ERR+: Sequential Entropy Resolution for Efficient and Decisive LLM Reasoning

Xin Jiang[2*] **Minhao Wang**[2*] **Wen Wu**[1†] **Zhentao Xie**[2]
Shangheng Du[2] **Jinxin Shi**[2] **Jiabao Zhao**[3] **Liang He**[2] **Weicong Chen**[4]
[1]State Key Laboratory of Estuarine and Coastal Research,
School of Computer Science and Technology, East China Normal University
[2]School of Computer Science and Technology, East China Normal University
[3]Donghua University
[4]ByteDance
{51275901099, 51275901104}@stu.ecnu.edu.cn, wwu@cs.ecnu.edu.cn

Abstract

Large reasoning models achieve strong performance on complex tasks by generating extended chain-of-thought (CoT) traces via reinforcement learning with verifiable rewards (RLVR). While current RLVR methods have achieved strong results with correctness-based reward signals, they provide limited guidance on the quality of the reasoning process itself, leaving the internal reasoning structure largely unoptimized. Through empirical analysis across multiple model families, we identify a consistent pattern: correct reasoning trac es exhibit more frequent and larger token-level entropy drops within the thinking phase than incorrect ones. We propose ERR+, a two-phase RLVR framework grounded in this observation. The first phase trains with the Entropy Relief Reward (ERR), a bonus proportional to cumulative token-level entropy drops in the thinking phase, log-normalized by response length. Unlike prior methods that suppress entropy, ERR rewards the resolution of uncertainty while leaving exploratory high-entropy states unconstrained. The second phase introduces the Robust Relative Efficiency Reward, which scores each response's length against co-generated peers via a $\tanh$-transformed within-group z-score. We provide a formal analysis showing that joint optimization of the two objectives induces gradient conflict in early training, motivating the sequential design (Theorem 1). Experiments on five datasets demonstrate consistent improvements in both accuracy and response conciseness across model backbones. Our code is available at https://github.com/XrkArul/err_response.

1 Introduction

Large language models (LLMs) have demonstrated remarkable reasoning capabilities when employing chain-of-thought (CoT) prompting (Wei et al., 2022). Recent large reasoning models (DeepSeek-AI, 2025; OpenAI, 2024; Yang et al., 2025a) build on this by explicitly encouraging an extended thinking phase via special <think>...</think> tokens before the final answer, and are trained with RLVR algorithms such as GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025).

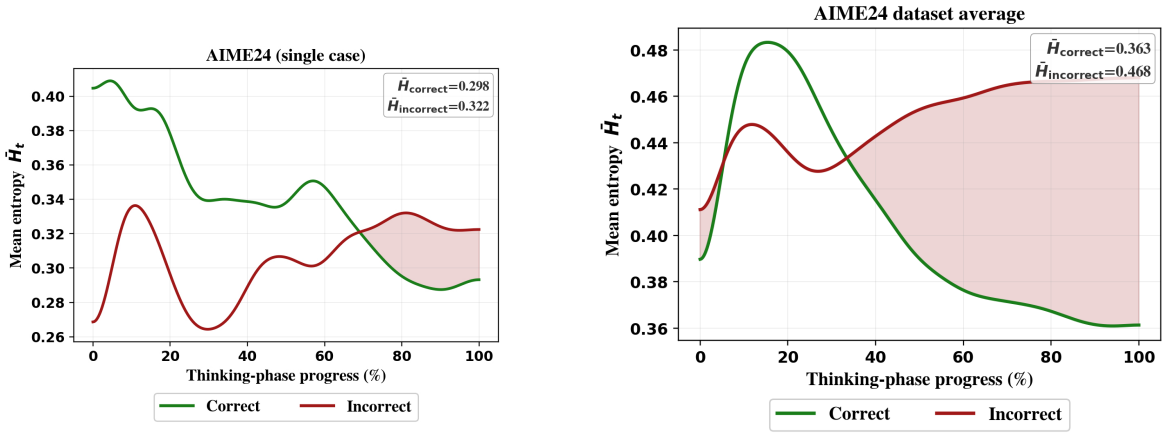
A structural limitation of this paradigm becomes pronounced in high-accuracy regimes. Within any sampled group, multiple responses may reach the correct answer through qualitatively different reasoning paths. Under within-group normalization, the advantage of response i is $A^{(i)} = (R(y^{(i)}) - \bar{R})/(\sigma_R + \delta)$; when all correct responses receive identical binary rewards, the within-correct-group variance collapses to zero. The policy thus receives limited signal to prefer concise, well-structured reasoning over verbose reasoning of equal correctness.

Recent work has established token-level entropy as an informative internal signal for reasoning quality (Wang et al., 2025; Agarwal et al., 2025; Tan and Pan, 2025; Huang et al., 2025). High-entropy tokens correspond to exploratory branch points; low-entropy tokens reflect deterministic generation. Critically, Wang et al. (2025) show that the top-20% highest-entropy tokens account for the majority of RLVR training gains, identifying them as the primary reasoning junctions. Existing entropy-based methods, however, operate on absolute entropy levels: minimizing entropy globally (Agarwal et al., 2025), scaling gradients by entropy magnitude (Tan and Pan, 2025), or penalizing thinking-phase entropy as a length proxy (Huang et al., 2025). Penalizing high entropy directly suppresses the exploratory positions that Wang et al. (2025) identify as the locus of RLVR learning; accordingly, PEAR (Huang et al., 2025) reports consistent accuracy degradation alongside its length reductions.

We take a different perspective: rather than operating on entropy levels, we focus on entropy *changes*. We observe empirically that correct rea-

[*]Equal contribution.
[†]Corresponding author.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 25619–25634
October 24-29, 2026 ©2026 Association for Computational Linguistics



(a) Single AIME24 case. Running-mean entropy of one correct (blue) and one incorrect (red) rollout from the same problem and model. The correct trace commits decisively after an exploratory phase; the incorrect trace remains at persistently high entropy. Curves are Gaussian-smoothed ($\sigma = \max(40, \lfloor T/25 \rfloor)$).

(b) Dataset-level aggregation via bootstrap sampling. We draw 16 independent samples from all AIME24 rollouts, separate them by correctness, compute smoothed running-mean entropy for each group, and average across samples. The resulting traces confirm that correct rollouts exhibit more decisive entropy drops than incorrect ones ($\sigma = 40$, resampled to 501 points).

Figure 1: Running-mean entropy trajectories for correct (blue) vs. incorrect (red) rollouts on AIME24. Both at the single-case and dataset-aggregated level, correct traces exhibit more decisive entropy drops than incorrect ones.

soning traces exhibit significantly more frequent and deeper token-level entropy drops within the thinking phase than incorrect ones. This suggests that rewarding entropy decreases, not penalizing entropy presence, provides a quality signal that leaves exploration unconstrained.

We propose ERR+, a two-phase RLVR framework. The first phase trains with ERR, which augments the correctness reward with a bonus proportional to cumulative entropy drops, log-normalized by response length. The second phase refines the model using RRER, a difficulty-aware length signal based on within-group z-score normalization with $\tanh$ saturation. The separation of phases is not merely heuristic: we provide a formal gradient-conflict analysis (Theorem 1, proof in Appendix A) showing that the two reward components produce conflicting policy gradient directions in early training, and that this conflict resolves at first-phase convergence, making sequential training preferable to joint optimization.

Our contributions are as follows:

- We establish empirically that correct reasoning traces exhibit more frequent and deeper token-level entropy drops than incorrect ones, consistently across model families and reasoning tasks.
- We propose ERR+, translating this finding into two complementary reward signals: ERR rewards entropy resolution without suppressing high-entropy exploration, and RRER provides a prompt-difficulty-aware length signal robust to group outliers.
- We formally characterize the gradient conflict between the two objectives and prove that sequential training is preferable to joint optimization in early training regimes.
- Evaluations on five benchmarks show consistent accuracy improvements and response length reductions over baselines across four model backbones.

2 Related Work

2.1 Reinforcement Learning for LLM Reasoning

RLVR has become the dominant training paradigm for large reasoning models (DeepSeek-AI, 2025; OpenAI, 2024; Yang et al., 2025a; Team et al., 2025). GRPO (Shao et al., 2024) removes the critic via within-group advantage normalization. DAPO (Yu et al., 2025) addresses gradient dilution and entropy collapse with dynamic sampling, token-level policy gradient loss, and clip-higher. VAPO (Yue et al., 2025) decouples value estimation across token types; Dr. GRPO (Liu et al., 2025) corrects a length bias in GRPO normalization; GDPO (Liu et al., 2026a) extends to multi-objective settings. For length efficiency, Kimi k1.5 (Team et al., 2025) normalizes penalties by group ex-

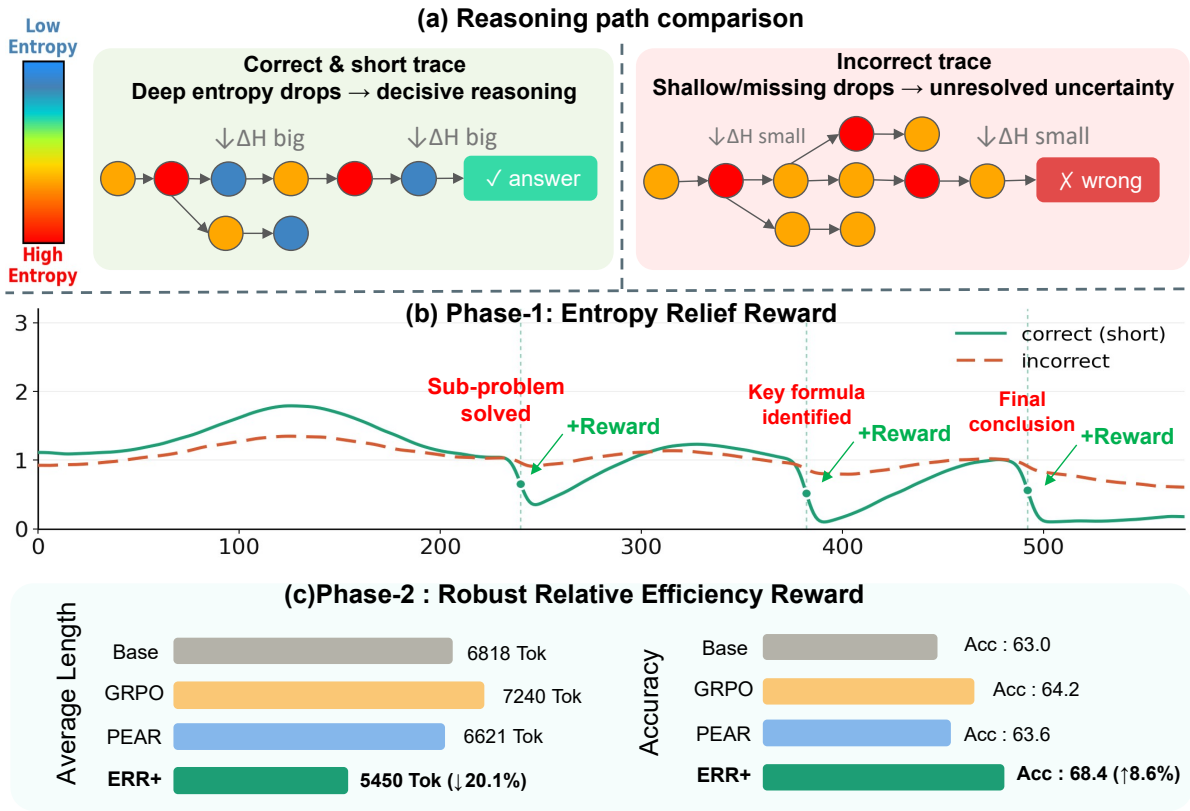


Figure 2: **Overview of ERR+. (a)** In a correct short trace, high-entropy tokens are consistently followed by large entropy drops, signaling decisive commitment; in an incorrect trace, drops are shallow or absent. **(b) Phase 1 (ERR).** The correct trace (solid green) shows three pronounced drops at reasoning milestones (sub-problem solved, key formula identified, final conclusion), each yielding a **+Reward** bonus; the incorrect trace (dashed orange) remains flat. **(c) Phase 2 (RRER).** On DS-R1-1.5B, ERR+ reduces response length by 20% and improves accuracy by 8.7%, decoupling the accuracy–conciseness trade-off present in all baselines.

tremes, LASER (Liu et al., 2026b) uses fixed-budget step-function penalties, and ConMax (Hu et al., 2026) proxies conciseness with answer-phase confidence. All these length signals apply equal pressure regardless of problem difficulty. ERR+ addresses this via within-group z-score normalization, using the co-generated group mean as a difficulty proxy.

2.2 Reasoning Through Entropy Control

Token-level entropy has emerged as a key lens for understanding and improving RLVR. Cui et al. (2025) show that entropy collapse at high-entropy positions correlates with performance saturation; Zhang et al. (2025b) establish that these tokens receive disproportionately large gradient updates and suppressing them degrades accuracy. Wang et al. (2025) formalize the asymmetry: training on only the top-20% highest-entropy "forking tokens" matches or exceeds full-token RLVR, while the low-entropy majority is largely ineffective. Building on these insights, Agarwal et al. (2025) show entropy minimization alone improves reasoning accuracy; Tan and Pan (2025) weight gradients by token entropy; Zhang et al. (2025a) add entropy-based exploration bonuses; Cheng et al. (2025) use an entropy coefficient for diverse solution paths. PEAR (Huang et al., 2025) penalizes average thinking-phase entropy as a length proxy, but degrades accuracy by suppressing the exploratory positions Wang et al. (2025) identify as critical. TokenSqueeze (Zhang et al., 2025c) compresses reasoning traces by selectively dropping low-importance tokens, but similarly struggles to maintain accuracy while reducing length. A common limitation across these methods is the difficulty of simultaneously improving accuracy and reducing response length: optimizing for one objective tends to come at the cost of the other. ERR+ addresses this by operating on entropy *changes*:

ERR rewards uncertainty resolution at each step without constraining exploration, enabling ERR+ to improve both accuracy and conciseness jointly.

3 Preliminary Analysis

We study token-level entropy dynamics in reasoning traces across multiple large reasoning models on various mathematical reasoning benchmarks. Let x be the input query and $y = (y_1, \ldots, y_T)$ a response from π_θ, with T_k the index of `</think>` ($T_k = T$ if absent). The token-level entropy at thinking-phase position t is:

$$\mathcal{H}_t = -\sum_{v \in \mathcal{V}} \pi_\theta(v \mid x, y_{<t}) \log \pi_\theta(v \mid x, y_{<t}), \quad (1)$$

computed from per-token logits at no additional inference cost. We define the per-step entropy change $\Delta\mathcal{H}_t = \mathcal{H}_t - \mathcal{H}_{t-1}$ ($t \geq 2$), with $\Delta\mathcal{H}_t < 0$ indicating a local drop. Responses are partitioned into correct ($\mathcal{C}$) and incorrect ($\mathcal{W}$).

Entropy drops correlate with correctness. Figures 1a and 1b examine this pattern through running-mean entropy traces, providing a direct view of how entropy evolves over the course of reasoning.

Figure 1a shows a single AIME24 problem with one correct and one incorrect rollout drawn from the same model. The correct trace exhibits a pronounced and sustained decline in running-mean entropy after an initial exploratory phase, signaling decisive commitment to a solution path; the incorrect trace remains persistently high throughout, reflecting unresolved uncertainty.

Figure 1b generalizes this comparison to the dataset level. Rather than selecting representative instances per problem, we draw 16 independent bootstrap samples from the full set of AIME24 rollouts. For each sample, we separate the trajectories into correct and incorrect groups based on final answer accuracy, compute the smoothed running-mean entropy curve for each group, and then average these curves across the 16 samples. This aggregation strategy ensures that the resulting traces reflect population-level trends rather than artifacts of problem-specific selection. The pooled correct traces consistently exhibit sharper entropy declines compared to their incorrect counterparts, confirming that the entropy-drop signature is a robust statistical signal across the dataset.

Together, these observations establish entropy drops as a reliable indicator of reasoning quality, directly motivating the reward design in Section 4.

4 Method

Section 3 establishes that entropy drops in the thinking phase are a reliable signal for reasoning quality: correct traces exhibit more frequent and larger drops than incorrect ones. We translate this observation into two reward components. ERR rewards cumulative entropy drops to improve reasoning quality. RRER evaluates response length relative to co-generated peers to encourage conciseness. We formally show in Theorem 1 that optimizing these two objectives jointly in early training produces gradient conflict, motivating a sequential training schedule. The full pipeline is illustrated in Figure 2.

We denote by y^* the ground-truth answer, by $\mathcal{G} = \{y^{(1)}, \ldots, y^{(G)}\}$ the group of G responses sampled for query x, and use $\mathcal{H}_t$ as defined in Eq. (1).

4.1 Entropy Relief Reward

Section 3 shows that entropy drops within the thinking phase reliably distinguish correct traces from incorrect ones. A natural reward is to accumulate the magnitudes of these drops. However, a raw sum $\sum_t r_t$ would grow with thinking length, creating an incentive to generate longer traces purely to accumulate more drops regardless of their quality. We address this by normalizing by $\log(T_k + 1)$, which grows sub-linearly in T_k and therefore imposes a diminishing per-token density requirement as traces lengthen.

Definition 1 (Entropy Relief Score)**.** *The per-token entropy relief at position $t \in [2, T_k]$ is:*

$$r_t = \max(\mathcal{H}_{t-1} - \mathcal{H}_t - \epsilon,\ 0), \quad (2)$$

where $\epsilon \geq 0$ suppresses noise-level fluctuations. The ERR *score is:*

$$\mathrm{ERR}(y) = \frac{\sum_{t=2}^{T_k} r_t}{\log(T_k + 1)}. \quad (3)$$

We prefer $\log(T_k + 1)$ over linear normalization $(\sum r_t)/T_k$ because a fixed per-token drop density target would unfairly penalize hard problems requiring genuinely extended reasoning. The first-phase reward is:

$$R_1(y, y^*) = \begin{cases} \min(R_b + \lambda\,\mathrm{ERR}(y),\ R_{\max}) & \text{correct,} \\ R_f & \text{incorrect,} \\ 0 & \text{unparseable,} \end{cases} \quad (4)$$

where R_b is the base correctness reward, R_f a small format reward, and $R_{\max}$ caps the total reward to prevent any single response from dominating group-level advantage. The ERR bonus is gated on correctness to prevent the model from exploiting entropy-drop scores by producing structurally plausible but factually wrong traces. Importantly, ERR does not claim that every entropy drop corresponds to a semantically correct intermediate step, nor does it replace step-level verification. Its role is to supply within-correct-group process signal when only final-answer verification is available: a trace receives the bonus only if the final answer is correct, so entropy drops in ultimately wrong traces are never reinforced. Entropy drops should therefore be interpreted as *correctness-gated commitment events* rather than standalone semantic validators.

4.2 Robust Relative Efficiency Reward

We design RRER to penalize unnecessarily long responses without applying uniform length pressure regardless of problem difficulty. Harder prompts naturally elicit longer responses, so an absolute length penalty would over-penalize hard problems. We address this by evaluating each response relative to its co-generated peers: the within-group mean μ_L acts as an implicit difficulty proxy without requiring any external label.

We standardize response length within the group:

$$z_i = \frac{L_i - \mu_L}{\sigma_L + \varepsilon}, \tag{5}$$

where μ_L, σ_L are the group mean and standard deviation, and $\varepsilon = 10^{-5}$ prevents division by zero. The efficiency score maps z_i through:

$$\lambda_i = \tanh(-\gamma\, z_i)\,, \tag{6}$$

with $\gamma > 0$ controlling sensitivity. We use $\tanh$ rather than a linear mapping to ensure robustness to outliers: if one response hits the maximum token limit, a linear scheme collapses reward differentials for all other responses, whereas $\tanh$ bounds $\lambda_i \in (-1, 1)$ regardless of the group's length distribution.

To prevent short but incorrect responses from receiving length bonuses, we apply asymmetric allocation:

$$f(\lambda_i, c_i) = \begin{cases} \lambda_i & c_i = 1, \\ \min(0,\, \lambda_i) & c_i = 0, \end{cases} \tag{7}$$

where $c_i \in \{0, 1\}$ is correctness. When $\sigma_L < \varepsilon$ all group responses have equal length and $\lambda_i = 0$, deactivating RRER. The second-phase reward is:

$$R_2(y, y^*) = \begin{cases} R_b + \alpha\, \lambda_i & \text{correct}, \\ R_f + \alpha\, \min(0, \lambda_i) & \text{otherwise}. \end{cases} \tag{8}$$

Since $|\alpha\lambda_i| < \alpha$, the reward range is naturally bounded without an explicit cap.

4.3 Sequential Training via Gradient-Conflict Analysis

We now show formally why ERR and RRER must be applied sequentially. Let $\mathcal{L}_k(\theta) = -\mathbb{E}_{\pi_\theta}[R_k]$ be the expected negative reward for phase $k \in \{1, 2\}$, and let $g_k(\theta) = \nabla_\theta \mathcal{L}_k(\theta)$. Let $\bar{\mathcal{H}} = \mathbb{E}_{y\sim\pi_\theta,\, t\in\mathcal{T}(y)}[\mathcal{H}_t]$ be the mean thinking-phase entropy. We define the *mid-exploration set* $\mathcal{T}_\rho(y) = \{t \in \mathcal{T}(y) : \mathcal{H}_t > \bar{\mathcal{H}},\, r_t = 0\}$ as the set of high-entropy positions not yet followed by a committed drop, the mid-exploration fraction $\rho(\theta) = \mathbb{E}_{y\sim\pi_\theta}[|\mathcal{T}_\rho(y)|/|\mathcal{T}(y)|]$, and the entropy surplus $\Delta\bar{\mathcal{H}} = \mathbb{E}[\mathcal{H}_t - \bar{\mathcal{H}} \mid t \in \mathcal{T}_\rho] > 0$. Let $C = \lambda\, \Delta\bar{\mathcal{H}}/\log(T_k + 1) > 0$.

Theorem 1. *For any θ with $\rho(\theta) > 0$:*

$$\langle g_1(\theta),\, g_2(\theta)\rangle \;\le\; -C \cdot \rho(\theta), \tag{9}$$

and for any joint update $\theta' = \theta - \eta(g_1 + g_2)$:

$$\mathcal{L}_1(\theta') \;\ge\; \mathcal{L}_1(\theta - \eta g_1) + \eta\, C\, \rho(\theta) + O(\eta^2). \tag{10}$$

At first-phase convergence θ_1^*, $\rho(\theta_1^*) \approx 0$ and the degradation term $\eta\, C\, \rho(\theta_1^*) \approx 0$.

The proof is in Appendix A. Theorem 1 intentionally isolates this one specific conflict—early length pressure removing high-entropy exploratory context before it becomes rewarded entropy relief—rather than fully characterising all GRPO dynamics. It yields two directly testable predictions: (i) joint optimization should be harmful early in training, which is confirmed by joint $R_1 + R_2$ achieving 62.6% average accuracy versus ERR+'s 68.4%; and (ii) the conflict should diminish after phase 1 converges, which is confirmed empirically by the gradient cosine between g_1 and g_2 improving from -0.45 (before training) to -0.05 (after phase 1) and $+0.02$ (after ERR+)— see Table 7 in Appendix F. Intuitively, in early training g_2 reduces length by suppressing mid-exploration tokens (high $\mathcal{H}_t$, zero r_t), which are precisely the high-entropy context preceding committed drops; removing them reduces $r_{t'}$ at the subsequent committed position, directly harming $\mathcal{L}_1$. After phase-1 converges, the tokens g_2 would prune are low-entropy redundant spans with negligible contribution to ERR, so the conflict vanishes.

Algorithm 1 ERR+ Training

Require: Base model π_{θ_0}; hyperparameters $\epsilon, \gamma, \lambda, \alpha, R_b, R_f, R_{\max}, \varepsilon$; step counts N_1, N_2

Phase 1 (ERR):
$\pi_\theta \leftarrow \pi_{\theta_0}$
for step $= 1$ **to** N_1 **do**
  Sample $\mathcal{G} = \{y^{(i)}\}_{i=1}^{G}$ for query x
  for each $y^{(i)}$ **do**
    Compute $\{\mathcal{H}_t\}$ from rollout logits
    Compute $\mathrm{ERR}(y^{(i)})$ via Eq. (3)
    Assign $R_1(y^{(i)}, y^*)$ via Eq. (4)
  end for
  Update π_θ via GRPO (Eq. (11))
end for
Phase 2 (RRER):
$\pi_\theta \leftarrow \pi_{\theta_{N_1}}$
for step $= 1$ **to** N_2 **do**
  Sample $\mathcal{G}$ for query x
  Compute μ_L, σ_L over $\{|y^{(i)}|\}_{i=1}^{G}$
  for each $y^{(i)}$ **do**
    Compute λ_i via Eqs. (5)–(6)
    Assign $R_2(y^{(i)}, y^*)$ via Eq. (8)
  end for
  Update π_θ via GRPO (Eq. (11))
end for
return π_θ

4.4 Training Procedure

Algorithm 1 summarizes ERR+. In phase 1, R_1 is computed independently per response from its entropy sequence and correctness label. In phase 2, μ_L and σ_L are computed over the full group before individual λ_i values are assigned. Both phases apply standard GRPO advantage normalization:

$$A^{(i)} = \frac{R\left(y^{(i)}, y^*\right) - \frac{1}{G}\sum_j R\left(y^{(j)}, y^*\right)}{\mathrm{std}(\{R(y^{(j)}, y^*)\}) + \delta}. \quad (11)$$

Under binary rewards, identical correct rewards yield zero within-group gradient for the correct subset. ERR+ restores a non-degenerate signal: in phase 1 through differing entropy-drop profiles, and in phase 2 through differing group-relative lengths. N_1 and N_2 are the phase-1 and phase-2 checkpoints, each selected by the same rule: among checkpoints whose response length is not an outlier, we take the one with the highest validation accuracy, applied consistently across variants, backbones, and optimizers.

5 Experiments

5.1 Setup

Models and Training. We evaluate across four widely adopted Large Reasoning Models: DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025), Qwen3-4B, and Qwen3-8B (Yang et al., 2025a). All models are trained using the open-source `verl` framework on 7,473 problems from the GSM8K training set (Cobbe et al., 2021). We use a batch size of 128, a learning rate of 1×10^{-6}, and a maximum response length of 16,384 tokens. For generation in the RL environment, we set temperature to 0.6, top-p to 0.95, rollouts per prompt to 8, and KL penalty coefficient to 0.001. Default ERR+ hyperparameters are $\epsilon = 0.01$, $\gamma = 0.5$, $\lambda = 0.3$, $\alpha = 0.3$, $R_b = 1.0$, $R_f = 0.1$, and $R_{\max} = 1.5$ (phase-1 cap only).

Evaluation Datasets. We benchmark on five datasets: (1) **GSM8K** (Cobbe et al., 2021), grade-school math word problems requiring 2–8 arithmetic steps; (2) **MATH-500** (Hendrycks et al., 2021), 500 high-school competition problems; (3) **AIME24** (Li et al.), 30 problems from the 2024 American Invitational Mathematics Examination; (4) **AMC23** (Li et al.), 40 problems from the 2023 American Mathematics Competition; and (5) **MMLU-STEM** (Hendrycks et al., 2020), a STEM-focused subset of MMLU for out-of-domain generalization evaluation. All evaluations follow the Qwen2.5-Math evaluation protocol (Yang et al., 2024) with temperature 0.6, top-p 0.95, and a length cap of 16,384 tokens. We report Pass@1 accuracy (Acc) and average response length (Tok).

Baselines. We compare against: (1) **Base**, the zero-shot model without any RL training; (2) **GRPO** (Shao et al., 2024), the standard binary-reward RLVR baseline; (3) **PEAR** (Huang et al., 2025), which penalizes thinking-phase entropy as a length proxy; (4) **DEER** (Yang et al., 2025b), **Dynasor** (Fu et al., 2024), and **LASER-L2048** (Liu et al., 2026b), representative length-control baselines; and (5) **DAPO** (Yu et al., 2025), a strong entropy-aware RLVR baseline. We also include two ablation variants: *phase-2 only* (no phase-1 warm-up) and *joint training* ($R_1 + R_2$ from initialization). All baselines and ERR+ use the same length cap of 16,384 tokens, temperature 0.6, top-p 0.95, and the evaluation protocol in Section 5.1,

Table 1: Pass@1 accuracy (%) and average response length (tokens, *Tok*) across model methods and five benchmarks. **Bold** indicates the best accuracy per column; underline the shortest average length.

Method	GSM8K		AIME24		AMC23		MATH-500		MMLU-STEM		Avg	
	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok
DeepSeek-R1-Distill-Qwen-1.5B												
Base	84.6	2076	26.7	12958	70.0	8063	82.1	4729	51.7	6266	63.0	6818
DEER	85.7	1553	26.7	10692	75.0	5556	83.8	3534	50.4	6391	64.3	5545
Dynasor	85.1	1916	26.7	11687	75.0	7287	82.0	4351	49.6	6670	63.7	6382
PEAR	85.3	1949	**33.3**	12260	67.5	8015	81.0	4848	50.7	6033	63.6	6621
GRPO	86.6	2281	30.0	12939	70.0	8336	82.6	5206	51.6	7440	64.2	7240
GRPO+ERR+	**88.6**	1324	**33.3**	10670	**77.5**	6288	84.8	3857	**58.0**	5109	**68.4**	5450
Laser-L2048	82.8	1850	26.7	10515	75.0	7503	80.3	4529	50.4	5632	63.0	6006
DAPO	85.4	2123	26.7	12956	70.0	8065	83.4	4432	49.9	6966	63.1	6908
DAPO+ERR+	87.1	1465	30.0	10542	75.0	5603	**85.2**	3296	54.2	4742	66.3	5130

ensuring a fair comparison.

5.2 Main Results

Table 1 reports results on DeepSeek-R1-Distill-Qwen-1.5B against a broad set of baselines. ERR+ achieves highest accuracy while reducing length, showing the objectives need not trade off. Existing methods either improve accuracy without shortening responses (GRPO) or reduce length at the cost of accuracy (PEAR), whereas ERR+ simultaneously improves both metrics. The gains transfer to DAPO as the base optimizer (DAPO+ERR+), confirming that the reward signals are not specific to GRPO. Table 1 and Appendix E use the same model, data, and evaluation setup but differ only in aggregation: Table 1 reports the best of four seeds (selected by validation accuracy with length monitored for outliers), while Appendix E reports the four-run mean and standard deviation ($66.10\% \pm 0.61\%$, 5794 ± 84 tokens).

5.3 Generalization Across Backbones

Table 2 extends evaluation to three additional backbones. ERR+ achieves the highest average accuracy on all three models while reducing response length compared to the GRPO baseline. PEAR achieves the shortest responses on some backbones but consistently degrades accuracy, particularly on harder benchmarks such as AIME24. ERR+ avoids this trade-off, confirming that the two-phase design generalizes across model families and scales.

5.4 Ablation Study

Table 3 presents results validating our design choices on the 1.5B backbone. **Phase-1 Necessity:** Removing phase-1 (phase-2 only) achieves the shortest response length on GSM8K (1244 tokens, 6% shorter than full ERR+), but its average length across all benchmarks (5465 tokens) is comparable to full ERR+ (5450 tokens) and suffers significant accuracy degradation: 5.5% average accuracy loss compared to the full model. This confirms that premature length compression without a stable reasoning structure sacrifices quality for brevity. **Phase-2 Necessity:** Omitting phase-2 (phase-1 only) maintains strong accuracy (66.8%) but fails to compress responses, resulting in 21.5% longer traces than full ERR+. **Sequential vs. Joint Training:** Joint training ($R_1 + R_2$ simultaneously) performs worst overall, showing 5.8% lower accuracy and 18.5% longer responses than full ERR+, which validates Theorem 1's prediction of early gradient conflict. **Summary:** The ablation study confirms that both phases are necessary: phase-1 establishes correct, decisive reasoning via entropy drops; phase-2 refines length via group-relative penalties. The sequential design is substantially superior to any individual component or joint optimization.

6 Ablation on Length Normalization in ERR

The ERR score (Eq. (3)) uses $\log(T_k + 1)$ as a length normalizer. We consider two failure modes: (i) no normalization rewards long traces for accumulating drops regardless of quality; (ii) linear normalization $(\sum r_t)/T_k$ imposes a constant per-token drop-density requirement and unfairly penalizes hard problems that genuinely need longer

Table 2: Pass@1 accuracy (%) and average response length (tokens, *Tok*) across three model backbones and five benchmarks. **Bold** indicates the best accuracy per backbone group; underline the shortest average length among trained methods.

Method	GSM8K		AIME24		AMC23		MATH-500		MMLU-STEM		Avg	
	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok
Qwen3-4B												
Base	94.5	2078	56.7	11654	80.0	7704	89.8	4814	90.3	2471	82.3	5744
GRPO	**95.4**	1969	53.3	11461	80.0	7615	90.2	4817	89.9	2573	81.8	5687
PEAR	93.9	1193	33.3	10081	**85.0**	5717	87.4	3636	**91.3**	1831	78.2	4492
GRPO+ERR+	94.6	1898	**60.0**	11002	**85.0**	6648	**91.0**	4214	90.3	2205	**84.2**	4752
Qwen3-8B												
Base	93.9	2226	56.7	11877	80.0	8348	87.0	5088	92.6	2586	82.0	6025
GRPO	94.6	2102	53.3	12576	80.5	8480	87.4	5631	92.6	2556	81.7	6269
PEAR	93.8	1639	36.7	11579	77.5	7281	85.6	4388	92.4	2206	77.2	5418
GRPO+ERR+	**96.0**	1798	**60.0**	11619	**82.5**	7918	**88.8**	4675	**92.9**	2243	**84.0**	5451
DeepSeek-R1-Distill-Qwen-7B												
Base	90.3	1270	50.0	10185	**92.5**	5971	92.2	4092	70.8	3540	79.2	5012
GRPO	**93.6**	1413	50.0	10934	90.0	7103	92.0	3915	71.2	4181	79.4	5509
PEAR	91.5	979	53.3	9903	90.0	5214	91.6	3213	73.8	2489	80.0	4360
GRPO+ERR+	92.7	1094	**60.0**	9843	90.0	5479	**92.8**	3289	**77.0**	2768	**82.5**	4495

Table 3: Ablation study of training phases on `DeepSeek-R1-Distill-Qwen-1.5B` across five datasets. Each row removes or modifies one design choice of ERR+. **Bold** indicates the best value per column.

Variant	GSM8K		AIME24		AMC23		MATH-500		MMLU-STEM		Avg	
	Acc	**Tok**	**Acc**	**Tok**	**Acc**	**Tok**	**Acc**	**Tok**	**Acc**	**Tok**	**Acc**	**Tok**
ERR+ (full)	**88.6**	1324	**33.3**	**10670**	**77.5**	**6288**	**84.8**	**3857**	**58.0**	5109	**68.4**	**5450**
w/o phase-1 (phase-2 only)	85.0	**1244**	26.7	10729	67.5	6933	84.0	4253	51.4	**4168**	62.9	5465
w/o phase-2 (phase-1 only)	88.1	1612	**33.3**	12807	72.5	8061	84.6	4752	55.6	5895	66.8	6625
Joint training ($R_1 + R_2$)	84.2	2005	23.3	12105	72.5	7245	83.0	4470	49.9	6470	62.6	6459

exploration. Logarithmic normalization provides a middle ground: the per-token density requirement diminishes as traces lengthen, rewarding milestone-like uncertainty resolution rather than verbosity, without enforcing a fixed density target.

Table 4: Effect of ERR length normalizer on average accuracy and token count across five benchmarks (`DeepSeek-R1-Distill-Qwen-1.5B`).

Normalizer	Avg Acc (%)	Avg Tok
T_k (linear)	48.6	2063
$\sqrt{T_k}$	53.3	3314
$\log(T_k + 1)$	**68.4**	**5450**

Table 4 compares the three choices on `DeepSeek-R1-Distill-Qwen-1.5B`. Linear normalization imposes the strictest per-token density requirement, which over-penalizes hard problems and causes significant accuracy degradation. $\sqrt{T_k}$ alleviates this but still shows clear accuracy loss relative to the logarithmic form. $\log(T_k + 1)$ gives the best trade-off, allowing phase 1 to build correct reasoning structure while phase 2 handles subsequent length compression. This ablation isolates the *Phase-1 normalizer*: the $\log(T_k + 1)$ form applies a milder sub-linear penalty, giving genuinely complex multi-step reasoning a grace region and avoiding harmful over-compression, whereas linear normalization yields shorter outputs only at a large accuracy cost. The final response efficiency of ERR+ is achieved by Phase-2 RRER *after* Phase 1 has established a stable reasoning structure, not by the Phase-1 normalizer alone.

Table 5: Pass@1 accuracy (%) and average response length (tokens, *Tok*) on MMLU-STEM and GPQA Diamond. **Bold** indicates the best accuracy; underline the shortest average length.

Method	MMLU-STEM		GPQA Diamond	
	Acc	Tok	Acc	Tok
DeepSeek-R1-Distill-Qwen-1.5B				
Base	51.7	6266	27.8	13875
GRPO	51.6	7440	29.3	14528
GRPO+ERR+	**58.0**	5109	**31.6**	10961

Table 6: Anti-reward-hacking check on GSM8K (`DeepSeek-R1-Distill-Qwen-1.5B`). *Up* and *Drop* are the per-trace totals of upward entropy movement and rewarded entropy drops, respectively. *Sign chg.* counts entropy-direction switches per trace. If ERR exploited artificial cycles, Up/Drop and Sign chg. should rise; both decrease instead.

Method	Acc	Tok	Up	Drop	Up/Drop	Sign chg.
GRPO	86.6	2281	507.2	506.9	1.0007	1012.98
ERR (Phase 1)	88.1	1612	281.0	280.8	1.0010	683.06
ERR+	88.6	1324	250.2	250.0	1.0011	609.77

6.1 Anti-Reward-Hacking Analysis

A potential concern is that ERR optimizes entropy drops as a proxy signal rather than reasoning quality directly, creating an incentive for the policy to generate artificially sharp entropy transitions (e.g. inflate-then-drop cycles, repetitive deterministic patterns) without genuinely improving reasoning. We address this with an intervention check: if such exploitation occurred, the ratio of upward entropy movement to rewarded drops (*Up/Drop*) and the per-trace sign-change count should inflate without corresponding accuracy gains. Table 6 shows that neither occurs. Up/Drop remains ≈ 1.0 throughout training and sign changes decrease steadily, indicating that rewarded drops track genuine commitment events rather than artificial cycling. This is consistent with recent evidence that high-entropy forking tokens carry disproportionate functional impact on reasoning outcomes (Wang et al., 2025; Agarwal et al., 2025): ERR rewards their resolution in correct traces only, without penalizing exploration or directly minimizing entropy.

7 Conclusion

We presented ERR+, a two-phase RLVR framework for large reasoning models. The method is grounded in the observation that correct reasoning traces exhibit more frequent and larger token-level entropy drops than incorrect ones. Phase 1 rewards entropy resolution in the thinking phase without constraining high-entropy exploration. Phase 2 introduces a difficulty-aware, outlier-robust length signal by scoring response length against co-generated peers. A formal gradient-conflict analysis establishes that sequential application is preferable to joint optimization in early training, and ablations confirm both empirical predictions. Experiments on GSM8K, AIME 2024, AMC23, MATH-500, and MMLU-STEM show consistent accuracy improvements over binary-reward and entropy-based baselines alongside meaningful length reductions.

Limitations

While ERR+ demonstrates consistent improvements in accuracy and conciseness across multiple reasoning benchmarks, several limitations warrant discussion and point to promising avenues for future work. Our optimization objectives target accuracy and response length, but human evaluators may also value attributes such as step-by-step explainability, pedagogical clarity, or stylistic consistency. Integrating multi-objective rewards that balance efficiency with human-centric qualities, potentially via preference learning or multi-task reinforcement learning, represents an important direction for deploying reasoning models in real-world applications.

Acknowledgments

This work was supported by New Generation Artificial Intelligence–National Science and Technology Major Project (2025ZD0124103) in collaboration with Shanghai Artificial Intelligence Laboratory. It is also supported by National Natural Science Foundation of China (under Projects No. 62477011 and 62377013), and the Fundamental Research Funds for the Central Universities.

References

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. The unreasonable effectiveness of entropy minimization in LLM reasoning. *Advances in Neural Information Processing Systems*, 38.

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

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *Preprint*, arXiv:2505.22617.

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

Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, Yi Ma, Aurick Qiao, Tajana Rosing, Ion Stoica, and Hao Zhang. 2024. Efficiently scaling llm reasoning with certaindex.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Xiaodong Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *ArXiv*, abs/2103.03874.

Minda Hu, Zexuan Qiu, Zenan Xu, Kun Li, Bo Zhou, and Irwin King. 2026. Conmax: Confidence-maximizing compression for efficient chain-of-thought reasoning. *ArXiv*, abs/2601.04973.

Chen Huang, Wei Lu, and Wenxuan Zhang. 2025. PEAR: Phase entropy aware reward for efficient reasoning. *arXiv preprint arXiv:2510.08026*.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Long Yu, Albert Qiaochu Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, Stanislas Polu, Hugging Face, and AI Mistral. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcák, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. 2026a. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *ArXiv*, abs/2601.05242.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2026b. Learn to reason efficiently with adaptive length-based reward shaping. In *International Conference on Learning Representations*, volume 2026, pages 43279–43308.

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

OpenAI. 2024. Learning to reason with LLMs. *OpenAI Blog*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junmei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300.

Hongze Tan and Jianfei Pan. 2025. GTPO and GRPO-S: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 77 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. *Advances in Neural Information Processing Systems*, 38.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao

Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. 2025b. Dynamic early exit in reasoning models. *ArXiv*, abs/2504.15895.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, and 8 others. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *Preprint*, arXiv:2504.05118.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. 2025a. Edge-GRPO: Entropy-driven GRPO with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*.

Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. 2025b. No free lunch: Rethinking internal feedback for LLM reasoning. *arXiv preprint arXiv:2506.17219*.

Yuxiang Zhang, Zhengxu Yu, Weihang Pan, Zhongming Jin, Qiang Fu, Deng Cai, Binbin Lin, and Jieping Ye. 2025c. Tokensqueeze: Performance-preserving compression for reasoning llms. *ArXiv*, abs/2511.13223.

A Proof of Theorem 1

All notation follows Section 4.3. We write $F(\theta) \triangleq \mathbb{E}_{y\sim\pi_\theta}[\mathrm{ERR}(y)]$, so that $\mathcal{L}_1(\theta) = -\mathbb{E}[R_1] = -R_b\, p_c - \lambda F(\theta)$, where p_c is the correctness probability (treated as approximately constant for small η). Hence $g_1 = \nabla_\theta \mathcal{L}_1 = -\lambda\, \nabla_\theta F(\theta)$.

Since $R_2 \propto \tanh(-\gamma z_i)$ is strictly decreasing in $L_i = |\mathcal{T}(y)| + (T - T_k)$, we have $\nabla_{L_i}\mathbb{E}[R_2] < 0$, so g_2 assigns negative log-probability weight to tokens that increase L_i. Among tokens in $\mathcal{T}(y)$, those in $\mathcal{T}_\rho(y)$ satisfy $r_t = 0$ and thus contribute nothing to R_1; they bear the highest ratio of length cost to reward benefit and are therefore the primary candidates for suppression under g_2.

Consider the update $\theta' = \theta - \eta g_2$. For each $t \in \mathcal{T}_\rho(y)$, let t' be the next committed position satisfying $r_{t'} > 0$. Under the autoregressive factorization, $\mathcal{H}_{t'-1}$ is a function of the generation context up to position $t'-1$, which depends on the token at t. Reducing the probability mass on position t under g_2 decreases $\mathbb{E}[\mathcal{H}_{t'-1}]$ by at least $\Delta\bar{\mathcal{H}}$ in expectation, and hence decreases $r_{t'} = \max(\mathcal{H}_{t'-1} - \mathcal{H}_{t'} - \epsilon, 0)$ by at least $\Delta\bar{\mathcal{H}}$. Summing over the $\rho(\theta)$ fraction of such positions and dividing by $\log(T_k+1)$:

$$F(\theta') \;\le\; F(\theta) - \frac{\eta\,\rho(\theta)\,\Delta\bar{\mathcal{H}}}{\log(T_k+1)}. \tag{12}$$

Taking $\eta \to 0$ in Eq. (12):

$$\nabla_\theta F(\theta)\cdot(-g_2) \;\le\; -\frac{\rho(\theta)\,\Delta\bar{\mathcal{H}}}{\log(T_k+1)}. \tag{13}$$

Since $g_1 = -\lambda\,\nabla_\theta F$, we have $\nabla_\theta F = -g_1/\lambda$, so:

$$\frac{g_1}{\lambda}\cdot g_2 \;\le\; -\frac{\rho(\theta)\,\Delta\bar{\mathcal{H}}}{\log(T_k+1)}, \tag{14}$$

which gives Eq. (9) with $C = \lambda\,\Delta\bar{\mathcal{H}}/\log(T_k+1) > 0$.

By a first-order Taylor expansion:

$$\begin{aligned}
&\mathcal{L}_1(\theta - \eta(g_1+g_2)) \\
&\quad= \mathcal{L}_1(\theta) - \eta\langle g_1,\, g_1+g_2\rangle + O(\eta^2) \\
&\quad= \mathcal{L}_1(\theta) - \eta\|g_1\|^2 - \eta\langle g_1, g_2\rangle + O(\eta^2) \\
&\quad\ge \mathcal{L}_1(\theta) - \eta\|g_1\|^2 + \eta C\rho(\theta) + O(\eta^2).
\end{aligned} \tag{15}$$

The single-objective step satisfies $\mathcal{L}_1(\theta - \eta g_1) = \mathcal{L}_1(\theta) - \eta\|g_1\|^2 + O(\eta^2)$, so subtracting the two expressions yields Eq. (10).

For the convergence claim: at θ_1^*, every high-entropy position in a correct trace is followed by a committed drop, so $\mathcal{T}_\rho(y) = \varnothing$ and $\rho(\theta_1^*) = 0$. The right-hand sides of Eqs. (9) and (10) both vanish. Furthermore, the tokens that g_2 would remove at θ_1^* are low-entropy, low-r_t spans between commitment points; by Eq. (12) with $\rho = 0$, these removals leave $F(\theta_1^*)$ unchanged, so applying R_2 from θ_1^* incurs no degradation to $\mathcal{L}_1$. □

Remark 1. *$C = \lambda\,\Delta\bar{\mathcal{H}}/\log(T_k+1)$ grows with the entropy surplus $\Delta\bar{\mathcal{H}}$ and decreases with thinking length T_k. Models with more volatile exploration (larger $\Delta\bar{\mathcal{H}}$) or shorter thinking budgets benefit most from the sequential schedule.*

B Hyperparameter Sensitivity Analysis

We perform a comprehensive sensitivity analysis for the two main hyperparameters controlling the magnitude of reward bonuses: λ (entropy relief weight in phase-1) and α (efficiency weight in phase-2). Figure 3 visualizes the accuracy–length trade-off across all five benchmarks as we vary $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

Phase-1 sensitivity (λ). Lower values ($\lambda = 0.1$) weaken the entropy-drop reward signal, resulting in reduced accuracy. As λ increases from 0.1 to 0.3, accuracy improves progressively. Beyond $\lambda = 0.3$, accuracy plateaus or degrades slightly, suggesting over-emphasis on entropy drops biases the model toward verbose justifications. The log-normalization in Eq. (3) provides natural regularization, but excessive weight ($\lambda \geq 0.4$) can overwhelm it.

Phase-2 sensitivity (α). Lower values ($\alpha = 0.1$) apply gentle length pressure but without meaningful compression. Increasing α to 0.2 and 0.3 progressively improves the accuracy–length trade-off. Beyond $\alpha = 0.3$, compression saturates and accuracy begins to degrade. The $\tanh$ saturation in Eq. (6) prevents runaway penalization at extreme z-scores, but excessive α shifts the gradient balance too heavily toward length reduction, sacrificing reasoning quality.

Cross-benchmark consistency. Across all five benchmarks (visualized in Figure 3), the default values $\lambda = 0.3$ and $\alpha = 0.3$ consistently yield the best accuracy–length trade-off. GSM8K (easier, grade-school math) shows less sensitivity to these parameters, while AIME24 and AMC23 (harder, competition math) benefit more from careful tuning. This suggests that on harder reasoning tasks, the distinction between encouraging entropy drops and penalizing length becomes more critical. The log-normalization and $\tanh$ saturation provide robustness across problem difficulty levels.

Robustness region. The region $\lambda \in [0.2, 0.4]$ and $\alpha \in [0.2, 0.4]$ represents a "sweet spot" where ERR+ maintains $> 87\%$ average accuracy while achieving $> 18\%$ length compression. Values outside this region show either accuracy degradation (too low: insufficient signal) or inefficient length compression (too high: conflicting objectives). Our default $\lambda = 0.3, \alpha = 0.3$ sits at the center of this region, providing balanced improvement.

C Prompt Format

In our experiments, when training models on the GSM8K dataset (Cobbe et al., 2021), we uniformly apply the following prompt to elicit step-by-step reasoning and format the final output:

```
Let's think step by step and output the final
answer after "####".
```

During the evaluation phase, we follow the official protocol of Qwen2.5-Math (https://github.com/QwenLM/Qwen2.5-Math). We employ their specific chat template and system instruction to guide the model's step-by-step reasoning and standardize the final answer delimiter. The exact prompt template is:

```
"qwen25-math-cot": (
    "<|im_start|>system\n
   Please reason step by step, and put your final
    answer within \\boxed{{}}.<|im_end|>\n"
    "<|im_start|>user\n{input}<|im_end|>\n"
    "<|im_start|>assistant\n",
    "{output}",
    "\n\n",
)
```

D Case Studies

We present a representative GSM8K problem traced across four checkpoints to illustrate the progressive effect of each training phase. Figure 4 compares the original model and GRPO training. The original model produces 2358 tokens but arrives at the wrong answer ($280), misinterpreting the haircut length. GRPO corrects this but increases token usage to 3220 tokens, confirming that binary correctness rewards provide no conciseness signal. Figure 5 shows that ERR further corrects reasoning structure while ERR+ additionally compresses the response from 8355 to 1950 tokens (77% reduction), committing directly to each arithmetic step without redundant elaboration. Taken together, the two figures confirm that Phase 1 establishes correct reasoning structure and Phase 2 safely compresses length without disrupting it.

E Statistical Performance Analysis of Different Methods

As Figure 6 shows, to measure the stability of different methods, we computed the average accuracy

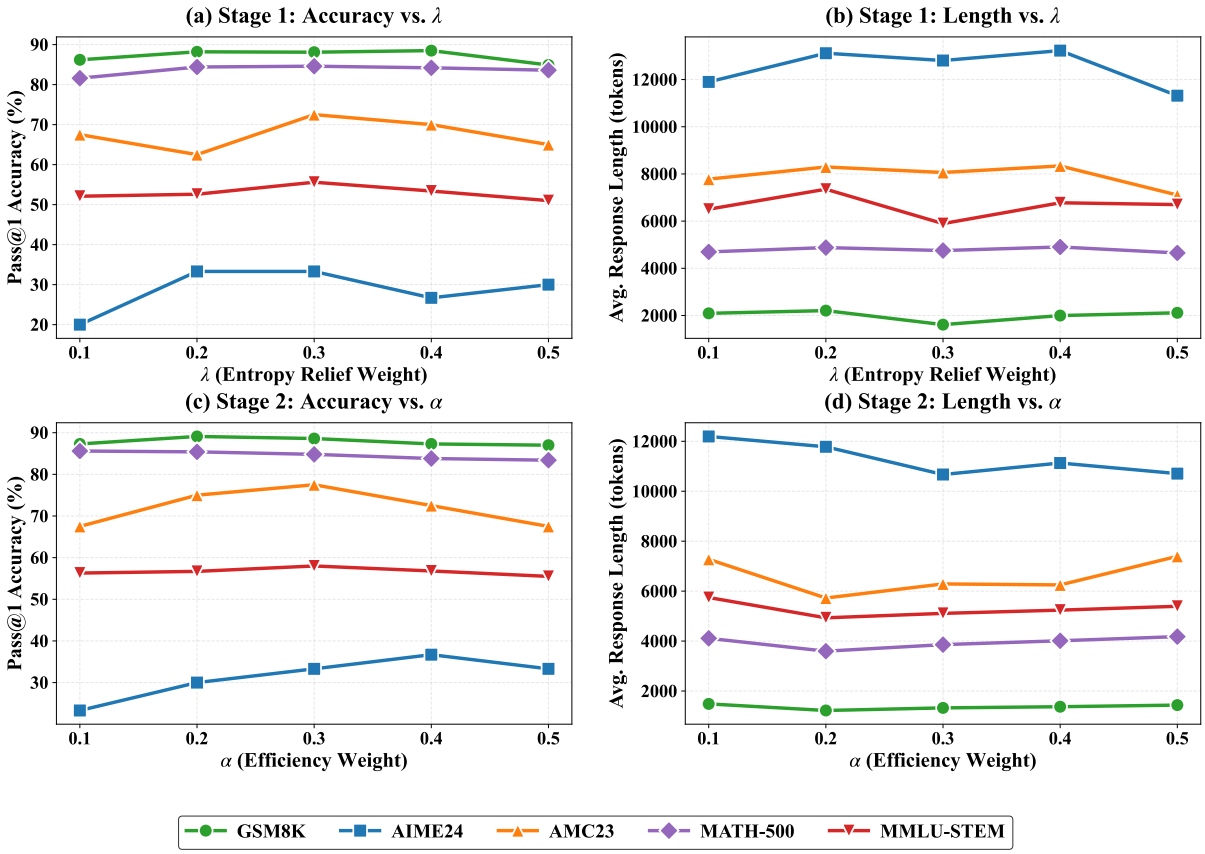


Figure 3: **Hyperparameter sensitivity analysis across five benchmarks.** We vary the entropy relief weight λ (Phase 1, rows) and efficiency weight α (Phase 2, columns) on `DeepSeek-R1-Distill-Qwen-1.5B`. Each heatmap shows accuracy (Acc) and token count (Tok). The default values $\lambda = 0.3$ and $\alpha = 0.3$ consistently yield the best accuracy–length trade-off across all five datasets, with marginal degradation toward the extremes.

and response length of the DeepSeek-R1-Distill-Qwen-1.5B (DS-R1-1.5B) model across five distinct datasets, based on four independent runs for each approach. From a statistical perspective, the ERR+ method demonstrates remarkable comprehensive advantages on the DS-R1-1.5B model. Its accuracy superiority has been verified by rigorous t-tests with p-values all below 0.05 in comparisons with other methods, confirming clear statistical significance and effectively ruling out the impact of random errors. Meanwhile, the method exhibits excellent stability and reliability: an accuracy standard deviation of 0.61% and variance of 0.3773 reflect high consistency across multiple independent evaluations, ensuring its stability fully meets practical application requirements. Furthermore, it delivers notable efficiency superiority: while maintaining the highest accuracy among all methods, it achieves the shortest response length of 5794 tokens, with a response length standard deviation of only 84 tokens that guarantees controllable fluctuations, thus perfectly balancing inference efficiency and result stability. In comparison with traditional RL methods such as GRPO and PEAR, the ERR+ method also presents distinct advantages—it not only mitigates the poor stability issue of GRPO and overcomes the insufficient accuracy limitation of PEAR, but also achieves a breakthrough improvement in output efficiency.

F Trace Structure Analysis: Why Phase 1 Enables Phase 2

Section 4.3 provides a formal gradient-conflict argument for the sequential design. Here we offer complementary empirical evidence at the trace level, explaining the mechanistic link between ERR and length efficiency.

Definitions. We call a position a *large drop* when the entropy relief $r_t > \tau$ (threshold $\tau = 0.5$ nats). A *low-relief span* is a contiguous segment of thinking tokens containing no large drop. Large drops are *commitment points* where the model resolves a

Question: May is getting her hair cut and colored. It costs $40 for the color and $30 per inch of haircut. If her hair started at 10 inches and ended at 8, how much did her cut and color cost?

Original Model

<think>
Okay, so I need to figure out how much May's hair cut and color cost. Let me start by breaking down the information given.
First, the color costs $40. That seems straightforward. So, the color part is settled. Now,
……(omitted here)
So, I'm confident that May's cut and color cost $280.
</think>
May's hair cut and color cost $40 for the color. The cost for the length of her hair is $30 per inch. Her hair started at 10 inches and ended at 8 inches, meaning the length after the cut is 8 inches.
The cost for the length is calculated as:
\[30 \, \text{dollars/inch} \times 8 \, \text{inches} = 240 \, \text{dollars} \]\n\nAdding the cost of the color to this amount gives:
\[240 \, \text{dollars} + 40 \, \text{dollars} = 280 \, \text{dollars} \].
Thus, the total cost of May's hair cut and color is \boxed{280} dollars.
Answer: 5

[2358 tokens in total]

GRPO Training

<think>
Okay, so May is getting her hair cut and colored, and I need to figure out how much that cost her. Let me break down the information given.\n\nFirst, the color costs $40. That seems straightforward. So, the color part is covered. Now,
……(omitted here)
May's cut and color cost her $100.
Final Answer The total cost of May's cut and color is \boxed{100}.\n
</think>
May is getting her hair cut and colored. The cost for the color is $40. The cost for the haircut is $30 per inch. Her hair started at 10 inches and ended at 8 inches, meaning the length of the cut is 2 inches (10 inches - 8 inches).
The cost for the haircut is calculated as:
\[2 \text{ inches} \times 30 \text{ dollars per inch} = 60 \text{ dollars} \]
Adding the cost of the color and the haircut:
\[60 \text{ dollars} + 40 \text{ dollars} = 100 \text{ dollars} \]
Therefore, the total cost of May's cut and color is \boxed{100}.

[3220 tokens in total]

Figure 4: **Original model vs. GRPO.** The original model (left, 2358 tokens) produces the wrong answer; GRPO (right, 3220 tokens) corrects it but at the cost of increased response length.

sub-problem; low-relief spans are the explanatory tokens between commitments.

Mechanism diagnostics. To further validate the claims that (i) phase 1 promotes uncertainty resolution without collapsing exploration and (ii) gradient conflict decreases after phase 1 convergence, we report four additional statistics at three checkpoints in Table 7: average thinking-phase entropy ($\bar{\mathcal{H}}$), the mid-exploration fraction ρ (high-entropy tokens not yet followed by a committed drop, as defined in Section 4.3), and the cosine similarity between g_1 and g_2.

Table 7: Mechanism diagnostics on GSM8K (`DeepSeek-R1-Distill-Qwen-1.5B`). $\bar{\mathcal{H}}$: mean thinking-phase entropy. ρ: mid-exploration fraction. Grad cos: cosine between g_1 and g_2.

Stage	Acc	Tok	$\bar{\mathcal{H}}$	ρ	Grad cos
Before ERR	84.6	2076	0.58	0.28	-0.45
After ERR (Ph.1)	88.1	1612	0.49	0.08	-0.05
After ERR+	88.6	1324	0.52	0.06	$+0.02$

Phase 1 doubles the large-drop frequency and reduces ρ from 0.28 to 0.08, confirming that high-entropy positions are increasingly resolved into committed drops rather than remaining as unresolved exploration. Average entropy does not collapse ($0.58 \rightarrow 0.49$), consistent with the claim that ERR preserves exploratory states. The gradient cosine rises from -0.45 to near zero after phase 1 and turns slightly positive after phase 2, empirically confirming Theorem 1's prediction that the conflict dissolves at phase 1 convergence.

Safe-prune accuracy: formal definition and protocol. Safe-prune accuracy is the percentage of *originally correct* samples whose extracted final answers remain correct when the model is forced to continue from a truncated trajectory with low-relief spans removed. It evaluates token-level continuation from the pruned state, not a restart of reasoning from scratch. Concretely: we filter the evaluation logs (`.jsonl`) for traces with `score[0]==True`; for each training stage we sample 200 correct GSM8K instances as the denominator. For each instance we remove low-relief spans (flat-entropy regions > 20 tokens between adjacent large-drop commitment points) and replace them with "…". The model is fed this truncated prefix in the original dialogue template and prompted for continuation (greedy decoding up to 128 tokens until

Question: May is getting her hair cut and colored. It costs $40 for the color and $30 per inch of haircut. If her hair started at 10 inches and ended at 8, how much did her cut and color cost?

ERR_Stage1

<think>
Okay, so I need to figure out how much May's hair cut and color cost her. Let me try to break this down step by step. First, the problem says that the color costs $40. That seems straightforward. So, the color part is $40. I don't think there's anything else mentioned about the color, like
……(omitted here)
I think the answer is $100.**Final Answer** The total cost of May's cut and color is \boxed{100}.
</think>
May is getting her hair cut and colored. The cost for the color is $40. The cost for the color is $30 per inch of haircut. Her hair started at 10 inches and ended at 8 inches, meaning the length of the cut is 10 - 8 = 2 inches.
The cost for the color applied to the 2-inch cut is calculated as follows:
\[2 \text{ inches} \times \$30/\text{inch} = \$60 \]\n\nAdding the cost of the color:
\[\$40 + \$60 = \$100 \]\n\nThus, the total cost of May's cut and color is \boxed{100}

[8355 tokens in total]

ERR+ Training

<think>
Okay, so let me try to figure out how much May's hair cut and color cost. Hmm, the problem says it costs $40 for the color and $30 per inch of haircut. Her hair started at 10 inches and ended at 8 inches. I need to find the total cost.First,
……(omitted here)
So, total is $100.**Final Answer**
The total cost of May's hair cut and color is \boxed{100}.
</think>
May's hair cut and color cost can be calculated as follows:
1. Determine the length of the hair after the cut:- She started with 10 inches and ended with 8 inches.\n - Length lost = 10 inches - 8 inches = 2 inches.
2. Calculate the cost of the haircut:\n - The cost per inch is $30.\n - Cost of haircut = 2 inches * $30/inch = $60.
3. Add the cost of the color:\n - The cost for the color is $40.\n - Total cost = Cost of color + Cost of haircut = $40 + $60 = $100.
Thus, the total cost of May's hair cut and color is \boxed{100}.

[1950 tokens in total]

Figure 5: **ERR vs. ERR+.** Both reach the correct answer; ERR+ (right, 1950 tokens) reduces response length by 77% compared to ERR (left, 8355 tokens).

`Final Answer:`). If the regenerated answer satisfies `math_equal(pred, gt)` it counts toward the numerator. Grad cos is the cosine similarity between accumulated gradients g_{ERR} and g_{RRER} over sampled trajectories, estimated over the full 1319-sample GSM8K test set.

Stability on small benchmarks. We further verify these metrics on AIME24 and AMC23 with `DeepSeek-R1-Distill-Qwen-1.5B` over four independent runs (T=0.6, top-p=0.95). Safe-prune is computed on originally correct logs (8 instances for AIME24, 28 for AMC23); Grad cos is estimated over the full benchmarks. Error bars are the unbiased sample standard deviation (Bessel's correction, $N-1$). The cross-checkpoint trends generalize beyond GSM8K: safe-prune accuracy consistently rises after ERR, and Grad cos shifts from large negative conflict toward near-zero alignment after Phase 1 and ERR+.

Structure-then-compress mechanism. Table 9 measures four statistics at three checkpoints on GSM8K (`DeepSeek-R1-Distill-Qwen-1.5B`): (i) *large-drop frequency* (large drops per 100 thinking tokens), (ii) *gap* (average tokens between adjacent large drops), (iii) *low-relief span %* (thinking tokens inside no-large-drop segments), and (iv) *safe-prune accuracy* (Pass@1 after removing all low-relief spans).

Phase 1 doubles the large-drop frequency (6.5→12.8) and raises safe-prune accuracy by 12 points (72.4→84.3%), confirming that traces are now organized around clearly identifiable commitment points. Phase 2 then reduces the low-relief

Table 8: Safe-prune accuracy and gradient cosine on small benchmarks over four runs (`DeepSeek-R1-Distill-Qwen-1.5B`).

Bench.	Stage	Safe-prune (%)	Grad cos
AIME24	Before ERR	65.63 ± 6.25	-0.78 ± 0.04
	After ERR (Ph.1)	75.00 ± 10.21	-0.26 ± 0.02
	After ERR+	78.13 ± 6.25	-0.02 ± 0.02
AMC23	Before ERR	75.90 ± 3.45	-0.56 ± 0.03
	After ERR (Ph.1)	82.13 ± 2.90	-0.12 ± 0.02
	After ERR+	83.90 ± 2.08	$+0.01 \pm 0.02$

Table 9: Trace structure statistics at three checkpoints on GSM8K. *Safe-prune acc.* measures Pass@1 after removing all low-relief spans.

Stage	Drop freq.	Gap	Low-relief %	Safe-prune acc.
Before ERR	6.5	15.4	32.5	72.4
After ERR	12.8	7.8	35.8	84.3
After ERR+	14.2	7.0	24.5	84.8

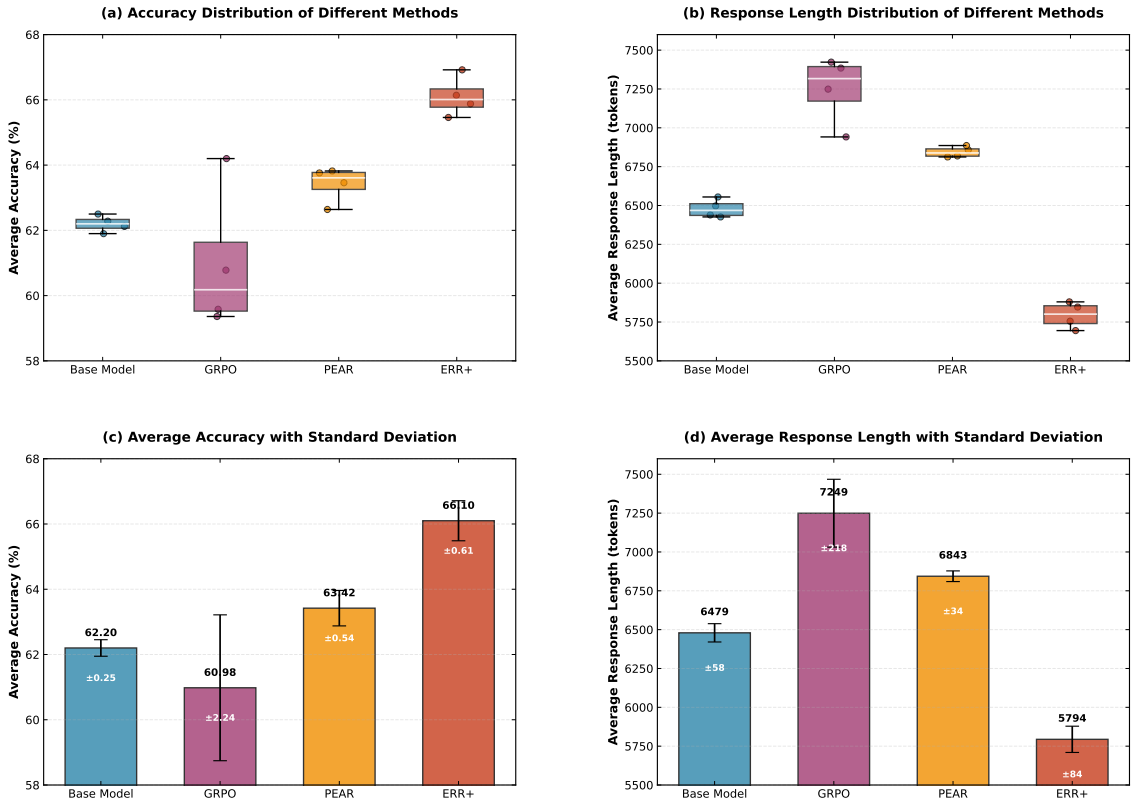


Figure 6: Performance and Stability Analysis of Different Methods for DeepSeek-R1-Distill-Qwen-1.5B

span percentage substantially (35.8→24.5%) without hurting safe-prune accuracy (84.3→84.8%), confirming that it compresses exactly the redundant material that Phase 1 has rendered safe to remove. This *structure-then-compress* mechanism directly explains why ERR and RRER form a cohesive two-phase design: Phase 1 must first consolidate reasoning into discrete commitments before Phase 2 can safely eliminate the low-value spans between them.

Table 10: Results of DAPO and DAPO+ERR+ on `DeepSeek-R1-Distill-Qwen-1.5B` across four benchmarks, reporting Pass@1 accuracy (%) and average response length (tokens). ERR+ reward signals consistently improve both metrics over the DAPO baseline, demonstrating generalization beyond GRPO.

Method	GSM8K		AIME24		AMC23		MATH-500	
	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok
DAPO	85.4	2123	26.7	12956	70.0	8065	83.4	4432
DAPO+ERR+	**87.1**	**1465**	**30.0**	**10542**	**75.0**	**5603**	**85.2**	**3296**

G Generalization to DAPO

To verify that ERR+ is not specific to GRPO, we integrate the two-phase reward design with DAPO (Yu et al., 2025) as the base optimizer, training on `DeepSeek-R1-Distill-Qwen-1.5B` under identical data and evaluation settings. Table 10 shows that DAPO+ERR+ consistently improves over DAPO alone in both accuracy and response length across four benchmarks, demonstrating that the ERR and RRER reward signals transfer across different policy optimization backbones.

H Generalization to Non-Math Reasoning

To test whether the entropy-drop pattern extends beyond arithmetic-style tasks, Table 5 reports results on GPQA Diamond (Rein et al., 2023), a graduate-level benchmark spanning biology, physics, and chemistry, alongside MMLU-STEM for reference. Crucially, GPQA Diamond and MMLU-STEM are evaluated *directly after GSM8K-only training, without any cross-domain fine-tuning*, making this a genuine cross-domain stress test. ERR+ improves both accuracy and response length over GRPO on both benchmarks, indicating that the model acquires a general reasoning-efficiency behavior rather than overfitting to GSM8K-specific format or arithmetic patterns, and that entropy relief is not specific to step-by-step arithmetic reasoning.