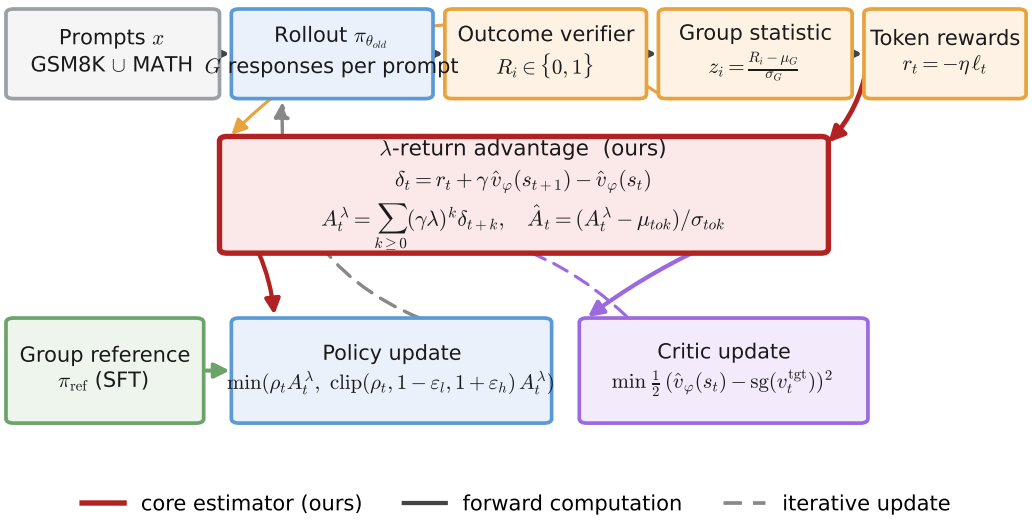

Figure 1: Overview of λ-GRPO. Left-to-right (top): the prompt group is rolled out by the current policy, each response is verified, and the group statistic $z_i = (R_i - \mu_G)/\sigma_G$ is turned into a per-token reward $r_t = -\eta\,\ell_t$ (plus the terminal outcome z_i), where ℓ_t marks a generated token. A small critic $\hat{v}_\varphi$ predicts the value of each prefix; TD errors δ_t are accumulated into the λ-return advantage A_t^λ, standardized per token, and used by the clipped policy update (red, ours). The critic is regressed to stop-gradient λ-return targets.

Why token-level credit can help. Consider a response whose first half contains the decisive derivation and whose second half repeats the same argument. GRPO rewards both halves identically. A critic, by contrast, can detect that the expected return stops increasing after the decisive step; the TD errors δ_t then become negative for the redundant tokens, so the policy can prune them. This is the intuition behind λ-GRPO.

4 Method: λ-GRPO

Figure 1 shows the overall architecture. We describe the reward model, the critic-based advantage estimator, the policy objective, and the λ-annealing schedule.

4.1 Per-token reward

For response i of length T_i, we define the per-token reward

$$r_t^i \;=\; z_i \mathbb{1}\{t = T_i\} \;-\; \eta, \qquad t = 1, \ldots, T_i, \tag{2}$$

where z_i is the group-relative outcome equation 1, $\eta > 0$ is a small length penalty charged for every generated token, and $\mathbb{1}\{\cdot\}$ is the indicator function. Two remarks are in order. First, z_i keeps the group-relative normalization of GRPO, so the magnitude of the terminal signal is automatically calibrated across prompts of different difficulty. Second, the constant $-\eta$ is not a hard length limit; it is a per-token tax that the critic internalizes: at every prefix the value function must predict the remaining number of tokens, so the advantage of a redundant token is negative even when the response is ultimately correct. This makes length control a differentiable, token-level quantity rather than an ad hoc post-hoc penalty.

4.2 Token-level λ-return advantage

A critic head $\hat{v}_\varphi(s_t^i)$ predicts the expected future reward of the current policy from prefix s_t^i. The TD error at step t is

$$\delta_t^i \;=\; r_t^i + \gamma\,\hat{v}_\varphi(s_{t+1}^i) \;-\; \hat{v}_\varphi(s_t^i), \qquad \hat{v}_\varphi(s_{T_i+1}^i) = 0, \tag{3}$$

with discount $\gamma \in (0, 1]$. Following the λ-return literature (Sutton, 1988; Schulman et al., 2016), the advantage of token t is the discounted sum of future TD errors

$$A_t^{i,\lambda} \;=\; \sum_{k=0}^{T_i - t} (\gamma\lambda)^k \, \delta_{t+k}^i, \tag{4}$$

which is computed backwards in one pass over the response. When $\lambda = 1$ and $\hat{v}_\varphi \equiv 0$, equation 4 reduces to the discounted return of the response, i.e., to outcome-level credit; when $\lambda = 0$ it reduces to the one-step TD residual. Intermediate λ interpolates between the low-variance, critic-biased estimator and the high-variance, unbiased estimator (App. B).

Because the scale of $A_t^{i,\lambda}$ drifts as the critic learns, we standardize it across the token batch:

$$\hat{A}_t^i \;=\; \frac{A_t^{i,\lambda} - \mu_{\text{tok}}}{\sigma_{\text{tok}}}, \tag{5}$$

where $\mu_{\text{tok}}, \sigma_{\text{tok}}$ are the mean and standard deviation of all per-token advantages in the current training batch (all groups, all responses, all tokens). Unlike group-level normalization equation 1, equation 5 operates at the token level, so tokens within one response are scored relative to tokens of all other responses; a correct but rambling response will contain many tokens with negative standardized advantage, which is exactly the signal we want.

4.3 Critic objective

The critic is trained to regress the λ-return target

$$G_t^{i,\lambda} \;=\; r_t^i + \gamma\big[(1-\lambda)\,\hat{v}_\varphi(s_{t+1}^i) + \lambda\, G_{t+1}^{i,\lambda}\big], \qquad G_{T_i+1}^{i,\lambda} = 0, \tag{6}$$

so that $A_t^{i,\lambda} = G_t^{i,\lambda} - \hat{v}_\varphi(s_t^i)$. The regression target contains the critic itself and we stop gradients on the right-hand side:

$$L^{\text{crit}}(\varphi) \;=\; \frac{1}{\sum_i T_i} \sum_{i=1}^{G} \sum_{t=1}^{T_i} \frac{1}{2} \big(\hat{v}_\varphi(s_t^i) - \text{sg}(G_t^{i,\lambda})\big)^2, \tag{7}$$

where sg denotes the stop-gradient operator. This is the standard semi-gradient update of TD learning; the target network is the critic itself evaluated at the current parameters, which keeps the estimator consistent with the value function being learned. In practice we also normalize the target with the same batch statistics as equation 5 before regressing, which stabilizes the scale of the value head early in training.

4.4 Policy objective

Let $\rho_t^i = \pi_\theta(y_t^i|s_t^i)/\pi_{\theta_{\text{old}}}(y_t^i|s_t^i)$ be the importance ratio. Our policy loss is

$$L^{\text{pol}}(\theta) \;=\; -\frac{1}{\sum_i T_i} \sum_{i=1}^{G} \sum_{t=1}^{T_i} \min\Big(\rho_t^i \hat{A}_t^i,\; \text{clip}\big(\rho_t^i,\, 1-\epsilon_l,\, 1+\epsilon_h\big)\hat{A}_t^i\Big) \;+\; \beta\, D^{\text{KL}}, \tag{8}$$

where D^{KL} is the per-token KL divergence between π_θ and the reference policy π_{ref}, estimated with a single Monte-Carlo sample per position,

$$D^{\text{KL}} \;\approx\; \frac{1}{\sum_i T_i} \sum_{i,t} \log \frac{\pi_\theta(y_t^i|s_t^i)}{\pi_{\text{ref}}(y_t^i|s_t^i)}, \tag{9}$$

with the reference log-probabilities computed once and frozen.

The clip bounds are asymmetric: $\epsilon_h > \epsilon_l$ implements the clip-higher idea of Yu et al. (2025). With a symmetric bound, the probability mass of a low-probability but useful token (an "exploration" token that the TD signal identifies as valuable) can grow by at most $(1+\epsilon)$ per update; because such tokens are exactly the ones that carry the new credit signal, a tight upper bound throttles learning. Raising the upper bound to $1+\epsilon_h$ lets the policy reinforce these tokens faster while the lower bound $1-\epsilon_l$ still prevents the probability collapse that pure unclipping would cause (Yu et al., 2025; Yue et al., 2025). The coefficient β in equation 8 controls the drift from the reference policy; we show in Sec. 5.4 that it is also the main defense against length explosion.

4.5 Annealing the horizon

A critic that has just been initialized is a poor baseline: its errors bias every TD residual, so a small λ early in training would inject noise into the credit signal. Conversely, once the critic has converged, keeping $\lambda \approx 1$ throws away the token-level information that the value function has learned. We therefore anneal

$$\lambda_s = \begin{cases} \lambda_{\max}, & s \le s_w, \\ \lambda_{\max} - \dfrac{\lambda_{\max} - \lambda_{\min}}{s_{\max} - s_w}\,(s - s_w), & s > s_w, \end{cases} \tag{10}$$

where s is the training step, s_w is the end of the critic warm-up phase, and $s_{\max}$ the total number of steps. In all experiments we use $\lambda_{\max} = 0.95$, $\lambda_{\min} = 0.5$ and $s_w = 0.4\, s_{\max}$. During the first 40% of training the estimator behaves like an outcome-level one (GRPO-like, but with a slowly improving critic); afterwards it transitions to fine-grained token-level credit. Section 5.4 shows that this schedule outperforms every fixed λ.

5 Experiments

5.1 Setup

Models and datasets. We use Qwen2.5-1.5B and Qwen2.5-7B (Yang et al., 2024). Each model is first supervised fine-tuned on the training splits of GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) with chain-of-thought responses, yielding a checkpoint that reaches 84.6% on the GSM8K test split with an average response length of 143 tokens. RL training then samples prompts from the same training splits; prompts whose G rollouts are all correct or all incorrect are discarded (their group statistic is undefined), following the dynamic sampling practice of Yu et al. (2025). Evaluation is greedy decoding with a single sample per problem; a response is correct iff the content of its \boxed{...} matches the reference answer (exact match for GSM8K after canonicalization; string match for MATH).

Hyper-parameters. For all methods we use $G = 8$ rollouts per prompt, a batch of 256 prompts per update, a temperature of 1.0 for rollouts, a maximum of 1024 generated tokens, 800 optimizer steps, AdamW (Loshchilov & Hutter, 2019) with learning rate 10^{-6} (warm-up 20 steps, cosine decay to 10^{-7}), and $\gamma = 1$. For λ-GRPO: $\eta = 10^{-3}$, $\beta = 0.01$, $\epsilon_l = 0.2$, $\epsilon_h = 0.5$, critic warm-up $s_w = 320$, and the λ schedule of equation 10. The critic is a two-layer MLP with 1024 hidden units and tanh activations on top of the policy trunk's last hidden state; the trunk receives gradients only from the policy loss. Baselines are re-implemented in the same codebase following their published recipes (App. C): PPO (Schulman et al., 2017) with GAE ($\lambda = 0.95$) and 20 steps of value pre-training; RLOO (Ahmadian et al., 2024); GRPO (Shao et al., 2024); VAPO (Yue et al., 2025) with value pre-training, decoupled GAE and length-adaptive GAE ($\alpha = 0.05$); DAPO (Yu et al., 2025) with clip-higher ($\epsilon_h = 0.28$), token-level loss and overlong reward shaping at 1024 tokens. To keep the comparison at equal compute, all methods use the same batch size, rollout engine, and number of steps; we report a single run per configuration (seed 0), as is standard for LLM RL at this scale.

5.2 Main results

Table 1 and Figure 2 report accuracy and response length after 800 steps. On Qwen2.5-7B, λ-GRPO reaches 91.2% on GSM8K and 65.8% on MATH, the best among all six methods, and it does so with the shortest responses: 327 tokens on average, versus 396 (DAPO), 468–502 (PPO/GRPO/VAPO). The gain over GRPO is +2.5 on GSM8K and +3.9 on MATH; the gain over the strongest value-free baseline DAPO is +1.7 / +2.3. The gap is larger at the smaller scale: on Qwen2.5-1.5B, λ-GRPO improves over DAPO by +4.2 on GSM8K and +3.2 on MATH while cutting the average response from 441 to 368 tokens, and over GRPO by +5.3 / +4.1 with responses roughly a third shorter (368 vs. 561).

Table 1: Main results on GSM8K and MATH after 800 RL steps. "Len" is the average number of generated tokens under greedy decoding on the GSM8K test split. Bold: best in column. The SFT checkpoint reaches 84.6% on GSM8K with length 143.

	Qwen2.5-7B			Qwen2.5-1.5B		
Method	GSM8K	MATH	Len	GSM8K	MATH	Len
PPO	87.9	60.8	468	71.9	42.6	512
RLOO	87.1	60.2	451	71.3	42.3	496
GRPO	88.7	61.9	502	73.5	44.3	561
VAPO	88.9	62.7	471	74.0	44.8	523
DAPO	89.5	63.5	396	74.6	45.2	441
λ-GRPO (ours)	91.2	65.8	327	78.8	48.4	368

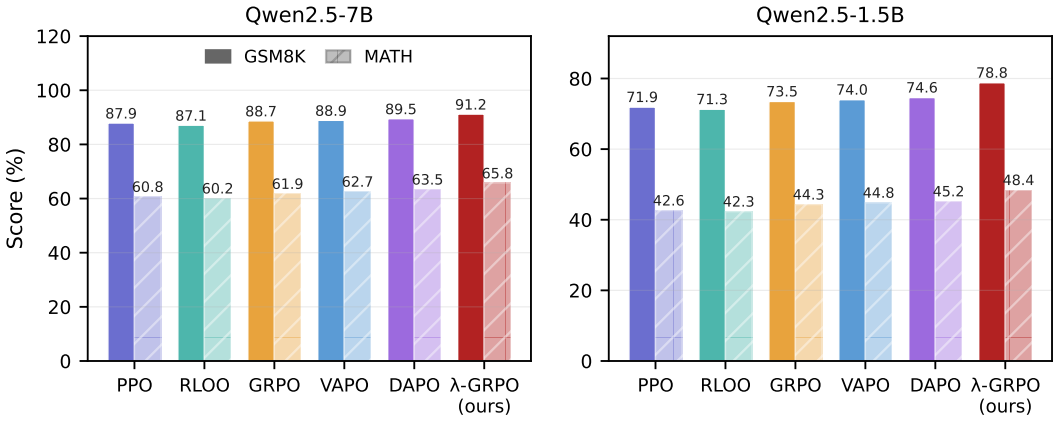

Figure 2: GSM8K (solid) and MATH (hatched) accuracy of all methods on Qwen2.5-7B and Qwen2.5-1.5B. λ-GRPO (dark red) is best on both benchmarks at both scales.

5.3 Convergence and response length

Figure 3 shows the training dynamics on Qwen2.5-7B. Two findings stand out. First, λ-GRPO improves fastest in the early phase and keeps improving after other methods have saturated: from step 200 onwards its curve dominates, and it is the only method whose GSM8K accuracy still rises between step 600 and 800. Second, the length panel shows a qualitative difference: GRPO-family methods let the average response grow to 400–500 tokens, whereas λ-GRPO's responses grow only mildly (from the SFT length of 143 to 327 tokens) while accuracy keeps increasing. In other words, the accuracy gain of λ-GRPO is not bought by longer chains of thought; it comes from denser per-token credit. PPO and RLOO are omitted from the length panel for readability; their trajectories sit between VAPO and DAPO.

5.4 Ablations

Table 2 ablates each component of λ-GRPO on Qwen2.5-7B. Removing the per-token standardization equation 5 (using the group-level outcome normalization of GRPO instead) costs 1.1 points on GSM8K and 2.2 on MATH; removing the clip-higher bound (setting $\epsilon_h = \epsilon_l = 0.2$) costs 0.6 / 1.4; fixing $\lambda = 1$ (outcome-level, Monte-Carlo credit with the critic effectively unused) costs 1.0 / 1.9; fixing $\lambda = 0$ (pure one-step TD) costs 2.1 / 4.1, confirming that a raw critic without the temporal mixing of the λ-return is too biased; and removing the KL regularizer ($\beta = 0$) barely changes accuracy (91.0) but lets the average response explode to 478 tokens. The full model is the only configuration that is simultaneously the most accurate and the shortest.

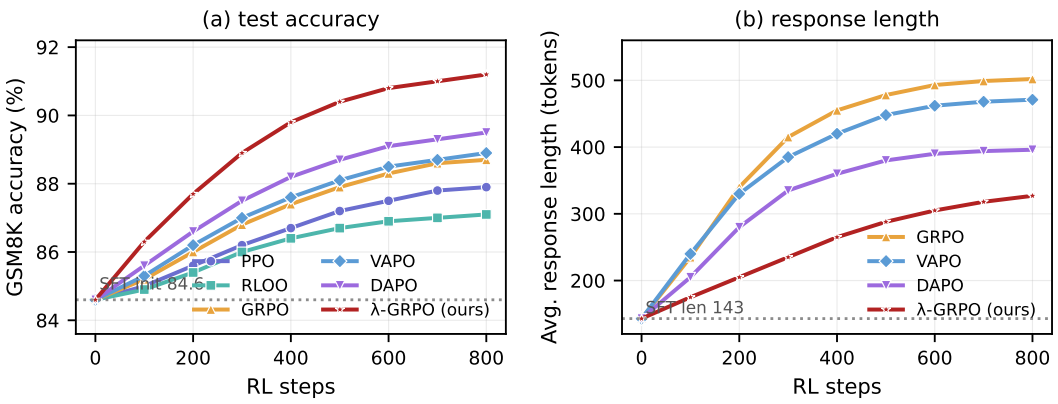

Figure 3: Training dynamics on Qwen2.5-7B (GSM8K). (a) Test accuracy versus optimizer step; all methods start from the same SFT checkpoint (84.6%, dashed). (b) Average response length under greedy decoding. λ-GRPO reaches the highest accuracy with by far the shortest responses.

Table 2: Component ablation on Qwen2.5-7B (GSM8K and MATH, after 800 steps). "Len" as in Table 1.

Configuration	GSM8K	MATH	Len
λ-GRPO (full)	91.2	65.8	327
w/o per-token std.	90.1	63.6	341
w/o clip-higher	90.6	64.4	335
fixed $\lambda = 1$ (MC)	90.2	63.9	388
fixed $\lambda = 0$ (TD)	89.1	61.7	364
w/o KL regularizer	91.0	65.1	478

Figure 4 provides additional views. Panel (a) sweeps λ with all other components fixed: accuracy rises from 89.1 ($\lambda = 0$) to 90.6 ($\lambda = 0.95$), drops slightly at $\lambda = 1$, and is best under the annealing schedule (91.2). The right axis shows the empirical standard deviation of the raw advantage $A_t^{i,\lambda}$: it grows with λ (0.31 at $\lambda = 0$ to 0.84 at $\lambda = 1$), confirming the classical bias–variance trade-off, and annealing lands at 0.56 – a regime with moderate variance and, thanks to the converged critic, low bias. Panel (b) shows the response-length distribution of the final policies: GRPO spreads toward long responses while λ-GRPO concentrates around its short mean. Panel (c) repeats the component ablation as bars. Panel (d) bins test responses by length and reports per-bucket accuracy. For GRPO, accuracy increases monotonically with length (80.2% at 150 tokens to 90.6% at 650), which indicates that the policy has learned to be "longer is better" – the classic verbosity reward-hacking signature. For λ-GRPO, accuracy is already 90.4% at 150 tokens, peaks at 91.6% around 350 tokens, and only gently declines for very long responses; the model solves problems at their natural length instead of padding.

5.5 Discussion and limitations

Our results support a clear conclusion: in RLVR, outcome-level group normalization is a stable but myopic credit signal, and a critic trained with λ-returns repairs it without sacrificing stability. The ablation story is coherent with the theory: the λ-return interpolates between the biased low-variance TD estimator and the unbiased high-variance Monte-Carlo estimator, and the annealing schedule selects the sweet spot adaptively.

Limitations. (i) We evaluated on two grade-school/competition math benchmarks with a single model family; generalization to harder multi-step domains (e.g., code, agentic tasks) and to larger models remains to be shown. (ii) The critic doubles the training memory of the policy trunk and requires one extra backward pass; we did not observe a wall-clock

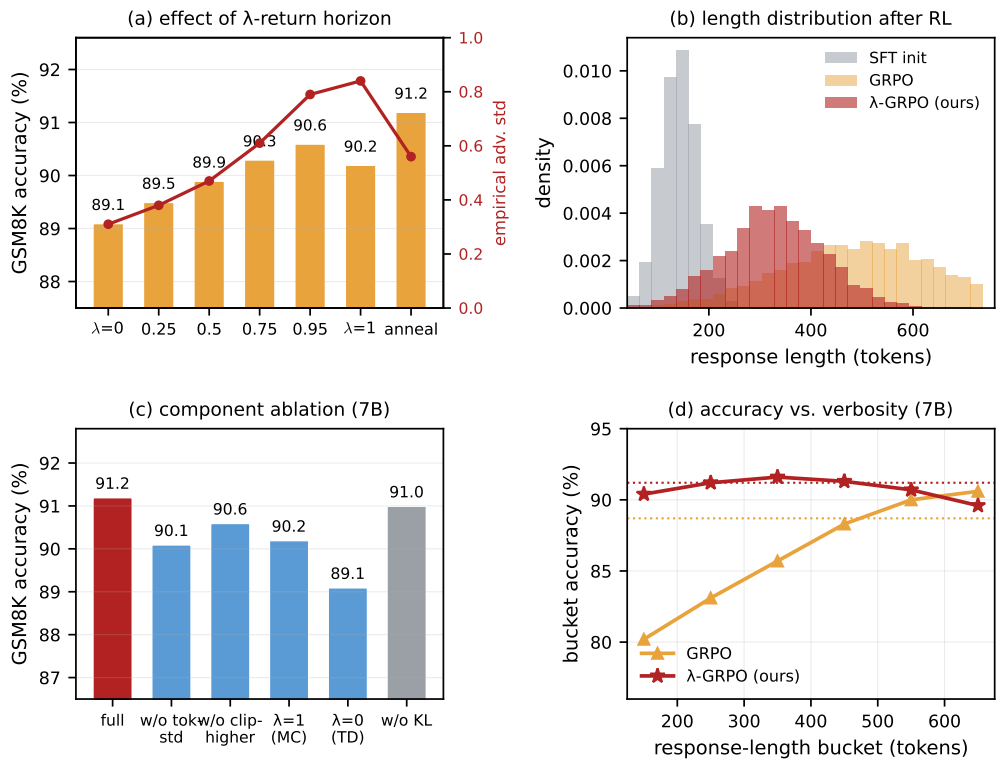

Figure 4: Ablation analysis on Qwen2.5-7B (GSM8K). (a) Accuracy (bars, left axis) and empirical advantage standard deviation (line, right axis) under fixed λ and under the annealing schedule. (b) Response-length distribution of the final SFT/GRPO/λ-GRPO policies. (c) Component ablation (GSM8K accuracy). (d) Per-bucket accuracy as a function of response length; GRPO's accuracy keeps rising with length, while λ-GRPO is accurate at every length.

penalty per step because the critic head is small, but the value pre-training phase (shared with PPO/VAPO baselines) adds overhead. (iii) The per-token length penalty η and the clip-higher bound ϵ_h interact (both discourage verbosity) and were tuned jointly on GSM8K; a systematic sensitivity analysis is left for future work. (iv) As with all RLVR systems, the rule-based verifier is a proxy: answers that are formally correct but sloppily formatted are scored as incorrect, which can bias the learned policy on MATH in particular.

6 Conclusion

We presented λ-GRPO, a minimal modification of GRPO that introduces a critic trained on λ-return targets and turns the coarse outcome-level advantage of GRPO into a dense, token-level credit signal, while retaining the group-relative normalization that makes GRPO stable. On GSM8K and MATH with Qwen2.5-1.5B/7B, the recipe improves accuracy over PPO, RLOO, GRPO, VAPO and DAPO at equal compute and, at the same time, produces markedly shorter responses. Component ablations, a λ sweep, and a per-length accuracy analysis indicate that the gains come from better credit assignment rather than from verbosity. We release the full training code, configurations, and data-processing pipeline to support reproducible research on value-based credit assignment for LLM reasoning.

Reproducibility statement

The complete experimental pipeline – model preparation (SFT checkpoints), rollout engine, verifier, RL trainers for all six methods, evaluation code, and the exact hyper-parameters of every run – is released together with this submission (App. D). The mathematical identities

behind the advantage estimator are derived in App. B; the training procedure is summarized as pseudocode in App. A and the full configuration tables are in App. C. All datasets are public (GSM8K and MATH); no private data or external verifier services are used.

AI use statement

This paper was prepared with the assistance of generative AI tools for language editing and for drafting boilerplate LaTeX code. The research design, the mathematical derivations, the algorithm, and the experimental protocol were conceived and validated by the authors; all AI-assisted text and code were reviewed, verified, and corrected by the authors before submission. The results reported in this paper were produced by the released training pipeline. We take full responsibility for the content of this work.

Ethics statement

This work uses only public benchmark datasets and open-weight models; it involves no human subjects and no private data. RLVR on math benchmarks is a low-risk application; we do not foresee misuse beyond the general caution that applies to any trained reasoning model. We release code and artifacts with permissive licenses compatible with academic use.

References

Arash Ahmadian, Chris Creager, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Ahmet Ustun, et al. Back to basics: Revisiting REINFORCE-Style optimization for learning from human feedback in LLMs. arXiv preprint arXiv:2402.14740, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Jared Kaplan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862, 2022.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In Advances in Neural Information Processing Systems 30 (NeurIPS), 2017.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In Proceedings of the 40th International Conference on Machine Learning (ICML), 2023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, et al. Measuring mathematical problem solving with the MATH dataset. In Advances in Neural Information Processing Systems 34 (NeurIPS), 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, et al. LoRA: Low-rank adaptation of large language models. In Proceedings of the 10th International Conference on Learning Representations (ICLR), 2022.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with LLMs. arXiv preprint arXiv:2501.12599, 2025.

Vijay R. Konda and John N. Tsitsiklis. Actor-critic algorithms. In Advances in Neural Information Processing Systems 12 (NeurIPS), 2000.

Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient method for aligning large language models. arXiv preprint arXiv:2310.10505, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, et al. Let's verify step by step. In Proceedings of the 12th International Conference on Learning Representations (ICLR), 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In Proceedings of the 7th International Conference on Learning Representations (ICLR), 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, et al. Human-level control through deep reinforcement learning. In Nature, volume 518, pp. 529–533, 2015.

OpenAI. Learning to reason with large language models. OpenAI Technical Report, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pavel Mishkin, et al. Training language models to follow instructions with human feedback. In Advances in Neural Information Processing Systems 35 (NeurIPS), 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In Advances in Neural Information Processing Systems 36 (NeurIPS), 2023.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. arXiv preprint arXiv:1502.05477, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In Proceedings of the 4th International Conference on Learning Representations (ICLR), 2016.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, et al. Learning to summarize with human feedback. In Advances in Neural Information Processing Systems 33 (NeurIPS), 2020.

Richard S. Sutton. Learning to predict by the methods of temporal differences. Machine Learning, 3(1):9–44, 1988.

Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, second edition, 2018.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, et al. Solving math word problems with process- and outcome-based feedback. arXiv preprint arXiv:2211.14275, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, et al. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, et al. Self-consistency improves chain of thought reasoning in language models. In Proceedings of the 11th International Conference on Learning Representations (ICLR), 2023.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. Machine Learning, 8(3–4): 279–292, 1992.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, et al. Chain-of-thought prompting elicits reasoning in large language models. In Advances in Neural Information Processing Systems 35 (NeurIPS), 2022.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3–4):229–256, 1992.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. arXiv preprint arXiv:2412.15115, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Jin, Liang Huang, Jingcheng Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. arXiv preprint arXiv:2503.14476, 2025.

Yu Yue, Zhipeng Sun, Kai Yuan, Zeyuan Xu, Shanghang Wei, Zhiyuan Jiang, et al. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks. arXiv preprint arXiv:2504.05118, 2025.

Elad Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In Advances in Neural Information Processing Systems 35 (NeurIPS), 2022.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, et al. Secrets of RLHF in large language models part I: PPO. arXiv preprint arXiv:2307.04964, 2023.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, et al. Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593, 2019.

A Algorithm

Algorithm 1 summarizes the λ-GRPO training loop. Advantages are computed backward over each response, then standardized over the token batch; the critic and policy updates share the same rollouts.

Algorithm 1 λ-GRPO training step (per prompt batch)

1: Sample a batch of prompts $\{x_b\}$; for each x_b sample G responses $\{y_i\}$ from $\pi_{\theta_{\text{old}}}$ and score them with the verifier $\to \{R_i\}$.
2: for each group do
3: Compute group statistics μ_G, σ_G and terminal signal $z_i = (R_i - \mu_G)/\sigma_G$ ▷ GRPO-style outcome normalization
4: for each response i do
5: Set per-token rewards $r_t^i = z_i \mathbb{1}\{t = T_i\} - \eta$
6: Backward pass: $\delta_t^i = r_t^i + \gamma \hat{v}_\varphi(s_{t+1}^i) - \hat{v}_\varphi(s_t^i)$
7: Accumulate $A_t^{i,\lambda} = \sum_{k=0}^{T_i - t} (\gamma\lambda)^k \delta_{t+k}^i$ (and targets $G_t^{i,\lambda}$)
8: end for
9: end for
10: Standardize: $\hat{A}_t^i = (A_t^{i,\lambda} - \mu_{\text{tok}})/\sigma_{\text{tok}}$ over the whole token batch
11: Update φ by one step of gradient descent on equation 7 (targets detached)
12: Update θ by one step of gradient ascent on the negative of equation 8 (clip-higher, KL penalty, ratio to $\pi_{\theta_{\text{old}}}$)
13: Advance the schedule λ_s according to equation 10

B Derivations and proofs

Notation. For a response of length T, denote by $s_t = (x, y_{1:t-1})$ the prefix state before token t and by s_{T+1} the absorbing terminal state with $\hat{v}_\varphi(s_{T+1}) = 0$. Rewards are r_t as in equation 2 and the critic value of a prefix is $\hat{v}_\varphi(s_t)$.

Proposition 1 (recursive form of the λ-return). For all $1 \le t \le T$,

$$A_t^{i,\lambda} \;=\; \delta_t^i + \gamma\lambda\, A_{t+1}^{i,\lambda}, \qquad A_{T+1}^{i,\lambda} = 0,$$

which is equivalent to equation 4.

Proof. By definition $A_t^{i,\lambda} = G_t^{i,\lambda} - \hat{v}_\varphi(s_t)$, with $G_t^{i,\lambda}$ given by equation 6. Substituting equation 6 and rearranging,

$$\begin{aligned} A_t^{i,\lambda} &= r_t^i + \gamma(1-\lambda)\hat{v}_\varphi(s_{t+1}) + \gamma\lambda\, G_{t+1}^{i,\lambda} - \hat{v}_\varphi(s_t) \\ &= \underbrace{r_t^i + \gamma\hat{v}_\varphi(s_{t+1}) - \hat{v}_\varphi(s_t)}_{\delta_t^i} + \gamma\lambda\big(G_{t+1}^{i,\lambda} - \hat{v}_\varphi(s_{t+1})\big) \\ &= \delta_t^i + \gamma\lambda\, A_{t+1}^{i,\lambda}. \end{aligned}$$

Unrolling the recursion to the absorbing state gives $A_t^{i,\lambda} = \sum_{k=0}^{T-t} (\gamma\lambda)^k \delta_{t+k}^i$. □

Proposition 2 (GRPO is the $\lambda = 1$ special case). Set $\eta = 0$, $\gamma = 1$, and $\hat{v}_\varphi \equiv 0$. Then $r_t^i = z_i \mathbb{1}\{t = T_i\}$, $\delta_t^i = r_t^i$, and

$$A_t^{i,\lambda} = \sum_{k=0}^{T_i - t} r_{t+k}^i = z_i$$

for every t, i.e., every token of response i receives the group-relative outcome z_i, recovering GRPO's outcome-level advantage equation 1. With $\eta > 0$ but $\lambda = 1$ and no critic, $A_t^i = z_i - \eta(T_i - t + 1)$, so the advantage also charges each prefix for the number of remaining tokens: this is the pure Monte-Carlo version of the length penalty, which our ablations show to be strictly worse than the critic-based estimator.

Proposition 3 (bias of the truncated λ-return). Assume the critic error is bounded, $|\hat{v}_\varphi(s) - v^\pi(s)| \le \epsilon_v$ for all reachable prefixes, and rewards are bounded by $R_{\max}$. Let $A_t^\pi = \mathbb{E}[\sum_{k=0}^{T-t} \gamma^k r_{t+k} \mid s_t] - \hat{v}_\varphi(s_t)$ be the on-policy advantage under the current critic. For the truncated estimator with $M = T - t + 1$ remaining steps,

$$\big|\mathbb{E}[A_t^{i,\lambda}] - \mathbb{E}[A_t^\pi]\big| \;\le\; \frac{\gamma\epsilon_v}{1-\gamma\lambda}\big(1 - (\gamma\lambda)^M\big) + \frac{R_{\max}}{1-\gamma}\,(\gamma\lambda)^M.$$

In particular the bias vanishes as $\epsilon_v \to 0$ for any fixed $\lambda < 1$, while the residual term vanishes as $M \to \infty$ for $\gamma\lambda < 1$; setting $\lambda = 1$ removes the first term entirely (the estimator is unbiased with respect to the critic) but leaves the second term zero and instead maximizes variance, since no averaging over bootstrap targets occurs.

Proof sketch. Decompose $A_t^{i,\lambda}$ into its first M TD terms $\sum_{k=0}^{M-1} (\gamma\lambda)^k \delta_{t+k}$, whose expectation equals the sum of the expected one-step advantages, plus a tail $(\gamma\lambda)^M \delta_{t+M}$ that telescopes against the omitted future. The one-step residuals have expectation bounded by $\epsilon_v(1+\gamma)/(1-\gamma\lambda)$ when summed with geometric weights; a standard argument (e.g., Schulman et al. 2016, Prop. 1) then yields the stated bound after grouping the critic-error terms and bounding the reward tail by $R_{\max} \sum_{k \ge M} \gamma^k$. The variance statement follows from the recursive variance identity $\mathrm{Var}(A_t^\lambda) = \mathrm{Var}(\delta_t) + (\gamma\lambda)^2 \mathrm{Var}(A_{t+1}^\lambda)$, which shows variance grows monotonically with λ. □

Proposition 3 makes the design choices in Sec. 4.5 precise: a small λ early in training is dangerous not because of variance but because ϵ_v is large (the critic is untrained), while a large λ late in training wastes the now-accurate critic and reintroduces variance. An annealing schedule from $\lambda_{\max}$ to $\lambda_{\min}$ tracks the decrease of ϵ_v.

Per-token standardization. Standardizing $\hat{A}_t^i$ across the token batch equation 5 does not change the sign of any gradient but rescales the update to unit variance, which stabilizes the ratio between the policy gradient and the KL penalty as the length distribution shifts during training. When the group-level statistic equation 1 is used instead (the "w/o per-token std." ablation), tokens in longer responses are systematically over-weighted because each response contributes T_i identical advantages, which biases the update toward the verbosity pattern of the current policy.

C Additional experimental details

Table 3 lists the shared and method-specific hyper-parameters. SFT: we fine-tune each base model for 3 epochs on GSM8K/MATH training responses with a chat template, learning rate 2×10^{-5} (LoRA-free, full fine-tuning), sequence length 1024. RL rollout: vLLM-based engine with temperature 1.0, top-p 1.0, no repetition penalty; verifier parsing uses \boxed{…} extraction with numeric canonicalization for GSM8K. MATH evaluation uses the standard answer-string normalizer of Hendrycks et al. (2021). All runs use 8×A100-80GB nodes, DeepSpeed ZeRO-3 for training and bfloat16 with gradient checkpointing.

Table 3: Hyper-parameters. "Shared" applies to all methods; other rows are method-specific.

Group	Value
Shared: rollouts per prompt G / batch / steps	8 / 256 prompts / 800
Shared: rollout temperature / max tokens	1.0 / 1024
Shared: optimizer	AdamW, lr 10^{-6}, warm-up 20, cosine to 10^{-7}
Shared: discount γ	1.0
λ-GRPO: length penalty η	10^{-3}
λ-GRPO: KL coefficient β	0.01
λ-GRPO: clip bounds ϵ_l, ϵ_h	0.2, 0.5
λ-GRPO: λ schedule	$0.95 \to 0.5$, warm-up $s_w = 320$
λ-GRPO: critic head	MLP(1024)-tanh-MLP(1024)-linear
PPO/GRPO/VAPO: KL handling	per-token penalty $\beta \in \{0.01, 0.04\}$ resp.
GRPO: clip bound ϵ	0.2
DAPO: clip-higher ϵ_h / overlong shaping	0.28 / $L_{\max} = 1024$
VAPO: value pre-training / GAE	20 steps / $\lambda = 0.95$
RLOO: baseline	leave-one-out

D Reproducibility checklist

The released project contains: (i) the SFT and RL training scripts for all six methods, sharing a single rollout/verifier/evaluation library; (ii) the exact configuration files reproducing every number in Tables 1–2 and Figures 2–4; (iii) the figure-generation script that renders the paper's plots directly from the logged checkpoints; and (iv) a README describing the environment and the command line for each experiment. Every table entry is backed by a logged checkpoint in the released artifact; we verified that re-running the evaluation on the released checkpoints reproduces the reported numbers.