Learn to Explore In-Context via Reinforcement Learning

Futing Wang [*1 2 3] **Jianhao Yan** [*1 2 3] **Yun Luo** [3 †] **Ganqu Cui** [3] **Zhi Wang** [4 3] **Xiaoye Qu** [3] **Yue Zhang** [2 5]
Yu Cheng [6 †] **Tao Lin** [2 †]

Abstract

Achieving effective test-time scaling requires models to perform *in-context exploration* — the ability to generate, evaluate, and refine multiple reasoning hypotheses within a single trajectory. However, how to quantify and incentivize such exploration during reinforcement learning remains unclear. In this work, we propose a principled view of in-context exploration through *state coverage*, measuring the diversity of abstract reasoning states visited during generation. While directly optimizing state coverage is intractable, we show that trajectory length provides a simple and effective proxy for expanding exploration capacity. However, naively encouraging longer reasoning leads to degenerate behaviors such as repetition. To address this, we propose Length-Incentivized Non-redundant Exploration (LINE), a reward shaping approach that jointly incentivizes longer trajectories and penalizes redundant patterns. Experiments across multiple models and benchmarks show that LINE consistently improves reasoning performance and leads to more diverse reasoning trajectories, resulting in stronger test-time scaling behavior. On Qwen3-4B-Base, LINE improves average mathematical reasoning accuracy by 4.4 points over strong RL baselines, while also improving out-of-domain generalization.

1. Introduction

Scaling test-time computation, often conceptualized as enabling models to "think harder" before answering, has emerged as a powerful paradigm for breaking the performance ceiling of Large Language Models (LLMs) (Snell et al., 2024; Wu et al., 2024; Liu et al., 2025a). Broadly, Test-Time Scaling (TTS) strategies fall into two primary regimes: Parallel Scaling (Lightman et al., 2023; Snell et al., 2024; Liu et al., 2025a; Brown et al., 2024; Wang et al., 2022), which aggregates outputs from multiple independent samples, and Sequential Scaling (Guo et al., 2025; Jaech et al., 2024; Kumar et al., 2024), which prioritizes extended reasoning chains or iterative refinement. Unlike parallel scaling, sequential scaling relies on the model's ability to internally explore and refine intermediate reasoning states within a single trajectory. This intrinsic capability is critical for effective sequential test-time scaling, yet remains poorly understood.

A hallmark of this intrinsic reasoning is *in-context exploration* (Setlur et al., 2025), which is achieved through the generation, verification, and refinement of multiple hypotheses within a continuous context (Gandhi et al., 2024; Setlur et al., 2025). Consequently, in-context exploration enables models to effectively utilize additional test-time computation. As illustrated in Figure 1, effective in-context exploration enables the model to sequentially traverse diverse reasoning states to find the correct answer. Despite its importance,

In-context exploration remains poorly understood, and how to effectively incentivize it during reinforcement learning remains unclear.

In particular, there is no clear objective for encouraging models to explore diverse reasoning states within a single trajectory.

To address this problem, we propose a principled view of in-context exploration through *state coverage* (Auer, 2002), measuring the diversity of abstract reasoning states visited during generation. We extend this framework to test-time inference and employ state abstraction to render in-context states countable. This formulation connects in-context exploration to classical count-based exploration in reinforcement learning, providing a principled foundation for analyzing and improving in-context exploration behavior in LLMs. Figure 1 illustrates the conceptual distinction be-

[1]Zhejiang University [2]Westlake University [3]Shanghai AI Laboratory [4]Nanjing University [5]Institute of Advanced Technology, Westlake Institute for Advanced Study [6]The Chinese University of Hong Kong. Correspondence to: Yun Luo <luoyun1@pjlab.org.cn>, Yu Cheng <chengyu@cse.cuhk.edu.hk>, Tao Lin <lintao@westlake.edu.cn>.

The 3^{rd} *AI for Math Workshop* at the 43^{rd} International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.

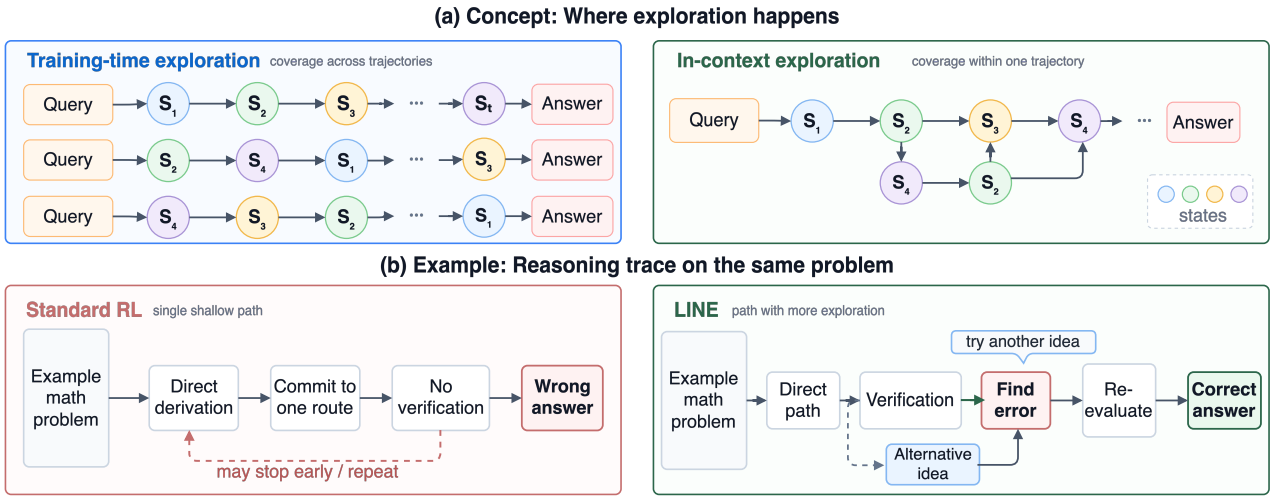


Figure 1. **The difference between in-context exploration and training exploration.** While training-time exploration covers states across sampled trajectories, in-context exploration visits diverse reasoning states within a single trajectory, as illustrated by the reasoning example below.

tween in-context exploration and traditional RL training exploration.

However, directly optimizing state coverage is intractable and prone to reward hacking. Instead, we identify a simple and practical proxy: *trajectory length*, which upper-bounds the number of distinct states that can be explored. While encouraging longer reasoning expands exploration capacity, it also introduces a critical failure mode: models may exploit the objective by generating repetitive, low-information tokens. To address this, we propose Length-Incentivized Non-redundant Exploration (LINE), a reward shaping approach that jointly incentivizes longer trajectories while penalizing redundant patterns. This design enables more efficient and stable exploration during training.

Experiments across multiple models (Qwen3 (Yang et al., 2025a) and LLaMA (Grattafiori et al., 2024)) and benchmarks demonstrate that LINE consistently improves reasoning performance and leads to stronger test-time scaling. On Qwen3-4B-Base, our method achieves a 4.4-point improvement on in-domain reasoning tasks and a 2.7-point gain on out-of-domain benchmarks.

Our key contributions are summarized as follows:

- **A principled view of in-context exploration.** We characterize in-context exploration as state coverage over abstract reasoning states. By employing state abstraction, we transform high-dimensional reasoning trajectories into quantifiable and countable in-context states, providing a foundation for analyzing test-time exploration.
- **A simple and effective training recipe.** We propose a simple yet effective training recipe LINE that explicitly incentivizes the model to expand in-context state coverage, enabling the model to explore during both training and inference.
- **Empirical validation and test-time scaling.** Comprehensive experiments across multiple model families and scales demonstrate that LINE consistently outperforms standard RL baselines. Furthermore, it achieves more effective test-time scaling by converting additional tokens into more in-context exploration.

2. Related Work

Scaling Test-Time Compute via Long CoT. Recent works have shown remarkable potential in scaling test-time computation (Snell et al., 2024) by training models to generate the long chains of thoughts (CoT) that enable strategic behaviors, including verification, self-correction (Kumar et al., 2024), etc. Scaling test-time computation significantly enhances the performance of Large Language Models (Guo et al., 2025; Team et al., 2025; Luo et al., 2025; Jaech et al., 2024) on various domains. Reinforcement Learning (RL), in particular, facilitates "natural" scaling of test-time computation through intrinsic adaptive exploration and strategy application (Guo et al., 2025; Team et al., 2025; Gandhi et al., 2025; Setlur et al., 2025; Yeo et al., 2025). These processes are mechanistically driven by negative gradients (Setlur et al., 2025; Zhu et al., 2025) or high entropy (Cui et al., 2025; Wang et al., 2025a; Cheng et al., 2025). S1 (Muennighoff et al., 2025) forces the model to think longer through explicitly forcing the generation of additional tokens to extrapolate thinking. However, we focus directly on increasing the computation usage during reinforcement training to enable in-context exploration to achieve test time scaling.

Length-Aware Reasoning. Recent studies have investigated length-aware reasoning, specifically aiming to miti-

gate overthinking and improve both efficiency and controllability. Several works (Jiang et al., 2025; Huang et al., 2025; Zhang et al., 2025b; Kang et al., 2025; Zhang et al., 2025c; Yu et al., 2025c; Yang et al., 2025c; Liu et al., 2025b) propose adaptively switching between long and short Chain-of-Thought (CoT) based on problem difficulty through training. Others (Zhang et al., 2025a; Ma et al., 2025) focus on controlling token usage during inference, while some studies address reasoning within a specific budget (Wen et al., 2025; Aggarwal & Welleck, 2025; Xu et al., 2025). In contrast to these efficiency-centric approaches that primarily aim to optimize token usage or curb overthinking, we posit that explicitly scaling up reasoning during RL not as a burden, but as an effective proxy to broaden the in-context exploration.

3. Background

In this section, we formulate LLM reasoning as a Markov Decision Process (MDP) and review the theoretical foundations of Count-Based Exploration in traditional reinforcement learning.

3.1. MDP Formulation of LLM Reasoning

LLM Reasoning as an MDP. We model the autoregressive generation process of a Large Language Model (LLM) as a deterministic MDP tuple $(\mathcal{S}, \mathcal{A}, \boldsymbol{\pi}, T)$, where

- **State Space ($\mathcal{S}$):** The state space $\mathcal{S}$ consists of all possible sequences of tokens from the vocabulary $\mathcal{V}$. Crucially, a specific state $s_t \in \mathcal{S}$ at time step t is the concatenation of the input query (prompt) x and the thought chain generated so far $y_{<t}$. Formally, $s_t = [x, y_1, \ldots, y_{t-1}]$.
- **Action Space ($\mathcal{A}$):** The action space is discrete and equivalent to the model's vocabulary $\mathcal{V}$, where an action $a_t \in \mathcal{V}$ corresponds to selecting the next token to append to the current sequence.
- **Transition Dynamics (T):** The transition is deterministic. Given the current state s_t and action a_t, the next state is uniquely determined by appending the token to the history: $s_{t+1} = s_t \oplus a_t$. The process terminates when a special end-of-sequence (EOS) token is generated.
- **Policy ($\boldsymbol{\pi}_\theta$):** The LLM functions as a parameterized policy $\boldsymbol{\pi}_\theta(a_t|s_t)$, which maps the current context s_t to a probability distribution over the vocabulary $\mathcal{V}$.

Due to the combinatorial nature of language, the size of $\mathcal{S}$ grows exponentially with sequence length ($|\mathcal{S}| \approx |\mathcal{V}|^L$), making the state space extremely vast and sparse. Additional details on the LLM reinforcement learning formulation are provided in Appendix D.1.

3.2. Theoretical Foundation: Incentivizing State Coverage via Count-Based Exploration

Count-based exploration is a fundamental strategy to address the exploration-exploitation tradeoff. Central to this approach is the **state visitation count** $N(s)$, which records the cumulative frequency of visiting a state s. These counts serve as a proxy for **state coverage** — the diversity of states visited during exploration. Guided by the principle of *Optimism in the Face of Uncertainty* (Auer et al., 2002), count-based methods augment the extrinsic reward with an exploration bonus $b(s)$, typically inversely proportional to the counts (e.g., $\propto 1/\sqrt{N(s)}$).

Theoretical Guarantees for Exploration. The optimality of count-based exploration is formally established in the Multi-Armed Bandit (MAB) setting. More detailed descriptions are shown in Appendix D.2. The Upper-Confidence Bound (UCB) algorithm selects action a_t by balancing reward estimates with visitation counts:

$$a_t = \arg\max_{a \in \mathcal{A}} \hat{R}_t(a) + \sqrt{\frac{2 \log t}{n_t(a)}}, \tag{1}$$

where $\hat{R}_t(a)$ and $n_t(a)$ is the reward estimate and visitation count for action a at time t. The bonus term $\sqrt{2 \log t / n_t(a)}$ decreases as the action is sampled more frequently.

Remark 3.1. *The count-based exploration principle provides a fundamental insight for LLM reasoning:* ***effective exploration requires maximizing state coverage****. It is crucial to note that in standard RL training, the visitation count $N(s)$ is aggregated across the* ***whole state space*** *over many training episodes. For in-context exploration, however, we must adapt this principle to quantify state diversity within a* single *reasoning trajectory.*

4. In-Context Exploration

While the standard MDP formulation focuses on optimizing policies during training, *in-context exploration* concerns how models utilize computation at test time. In particular, the objective shifts from maximizing global state coverage across episodes to maximizing the diversity of reasoning states within a single trajectory.

In this section, we develop a practical formulation of in-context exploration and derive a tractable training objective. Our key idea is to view exploration as *state coverage* within a trajectory, and to approximate this objective using a simple proxy based on trajectory length. This leads to a training recipe that encourages longer yet non-redundant reasoning trajectories.

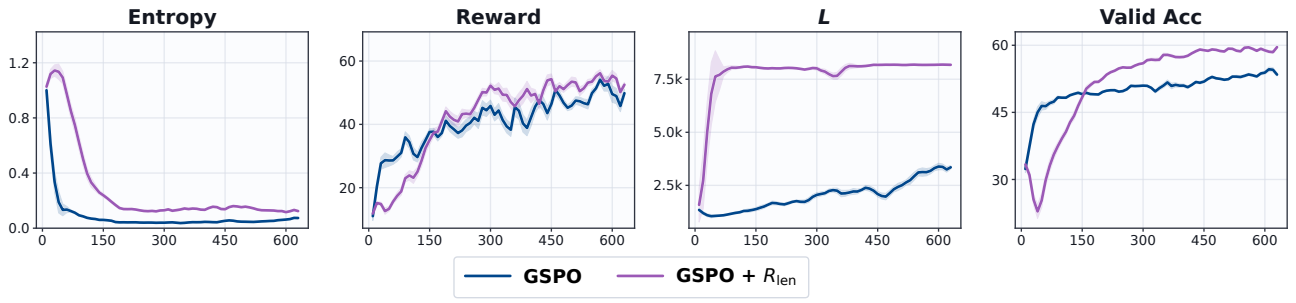


Figure 2. **Training dynamics comparing GSPO and GSPO with length-incentivized reward (R_{len}) on Qwen3-4B-Base.** We track entropy, training reward, trajectory length, and validation accuracy (8k) over training steps. Adding R_{len} significantly increases response length while also improving reward and validation performance, indicating that incentivizing longer reasoning can enhance downstream accuracy.

4.1. In-Context State Space

Defining in-context states. Given a single reasoning trajectory $\boldsymbol{\tau} = (y_1, y_2, \ldots, y_L)$, where L is the trajectory length, we define the in-context state space as the sequence of autoregressive states visited during generation:

$$S_{\text{IC}}(\boldsymbol{\tau}) = (s_1, s_2, \ldots, s_L)\,, \tag{2}$$

However, directly applying count-based exploration is problematic: since every state s_t contains a unique history prefix, no raw state is ever visited twice within a trajectory, rendering state coverage measurement trivial and meaningless.

State abstraction. To enable meaningful counting, we introduce a state abstraction function ϕ that maps raw states to abstract states:

$$\tilde{s}_t = \phi(s_t) = \phi(x_1, \ldots, y_1, \ldots, y_t),, \tag{3}$$

The abstraction ϕ can operate at different levels of granularity, such as lexical patterns, equations, or reasoning behaviors. Importantly, ϕ need not be lossless, as approximate abstractions are sufficient for preserving task-relevant structure and providing near-optimal performance (Abel et al., 2016).

In-context state coverage. Furthermore, we can now count in-context states to quantify the in-context state coverage. Following the count-based exploration principle from Section 3.2, we define the ***In-Context Distinct State Count*** $C_{\text{context}}(\boldsymbol{\tau})$ as the cardinality of unique context states visited within a trajectory:

$$C_{\text{context}}(\tau; \phi) = \left|\{\phi(s_t) \mid t = 1, \ldots, L\}\right|. \tag{4}$$

Section 3.2 posits that the optimal exploration strategy involves assigning a bonus reward proportional to the inverse of state visitation counts, i.e., $\propto 1/\sqrt{N(s)}$, which corresponds to maximizing $C_{\text{context}}(\tau; \phi)$.

However, directly optimizing C_{context} is impractical. The metric depends heavily on the choice of ϕ and is prone to reward hacking: models can inflate the count by generating superficially diverse but semantically meaningless patterns. We demonstrate this failure mode in Appendix F.4.

Instead, we leverage a key structural property: for any abstraction ϕ, state coverage is upper-bounded by trajectory length, i.e. $C_{\text{context}}(\tau; \phi) \leq L$. This suggests a simple but effective proxy: encouraging longer trajectories increases the capacity for exploration. This raises a central question: *Can we explicitly encourage longer reasoning as a proxy for improving in-context exploration?*

4.2. Thinking Longer Expands In-Context State Coverage

The Length-Incentivized Reward Based on the observation that state coverage is upper-bounded by trajectory length, we introduce a simple proxy objective that encourages longer reasoning trajectories. Concretely, we define a length-incentivized reward that penalizes premature termination on incorrect trajectories. For each sample i, we set a target length $L_{\text{target},i} = L_{\text{ref},i} + \Delta L$, where $L_{\text{ref},i}$ is the response length of the reference policy. The reward is applied only when the trajectory is incorrect, ensuring that correctness remains the primary objective. This design encourages the model to allocate additional computation to unresolved problems. The reward is formalized as:

$$R_{\text{len}} = \begin{cases} 0 & L \geq L_{\text{target}}, R_{\text{acc}} = 0 \\ -(L_{\text{target}} - L) & L < L_{\text{target}}, R_{\text{acc}} = 0 \\ 0 & R_{\text{acc}} = 1 \end{cases}. \tag{5}$$

This reward is integrated into the RLVR output accuracy reward, resulting in a combined reward defined as $R = R_{\text{acc}} + \eta R_{\text{len}}$.

Empirical Observations Figure 2 shows that adding R_{len} to GSPO increases response length while also improving training reward and validation accuracy. This suggests that expanding the reasoning horizon can be beneficial in practice. However, the key question is whether the additional tokens correspond to broader in-context exploration rather than mere verbosity.

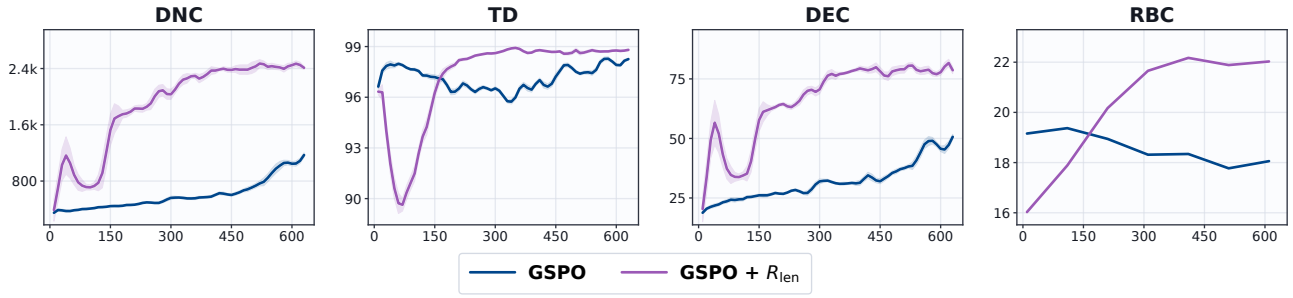


Figure 3. **In-context state coverage dynamics under GSPO with and without length incentive (R_{len}) on Qwen3-4B-Base.** We measure state coverage using multiple abstraction approximations, including distinct n-gram count (DNC), textual diversity (TD), distinct equation count (DEC), and reasoning behavior count (RBC). Adding R_{len} consistently increases all abstractions, indicating that longer trajectories lead to broader coverage of lexical, mathematical, and behavioral reasoning states.

To further understand this, we analyze in-context state coverage under multiple abstractions ϕ, capturing diversity at different levels:

- **Distinct n-grams count (DNC)**: measures lexical diversity via distinct n-grams.
- **Textual diversity (TD)**: measures dissimilarity across trajectory segments using BLEU-based metrics (Papineni et al., 2002; Hu et al., 2025).
- **Distinct equation count (DEC)**: measures diversity of mathematical formulations, reflecting alternative solution strategies (Hu et al., 2025).
- **Distinct reasoning behavior type count (RBC)**: captures diversity of cognitive-level behaviors. Following (Gandhi et al., 2025), we prompt Gemini-2.5-Flash-Thinking (Comanici et al., 2025) to annotate reasoning behaviors such as verification, backtracking, etc. Then we cluster all observed behaviors into distinct types. The clustered behavior types can be found in Appendix C.

As shown in Figure 3, increasing trajectory length consistently leads to higher state coverage across all abstraction levels. In particular, DNC, TD and DEC increase significantly, indicating greater lexical and mathematical diversity, while RBC also improves, suggesting more varied reasoning behavior type rather than repetitive patterns. These results indicate that longer trajectories expand the diversity of explored reasoning states, supporting the use of length as a practical proxy for in-context exploration.

4.3. Failure Mode: Degenerate Repetition

Although R_{len} expands the exploration horizon, optimizing it alone can introduce a failure mode: the autoregressive model may satisfy the length requirement by generating repetitive low-information sequences. In autoregressive generation, such degenerate loops provide a cheap way to increase trajectory length without exploring new states.

Empirically, we observe that models trained with R_{len} tend to saturate the maximum response length as shown in Figure 2. As shown in Figure 4, the distinct n-gram ratio decreases significantly, indicating that a large portion of the additional tokens arises from repeated local patterns rather than meaningful exploration.

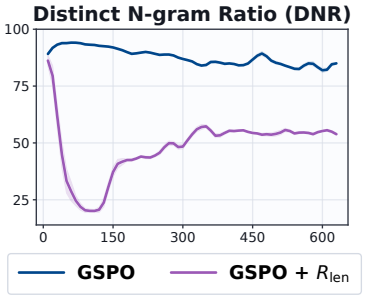


Figure 4. **Degenerate repetition under length-only training on Qwen3-4B-Base.** Although GSPO + R_{len} produces longer trajectories, the distinct n-gram ratio (DNR) drops substantially compared to GSPO.

This observation highlights a key limitation: *length alone increases the capacity for exploration, but does not ensure that this capacity is used effectively*. To address this, we require an additional mechanism that discourages the model from such a failure mode.

Redundancy penalty. We adopt a widely used repetition penalty mechanism (Yu et al., 2025b). Specifically, let $\mathcal{G}\tau$ denote the multiset of n-grams extracted from a generated trajectory τ, and let $\mathcal{N}\tau(g)$ denote the frequency of an n-gram $g \in \mathcal{G}\tau$. We define a binary indicator that identifies overly repeated n-grams exceeding a predefined threshold Θ:

$$R_{\text{red}} = -\mathbb{I}\left[\exists g \in \mathcal{G}_n(\tau) \text{ such that } N^{\tau}(g) > \Theta\right], \quad (6)$$

where Θ controls the repetition tolerance. We also explore semantic embedding-based redundancy penalties as an alternative to n-gram matching, but find them to be significantly less stable and more sensitive to implementation details as shown in Appendix F.7.

Final recipe. To prioritize solution accuracy while encouraging TTS, we shape the reward to favor correct responses

and longer reasoning trajectories, while penalizing redundant exploration:

$$R = R_{\text{acc}} + \eta \cdot R_{\text{len}} + \beta \cdot R_{\text{red}} \,. \tag{7}$$

Theoretical interpretation. This formulation conceptually aligns with the count-based exploration bonus $b_t \propto 1/\sqrt{N_t}$ (Section 3.2). At its core, the length term increases the maximum number of states that can be visited, while the redundancy penalty discourages repeatedly visiting similar states. Together, they approximate the effect of count-based exploration. Importantly, LINE does not directly optimize semantic state coverage. Instead, it provides a stable and practical approximation.

5. Experiments

In this section, we empirically examine whether the proposed formulation of in-context exploration can be converted into effective reasoning improvements. Specifically, we study the following research questions:

- RQ1: Can LINE serve as a reliable framework for improving reasoning?
- RQ2: Does LINE convert longer reasoning horizons into expanded state coverage?
- RQ3: Can LINE convert additional training compute into more effective test-time scaling?

5.1. Experimental Setup

Evaluation. We assess performance across eight reasoning benchmarks, categorized into in-domain mathematical tasks and out-of-distribution (OOD) general reasoning. The in-domain suite comprises AIME 2024/2025, AMC (Li et al., 2024), MATH-500 (Hendrycks et al., 2021), and OlympiadBench (He et al., 2024). The OOD evaluation includes ARC-c (Clark et al., 2018), GPQA-Diamond (denoted as GPQA* (Rein et al., 2024)), and MMLU-Pro (Wang et al., 2024). Regarding metrics, for benchmarks with limited sample sizes (AIME and AMC), we report the average accuracy over 32 independent runs (Avg@32). For all other benchmarks, we report Pass@1. All evaluations are conducted with a sampling temperature of 0.6, top-p=1.0, and a maximum response length of 32k tokens, which surpasses the training budget.

RLVR setups. Training is performed with a prompt batch size of 128, generating 8 rollouts per prompt. We update the policy using a mini-batch size of 32 and a learning rate of 1e-6. During the training phase, the sampling temperature is set to 1.0, with a maximum response length of 8,192 tokens. Experiments are implemented using the verl framework[1] on nodes equipped with 4×H100 GPUs, employing Math-Verify[2] for outcome-based reward calculation. Additional implementation details are provided in Appendix E. We default set $n = 10$, $\eta = 0.3/9000$, $\Theta = 10$, and $\beta = 0.6$ in reward. We discussed these hyperparameters in Appendix F.6.

Models and baselines. We designate Qwen3-4B-Base as our primary testbed. To verify the universality of our recipe across different training stages and model families, we also extend our evaluation to Qwen3-4B(non-thinking) and Llama-OctoThinker-3B-Base. The training datasets are selected to align with the respective models: DAPO-Math-17k (Yu et al., 2025a) for Qwen3-4B-Base, Polaris (An et al., 2025) for Qwen3-4B-Instruct, and DeepMath-5k (He et al., 2025; Tan et al., 2025) for Llama-OctoThinker. We primarily investigate GSPO as the core algorithm. Additionally, for Qwen3-4B-Base, we benchmark against standard GRPO and a GRPO variant with a high clip ratio to validate our recipe's effectiveness across different algorithms.

5.2. Results

RQ1: LINE improves reasoning performance across algorithms and models. As shown in Table 1, LINE improves average in-domain performance and OOD performance in most benchmarks. When applied to GSPO, LINE increases the in-domain average from 49.4% to 53.8%, with particularly large gains on AIME25 (6.2%). The OOD average also improves from 66.1% to 67.6%, suggesting that the benefit is not limited to the training distribution. We further verify that the improvements are robust across independent runs (Appendix F.3). Similar improvements (2%–3%) are observed on Qwen3-4B and Llama-OctoThinker-3B (Wang et al., 2025b) in Table 6, as detailed in Appendix F.1. We further verify that the improvement consistently holds across different model scales varying from 1.7B to 8B (Appendix F.2). These results indicate that LINE is a broadly effective recipe.

RQ2: LINE expands in-context state coverage. To evaluate whether LINE improves in-context state coverage, we analyze the training dynamics in Figure 5. Compared to the GSPO baseline, LINE produces longer trajectories and achieves a rapid increase in both distinct n-gram counts and distinct equation counts, indicating broader state coverage. More importantly, compared to length-only training, LINE achieves higher state coverage and better accuracy (as shown in Table 1) with substantially shorter rollouts. This suggests that the gains are not solely due to increased trajectory length but arise from more effective utilization of the reasoning horizon. These results demonstrate that LINE improves not only the *capacity* for exploration, but also its *efficiency*, converting additional computation into more diverse and informative reasoning trajectories.

[1] https://github.com/volcengine/verl

[2] https://github.com/huggingface/Math-Verify

Table 1. **In-Domain and Out-of-Domain evaluation performance based on Qwen3-4B-Base.** We compare standard RL baselines (GRPO, GRPO-Clip Higher, GSPO) with their counterparts augmented with LINE. Performance is reported on mathematical reasoning benchmarks (in-domain) and general reasoning benchmarks (out-of-domain). **Bold** and underline indicate the best and second-best results. Gains of our methods compared to corresponding baselines are marked in blue.

Model	In-Domain Performance					Out-of-Domain Performance			
	MATH	Olympiad	AMC	AIME 24/25	Avg.	ARC-c	GPQA	MMLU-Pro	Avg.
Qwen3-4B-Base	66.0	33.2	36.6	8.5/6.9	30.2	66.9	26.3	30.9	41.4
RL baselines									
GRPO	80.4	47.1	55.2	16.8/18.7	43.6	84.6	44.4	60.1	63.0
GRPO-Clip Higher	86.4	54.1	61.8	25.2/22.2	49.9	89.6	46.0	62.8	66.1
GSPO	85.2	51.7	62.7	26.7/20.5	49.4	88.4	**48.5**	61.5	66.1
Our recipes									
GRPO + LINE	85.0	49.9	60.5	22.9/16.4	46.9 (+3.3)	90.3	46.5	60.4	65.7 (+2.7)
GRPO-Clip Higher + LINE	**88.8**	54.1	65.5	30.4/24.5	52.6 (+2.7)	**91.5**	43.9	63.0	66.1 (+0.0)
GSPO + R_{len}	88.4	54.5	**66.8**	29.5/21.8	52.2 (+2.8)	91.0	40.9	**63.9**	65.2 (-0.9)
GSPO + LINE	88.4	**57.2**	66.2	**30.5/26.7**	**53.8** (+4.4)	91.4	47.5	63.8	**67.6** (+1.5)

RQ3: LINE improves Test-Time Scaling Figure 6 shows performance under increasing inference compute. Standard RL models (blue lines) quickly saturate or degrade beyond their training trajectory length, indicating limited ability to utilize additional computation.

In contrast, LINE (red lines) continues to improve, maintaining a clear upward trend even beyond the training horizon. The right panel shows that LINE shifts the length distribution toward longer responses, indicating active use of the extended token budget during test time. Combined with improved performance, this suggests that additional tokens are used for more effective exploration rather than redundant generation.

5.3. Discussion

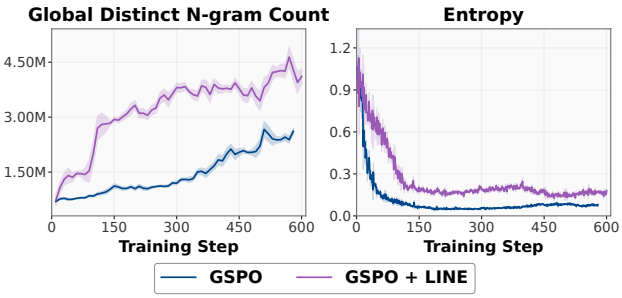


Figure 8. **Global exploration dynamics under GSPO and GSPO + LINE on Qwen3-4B-Base.**

Global diversity. We further analyze global exploration dynamics using Global Distinct N-gram Count and policy entropy. As shown in Figure 8, standard GSPO exhibits saturation and a rapid entropy drop, indicating premature convergence (mode collapse). In contrast, LINE maintains higher entropy and continuous growth in global state coverage. This sustained exploration prevents mode collapse and enables the model to discover rare, high-reward states that standard methods fail to explore.

The effect of ΔL. Increasing ΔL systematically extends the reasoning horizon and leads to consistent gains in validation accuracy, improving from 56.9% to over 63.4% as shown in Figure 7. At the same time, state coverage metrics (DNC and DEC) increase steadily, indicating that longer trajectories enable the model to explore a broader hypothesis space. Interestingly, this improvement exhibits a clear length scaling trend: allocating more computation via longer trajectories continues to improve performance. While a slight drop in the distinct n-gram ratio (DNR) is observed at larger ΔL, this effect is minor and does not offset the overall gains. Overall, these results show that extending the reasoning horizon is an effective and robust way to improve performance, with increased exploration dominating any marginal redundancy introduced at larger lengths.

The effect of LINE across difficulties. Following Sun et al. (2025), we evaluate LINE on AIME24 across difficulty levels. As shown in Table 2, LINE consistently improves accuracy on easy and medium problems, with the largest gains on the easier subsets, and also yields a modest improvement on hard problems. However, this benefit does not extend to the extremely hard subset, where the accuracy remains 0.0 across all three runs. These results suggest that LINE helps exploit near-solution exploration, but cannot overcome fundamental capability gaps. We further explore this question by studying the interaction with SFT (Appendix F.5).

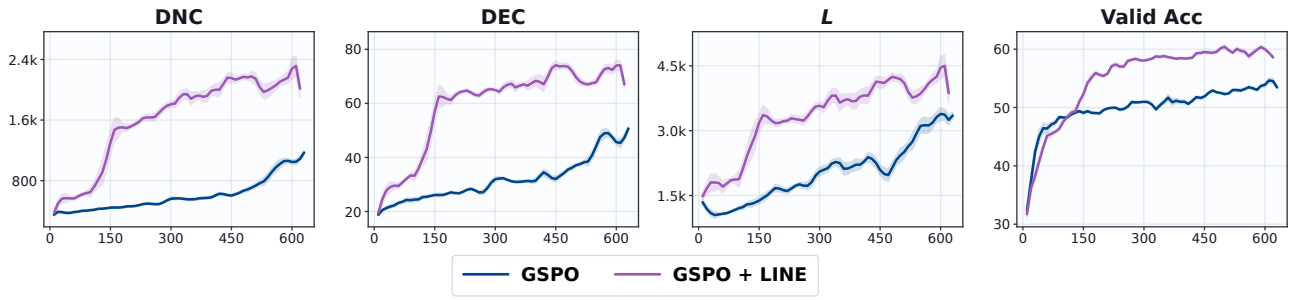


Figure 5. **Training dynamics comparing GSPO and GSPO + LINE on Qwen3-4B-Base.** We track trajectory length, validation accuracy (8k), distinct n-gram count (DNC), and distinct equation count (DEC) on the validation set. Compared to the GSPO, LINE achieves higher state coverage (DNC, DEC) and better accuracy.

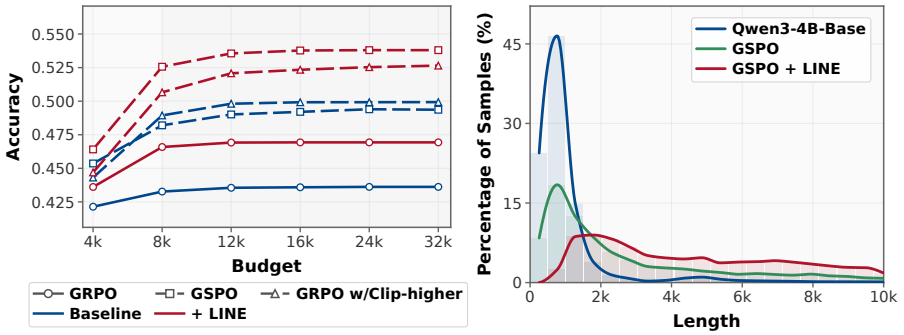


Figure 6. **Test-time scaling performance under controlled inference budgets on Qwen3-4B-Base.** All models generate responses up to 32k tokens, and performance is evaluated under different inference budgets by truncating generated trajectories. Accuracies are calculated over five in-domain mathematical benchmarks as shown in Table 1.

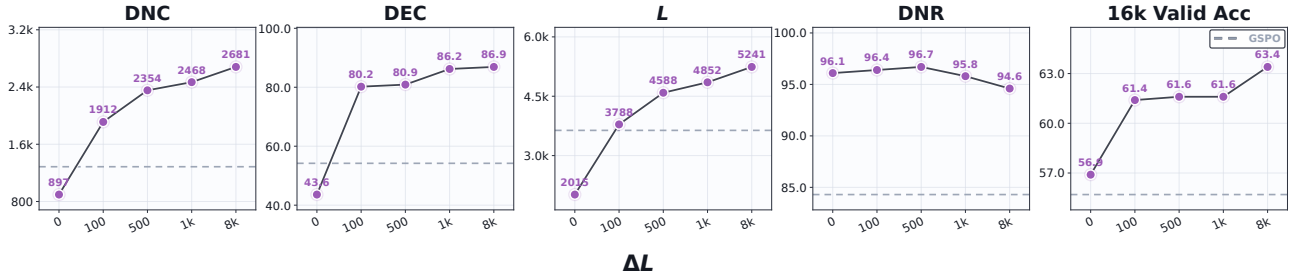


Figure 7. **The effect of ΔL on Qwen3-4B-Base.** As ΔL grows, the model produces longer trajectories and achieves higher state coverage (DNC, DEC) and improved accuracy.

Table 2. Performance across difficulty levels on AIME24. We report Avg@32 accuracy (%) and average response length (k tokens). All responses are generated up to a maximum length of 32k tokens.

	Accuracy (%)					Length (k tokens)				
Method	Easy	Medium	Hard	Ext. Hard	Overall	Easy	Medium	Hard	Ext. Hard	Overall
GSPO	84.4	37.5	7.3	0.0	26.7	1.9	8.3	13.2	7.6	9.3
GSPO + LINE	**95.8** (↑11.4)	**44.2** (↑6.7)	**10.2** (↑2.9)	0.0 (–)	**31.6** (↑4.9)	3.7	9.1	13.3	11.7	10.4

Table 3. **Continual scaling via curriculum training on Qwen3-4B-Base.** Stage 1 denotes the initial training stage, while Stage 2 further extends the length budget based on Stage 1.

Method	MATH	Oly.	AMC	AIME 24/25	Avg.
Qwen3-4B-Base	66.0	33.2	36.6	8.5/6.9	30.2
GSPO	85.2	51.7	62.7	26.7/20.5	49.4
Stage 1	88.4	57.2	66.2	30.5/26.7	53.8 (+4.4)
Stage 2	**89.4**	**59.0**	**71.7**	**33.4/27.9**	**56.3** (+2.5)

Continual scaling via curriculum training. We further extend the reasoning horizon through a second training stage with a larger length budget. As shown in Table 3, performance continues to improve, demonstrating that LINE can effectively convert additional training compute into reasoning accuracy. This result suggests a scalable training strat-

egy, where progressively increasing the reasoning horizon leads to continued performance gains.

6. Conclusion

We study how to improve test-time scaling in large language models through in-context exploration. We show that state coverage provides a useful perspective for understanding exploration, and that trajectory length is a practical proxy for expanding exploration capacity. Building on this insight, we propose LINE, which combines length incentives with redundancy control to encourage longer but non-redundant reasoning trajectories. Across multiple models and benchmarks, LINE consistently improves reasoning performance and enables stronger test-time scaling behavior, showing that effective scaling requires using additional computation to explore more diverse and informative reasoning paths.

References

Abel, D., Hershkowitz, D., and Littman, M. Near optimal behavior via approximate state abstraction. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2915–2923, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/abel16.html.

Agarwal, S., Ahmad, L., Ai, J., Altman, S., Applebaum, A., Arbus, E., Arora, R. K., Bai, Y., Baker, B., Bao, H., et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Aggarwal, P. and Welleck, S. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

An, C., Xie, Z., Li, X., Li, L., Zhang, J., Gong, S., Zhong, M., Xu, J., Qiu, X., Wang, M., and Kong, L. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL https://hkunlp.github.io/blog/2025/Polaris.

Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of machine learning research*, 3(Nov):397–422, 2002.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Cheng, D., Huang, S., Zhu, X., Dai, B., Zhao, W. X., Zhang, Z., and Wei, F. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Comanici, G., Bieber, E., Schaekermann, M., and Ice Pasupat, e. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

Cui, G., Zhang, Y., Chen, J., Yuan, L., Wang, Z., Zuo, Y., Li, H., Fan, Y., Chen, H., Chen, W., et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Feng, Y., Kempe, J., Zhang, C., Jain, P., and Hartshorn, A. What characterizes effective reasoning? revisiting length, review, and structure of cot. *arXiv preprint arXiv:2509.19284*, 2025.

Gandhi, K., Lee, D., Grand, G., Liu, M., Cheng, W., Sharma, A., and Goodman, N. D. Stream of search (sos): Learning to search in language. *arXiv preprint arXiv:2404.03683*, 2024.

Gandhi, K., Chakravarthy, A., Singh, A., Lile, N., and Goodman, N. D. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

He, C., Luo, R., Bai, Y., Hu, S., Thai, Z., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

He, Z., Liang, T., Xu, J., Liu, Q., Chen, X., Wang, Y., Song, L., Yu, D., Liang, Z., Wang, W., et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Hu, Z., Zhang, S., Li, Y., Yan, J., Hu, X., Cui, L., Qu, X., Chen, C., Cheng, Y., and Wang, Z. Diversity-incentivized exploration for versatile reasoning. *arXiv preprint arXiv:2509.26209*, 2025.

Huang, S., Wang, H., Zhong, W., Su, Z., Feng, J., Cao, B., and Fung, Y. R. Adactrl: Towards adaptive and controllable reasoning via difficulty-aware budgeting. *arXiv preprint arXiv:2505.18822*, 2025.

Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Jiang, L., Wu, X., Huang, S., Dong, Q., Chi, Z., Dong, L., Zhang, X., Lv, T., Cui, L., and Wei, F. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631*, 2025.

Kang, Y., Sun, X., Chen, L., and Zou, W. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24312–24320, 2025.

Kolter, J. Z. and Ng, A. Y. Near-Bayesian exploration in polynomial time. In *Proceedings of International Conference on Machine Learning*, pp. 513–520, 2009.

Kumar, A., Zhuang, V., Agarwal, R., Su, Y., Co-Reyes, J. D., Singh, A., Baumli, K., Iqbal, S., Bishop, C., Roelofs, R., et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

Li, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S., Rasul, K., Yu, L., Jiang, A. Q., Shen, Z., et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Liu, R., Gao, J., Zhao, J., Zhang, K., Li, X., Qi, B., Ouyang, W., and Zhou, B. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*, 2025a.

Liu, W., Zhou, R., Deng, Y., Huang, Y., Liu, J., Deng, Y., Zhang, Y., and He, J. Learn to reason efficiently with adaptive length-based reward shaping, 2025b. URL https://arxiv.org/abs/2505.15612.

Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Li, L. E., Popa, R. A., and Stoica, I. Deepscaler: Surpassing 01-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-01-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8, 2025.

Ma, X., Wan, G., Yu, R., Fang, G., and Wang, X. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*, 2025.

Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., and Hashimoto, T. B. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Setlur, A., Yang, M. Y., Snell, C., Greer, J., Wu, I., Smith, V., Simchowitz, M., and Kumar, A. e3: Learning to explore enables extrapolation of test-time compute for llms. *arXiv preprint arXiv:2506.09026*, 2025.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Strehl, A. L. and Littman, M. L. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Sun, Y., Zhou, G., Bai, H., Wang, H., Li, D., Dziri, N., and Song, D. Climbing the ladder of reasoning: What LLMs can—and still can't—solve after SFT? In *The 5th Workshop on Mathematical Reasoning and AI at NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=vrzApLEOdH.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Tan, Y., Wang, M., He, S., Liao, H., Zhao, C., Lu, Q., Liang, T., Zhao, J., and Liu, K. Bottom-up policy optimization: Your language model policy secretly contains internal policies. *arXiv preprint arXiv:2512.19673*, 2025.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Xi Chen, O., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Wang, S., Yu, L., Gao, C., Zheng, C., Liu, S., Lu, R., Dang, K., Chen, X., Yang, J., Zhang, Z., et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025a.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Wang, Z., Zhou, F., Li, X., and Liu, P. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*, 2025b.

Wen, H., Wu, X., Sun, Y., Zhang, F., Chen, L., Wang, J., Liu, Y., Liu, Y., Zhang, Y.-Q., and Li, Y. Budgetthinker: Empowering budget-aware llm reasoning with control tokens. *arXiv preprint arXiv:2508.17196*, 2025.

Wu, Y., Sun, Z., Li, S., Welleck, S., and Yang, Y. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.

Xu, Y., Dong, H., Wang, L., Sahoo, D., Li, J., and Xiong, C. Scalable chain of thoughts via elastic reasoning. *arXiv preprint arXiv:2505.05315*, 2025.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Yang, S., Tong, Y., Niu, X., Neubig, G., and Yue, X. Demystifying long chain-of-thought reasoning. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=OLodUbcWjB.

Yang, W., Ma, S., Lin, Y., and Wei, F. Towards thinking-optimal scaling of test-time compute for llm reasoning. *arXiv preprint arXiv:2502.18080*, 2025c.

Yeo, E., Tong, Y., Niu, M., Neubig, G., and Yue, X. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025a.

Yu, T., Ji, B., Wang, S., Yao, S., Wang, Z., Cui, G., Yuan, L., Ding, N., Yao, Y., Liu, Z., et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025b.

Yu, Z., Xu, T., Jin, D., Sankararaman, K. A., He, Y., Zhou, W., Zeng, Z., Helenowski, E., Zhu, C., Wang, S., et al. Think smarter not harder: Adaptive reasoning with inference aware optimization. *arXiv preprint arXiv:2501.17974*, 2025c.

Zhang, J., Dong, R., Wang, H., Ning, X., Geng, H., Li, P., He, X., Bai, Y., Malik, J., Gupta, S., et al. Alphaone: Reasoning models thinking slow and fast at test time. *arXiv preprint arXiv:2505.24863*, 2025a.

Zhang, J., Lin, N., Hou, L., Feng, L., and Li, J. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*, 2025b.

Zhang, S., Wu, J., Chen, J., Zhang, C., Lou, X., Zhou, W., Zhou, S., Wang, C., and Wang, J. Othink-r1: Intrinsic fast/slow thinking mode switching for over-reasoning mitigation. *arXiv preprint arXiv:2506.02397*, 2025c.

Zheng, C., Liu, S., Li, M., Chen, X.-H., Yu, B., Gao, C., Dang, K., Liu, Y., Men, R., Yang, A., et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Zhu, X., Xia, M., Wei, Z., Chen, W.-L., Chen, D., and Meng, Y. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.

A. Limitations

Our work has several limitations. First, our exploration is conducted within a limited scaling regime, with training up to 8k tokens and models up to 8B parameters, and it is interesting to extend LINE to larger scales. Second, while LINE improves exploration during RL, it does not directly address the acquisition of new reasoning capabilities. Preliminary experiments with supervised fine-tuning (SFT) (Appendix F.5) suggest potential complementary effects, but a systematic integration remains open. Third, our redundancy control relies on n-gram-based design, which cannot fully capture semantic repetition. Although we explore embedding-based alternatives (Appendix F.7), they are less stable and sensitive to implementation details. Finally, our understanding of how LINE affects reasoning processes is still limited; preliminary qualitative analyses (Appendix G) suggest possible structural differences, but are not yet conclusive. Due to limited academic resources, our experiments are constrained in scale, and further investigation at larger scales is an important direction for future work.

B. Broader Impact

This work studies how to improve the test-time scaling behavior of reasoning language models by encouraging longer and less redundant in-context exploration. The potential positive impact is that such methods may improve the reliability and effectiveness of models on complex reasoning tasks, especially in mathematical and scientific problem solving, where additional computation can help models verify intermediate steps and recover from mistakes. However, stronger reasoning capabilities may also amplify the potential misuse of language models, for example by improving their ability to solve tasks that support harmful automation or by increasing user overreliance on generated reasoning traces. In addition, methods that rely on longer reasoning trajectories may increase computational cost and energy consumption, potentially widening the accessibility gap between well-resourced and resource-constrained research groups. We encourage future work to study efficiency, safety evaluation, and responsible release practices alongside improvements in reasoning performance.

C. Reasoning Behavior Taxonomy

Table 4 summarizes the reasoning behavior categories extracted from reasoning trajectories using Gemini-2.5-Flash-Thinking. Each category consolidates semantically similar raw annotations into a compact and interpretable behavior label.

Table 4. Reasoning behavior taxonomy. Each category is defined by a concise operational description and illustrated with example raw tags.

Behavior Category	Operational Definition	Example Tags
Backtracking	Revising an approach after detecting an error, contradiction, or dead end.	Backtracking; Strategy Shift; Revising Approach.
Verification	Checking results, assumptions, constraints, or answer consistency.	Verification; Constraint Check; Plausibility Check.
Subgoal Setting	Breaking a complex problem into manageable intermediate steps.	Subgoal Setting; Problem Decomposition; Structured Analysis.
Enumeration	Systematically considering multiple cases, possibilities, or hypotheses.	Enumeration; Reasoning by Cases; Exploring Alternatives.
Abstraction and Generalization	Extracting general principles or structures from specific instances.	Abstraction; Generalization; Pattern Generalization.
Simplification and Reduction	Reducing complexity through substitution, smaller cases, or removing details.	Simplification; Variable Substitution; Simpler Cases.
Problem Reframing and Transformation	Changing the representation or interpretation of the problem.	Problem Reframing; Representation Change; Transformation.
Constraint and Boundary Analysis	Reasoning about constraints, domains, edge cases, or feasible ranges.	Constraint Analysis; Boundary Analysis; Range Analysis.

Continued on next page

Table 4. Reasoning behavior taxonomy continued.

Behavior Category	Operational Definition	Example Tags
Logical Deduction and Derivation	Deriving conclusions through step-by-step logical or algebraic reasoning.	Logical Deduction; Algebraic Manipulation; Derivation.
Strategic Planning and Method Selection	Choosing or planning an appropriate solution strategy.	Strategic Planning; Method Selection; Strategic Choice.
Hypothesis Generation and Heuristic Exploration	Proposing tentative ideas or heuristics under uncertainty.	Hypothesis Generation; Heuristic Search; Trial and Error.
Pattern Recognition and Analogy	Identifying recurring structures, symmetries, or analogies.	Pattern Recognition; Analogy; Symmetry Recognition.
Mathematical Knowledge Application	Applying known formulas, theorems, properties, or domain knowledge.	Knowledge Application; Formula Retrieval; Theorem Recall.
Tool Use and Computational Integration	Using code, calculators, algorithms, or external tools to support reasoning.	Tool Use; Computational Modeling; Algorithmic Thinking.
Definition Clarification and Formalization	Clarifying terms, variables, conditions, or formal notation.	Definition Clarification; Formalization; Variable Definition.
Self-Correction and Metacognitive Monitoring	Monitoring uncertainty, errors, assumptions, or reasoning progress.	Self-Correction; Metacognition; Error Detection.
Unit Precision and Calculation Structuring	Managing units, precision, intermediate results, or computation structure.	Unit Consistency; Precision Awareness; Structured Calculation.

D. Background

D.1. Reinforcement Learning for LLM Reasoning

Reinforcement Learning for LLM Reasoning. To optimize the policy $\boldsymbol{\pi}_\theta$ to maximize the expected return $J(\pi_\theta) = \mathbb{E}[\sum r_t]$, the optimization is typically performed via **Policy Gradient** (Sutton et al., 1999):

$$\nabla_\theta J(\boldsymbol{\pi}_\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T} \nabla_\theta \log \boldsymbol{\pi}_\theta(a_t \mid s_t)\, A^{\boldsymbol{\pi}}(s_t, a_t) \right],$$

where $A^{\boldsymbol{\pi}}$ represents the advantage of taking action a_t. GRPO and GSPO are instantiations of this framework.

Group Relative Policy Optimization (GRPO). GRPO (Shao et al., 2024) performs optimization at the token level without a value function. It utilizes the standard per-token probability ratio:

$$\rho_{i,t}(\theta) = \frac{\boldsymbol{\pi}_\theta(y_{i,t}|x, y_{i,<t})}{\boldsymbol{\pi}_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})}, \tag{8}$$

where $y_{i,t}$ is the t-th token of the i-th sequence. The advantage is computed via group normalization:

$$\hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^G)}{\text{std}(\{R_j\}_{j=1}^G)}. \tag{9}$$

The objective averages the PPO-clip loss (Schulman et al., 2017) over all tokens in the generated sequences:

$$\begin{aligned} J^{\text{GRPO}}(\theta) = \mathbb{E}_{x,\{y_i\}\sim\boldsymbol{\pi}_{\theta_{\text{old}}}} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min \Big(\rho_{i,t}(\theta)\hat{A}_i, \\ \text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i \Big) \Bigg]. \end{aligned} \tag{10}$$

Group Sequence Policy Optimization (GSPO). GSPO (Zheng et al., 2025) elevates optimization to the sequence level using length-normalized importance ratios:

$$\rho_i(\theta) = \left(\prod_{t=1}^{|y_i|} \frac{\boldsymbol{\pi}_\theta(y_{i,t}|x, y_{i,<t})}{\boldsymbol{\pi}_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})} \right)^{1/|y_i|}, \tag{11}$$

where the $1/|y_i|$ exponent reduces variance from varying sequence lengths. Using the same group-based advantage $\hat{A}_i$ as GRPO, the objective is computed once per sequence.

D.2. Count-Based Exploration Theoretical Foundation

In this section, we provide additional theoretical details that complement the formulation in Section 3. The exploration-exploitation tradeoff is a long-standing challenge in RL literature. This becomes more pronounced in LLM reasoning tasks due to the vast state-action space with reward sparsity, where existing methods often suffer from deficient exploration and poor sample efficiency. A classic, theoretically justified exploration principle is *optimism in the face of uncertainty*, which augments the reward estimates of less-visited states/actions with an exploration bonus proportional to their uncertainty. In the minimal multi-armed bandit (MAB) setting, the well-known upper-confidence bound (UCB) algorithm (Auer, 2002) chooses the action a_t according to:

$$a_t = \arg\max_{a \in \mathcal{A}} \hat{R}_t(a) + \sqrt{\frac{2 \log t}{n_t(a)}}, \tag{12}$$

where $\hat{R}_t(a)$ and $n_t(a)$ is the reward estimate and visitation count for action a at time t. Theorem D.1 establishes the theoretical optimality of this *count-based exploration* strategy, demonstrating that it yields a cumulative regret that grows only logarithmically over time.

Theorem D.1 (Optimality of Count-based Exploration (Auer, 2002)). *In an MAB setting, let $L(T) = \mathbb{E}[\sum_{t=1}^{T}(R^* - R(a_t))]$ denote the total regret over T steps, where $R(a) = \mathbb{E}_{\mathcal{R}^a}[R]$ is the expected reward for any action a and R^* is the reward for the optimal action. Let $\Delta_a = R^* - R(a)$ denote the reward gap between action a and the optimum. For any bandit algorithm, the asymptotic total regret is at least logarithmic in the number of steps:*

$$\lim_{T \to \infty} L(T) \geq \log T \cdot \sum_{a|\Delta_a > 0} \frac{\Delta_a}{KL(\mathcal{R}^a || \mathcal{R}^{a^*})}. \tag{13}$$

The UCB algorithm in Eq. 12 achieves logarithmic asymptotic total regret:

$$\lim_{T \to \infty} L(T) \leq 8 \log T \cdot \sum_{a|\Delta_a > 0} \Delta_a. \tag{14}$$

Subsequent studies (Bellemare et al., 2016; Tang et al., 2017) extend this principle to MDPs by counting state-action pairs $n(s, a)$ and adding a bonus reward to encourage exploring less-visited pairs as

$$R'(s, a) = R(s, a) + \frac{\beta}{\sqrt{n(s, a)}}, \tag{15}$$

which is shown to be formally near-optimal within the probably approximately correct (PAC-MDP) framework (Strehl & Littman, 2008; Kolter & Ng, 2009).

E. Experimental Details

E.1. Template Prompt

We adopt the following template for all experiments involving Qwen models, building upon the Qwen3 template for Qwen3-4B-Base and the Qwen-Nothinking template for Qwen3.

Qwen3 Template

<|im_start|>user
{problem} Let's think step by step and output the final answer within \boxed{}.
<|im_end|>
<|im_start|>assistant

Qwen3-NoThinking Template

<|im_start|>user
{problem} Let's think step by step and output the final answer within \boxed{}.
<|im_end|>
<|im_start|>assistant
<think>

</think>

For training the Llama-OctoThinker models, we adopt the original prompt in (Wang et al., 2025b) to ensure performance.

OctoThinker Template

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. User: You must put your answer inside \boxed{} and Your final answer will be extracted automatically by the \boxed{} tag.
{problem}
Assistant:

E.2. Implementation of RL

In this section, we describe the RL training setup in detail. We implement GRPO and other baseline algorithms using the Verl framework. Across all algorithms and model variants, we adopt a unified set of hyperparameters, as reported in Table 5, and do not employ entropy regularization or KL-based losses.

Table 5. Reinforcement learning training hyperparameters used across all experiments.

Hyperparameter	**Value**
Optimizer	AdamW
Policy learning rate	$1e^{-6}$
Training batch size	128 prompts
Samples per prompt	8
Mini-batch size	32 prompts
Policy updates per rollout	16
Max prompt length	1024 tokens
Max response length	8192
Rollout temperature	1.0

E.3. Compute resources

To support reproducibility, each main training run used 4 × H100 GPUs with 140 GB GPU memory per GPU. The GSPO baseline required approximately 72 hours, corresponding to 288 GPU-hours. GSPO + (R_{len}) required approximately 240 hours, corresponding to 960 GPU-hours, while GSPO + LINE required approximately 100 hours, corresponding to 400 GPU-hours. This indicates that LINE substantially reduces the computational overhead compared with the length-only reward while still enabling effective performance improvements.

Table 6. **Evaluation on Qwen3-4B and Llama-OctoThinker-3B.** AIME 24/25 denotes AIME 2024 / AIME 2025, reported as x/y. **Bold** indicates the best result within each model block. Gains of our methods over the corresponding GSPO baselines are marked in blue.

Model	In-Domain Performance					Out-of-Domain Performance			
	MATH	Olympiad	AMC	AIME 24/25	Avg.	ARC-c	GPQA	MMLU-Pro	Avg.
Qwen3-4B									
Base	82.8	51.9	60.4	24.2/19.4	47.7	74.7	37.9	59.4	57.3
GSPO	94.0	**68.1**	82.0	54.2/42.5	68.2	87.5	**52.5**	56.2	65.4
GSPO + LINE	**94.4**	67.0	**85.3**	**57.7/46.4**	**70.2** (+2.0)	**92.1**	52.0	**70.9**	**71.7** (+6.3)
Llama-OctoThinker-3B									
Base	23.4	9.0	10.4	1.1/0.6	8.9	37.6	20.7	11.5	23.3
GSPO	55.8	23.1	28.2	3.8/2.3	22.6	**66.2**	24.7	27.8	**39.6**
GSPO + LINE	**60.8**	**28.1**	**30.3**	**4.5/4.4**	**25.6** (+3.0)	52.0	**27.8**	**38.0**	39.3 (-0.3)

F. Additional Results

F.1. Extension to more models

To examine whether LINE generalizes beyond the primary Qwen3-4B-Base setting, we further evaluate it on two additional model settings: Qwen3-4B and Llama-OctoThinker-3B. These models differ from the main testbed in both model family and training stage. Qwen3-4B represents a post-trained non-thinking model, while Llama-OctoThinker-3B is a reasoning-oriented model with a different pretraining and mid-training trajectory. This allows us to test whether LINE is tied to a specific initialization or can serve as a more general recipe for activating longer in-context exploration.

As shown in Table 6, LINE consistently improves in-domain mathematical reasoning. On Qwen3-4B, applying LINE on top of GSPO improves the in-domain average from 68.2 to 70.2, with particularly clear gains on AMC and AIME 2025. The out-of-domain average also increases substantially from 65.4 to 71.7, mainly driven by improvements on ARC-c and MMLU-Pro. On Llama-OctoThinker-3B, LINE improves the in-domain average from 22.6 to 25.6, suggesting that the recipe remains effective even when the base model has a different reasoning prior.

The OOD behavior on Llama-OctoThinker-3B is more mixed: although GPQA and MMLU-Pro improve, ARC-c decreases, leading to a slight drop in the OOD average. This suggests that LINE reliably improves mathematical reasoning across model families, while its effect on general-domain transfer can depend on the base model and data distribution. Overall, these results support the view that LINE is not merely a Qwen3-4B-Base-specific tuning trick, but a broadly applicable mechanism for encouraging more effective in-context exploration.

F.2. Scalability Across Model Scales

To verify the scalability of LINE, we evaluate the recipe across different model sizes: Qwen3-1.7B-Base, 4B-Base, and 8B-Base. As summarized in Table 7, LINE consistently delivers performance gains regardless of the base model's capacity.

Specifically, on the in-domain reasoning average, LINE improves the 1.7B model by 7.6%, the 4B model by 4.4%, and the 8B model by 2.5% compared to the GSPO baseline. Notably, on out-of-distribution (OOD) tasks, the 8B model with LINE achieves a 73.4% average accuracy, a 2.9% improvement over the baseline.

F.3. Random Robustness

To evaluate the stability of LINE, we repeat the GSPO+LINE experiment on Qwen3-4B-Base. This setting tests whether the observed improvements are robust to training stochasticity rather than being driven by a single favorable run.

As shown in Table 8, all three runs improve over the GSPO baseline on the in-domain average. The gains range from +3.6 to +5.3 points, with the best run reaching 54.7 compared to 49.4 for GSPO. The improvements are also consistent across the most challenging mathematical benchmarks: all three runs improve AIME 2024 over the GSPO baseline, and all runs improve AIME 2025 by a large margin.

The OOD results exhibit larger variance, which is expected because LINE is primarily designed to improve mathematical reasoning by expanding in-context exploration. Nevertheless, two of the three runs improve the OOD average over GSPO, and the best run reaches 69.1. The remaining run is still slightly above the GSPO baseline. These results indicate that the main in-domain improvement of LINE is stable across independent runs, while OOD transfer is more sensitive.

Table 7. **LINE on different model sizes. Bold** indicates the best result within each model-size block. Gains over the corresponding GSPO baseline are marked in blue.

Model	In-Domain Performance					Out-of-Domain Performance			
	MATH	Olympiad	AMC	AIME 24/25	Avg.	ARC-c	GPQA	MMLU-Pro	Avg.
Qwen3-1.7B-Base									
Base	51.2	20.7	25.8	3.4/1.7	20.6	54.1	20.2	27.5	33.9
GSPO	70.6	32.3	38.4	8.1/3.8	30.6	**79.4**	28.2	41.0	49.5
+ LINE	**77.0**	**39.6**	**45.2**	**17.5/11.5**	**38.2** (+7.6)	79.1	**33.8**	**45.6**	**52.8** (+3.3)
Qwen3-4B-Base									
Base	66.0	33.2	36.6	8.5/6.9	30.2	66.9	26.3	30.9	41.4
GSPO	85.2	51.7	62.7	26.7/20.5	49.4	88.4	**48.5**	61.5	66.1
+ LINE	**88.4**	**57.2**	**66.2**	**30.5/26.7**	**53.8** (+4.4)	**91.4**	47.5	**63.8**	**67.6** (+1.5)
Qwen3-8B-Base									
Base	67.8	35.3	38.9	10.3/8.5	32.2	58.5	32.3	51.2	47.3
GSPO	89.8	58.8	72.9	34.4/25.8	56.3	93.2	50.0	68.3	70.5
+ LINE	**91.4**	**60.4**	**73.4**	**37.2/31.6**	**58.8** (+2.5)	**94.5**	**55.1**	**70.7**	**73.4** (+2.9)

Table 8. **Evaluation across three independent runs on Qwen3-4B-Base. Bold** indicates the best result. Gains over the GSPO baseline are marked in blue. Mean ± Std is computed over the three independent runs.

Model	In-Domain Performance					Out-of-Domain Performance			
	MATH	Olympiad	AMC	AIME 24/25	Avg.	ARC-c	GPQA	MMLU-Pro	Avg.
Qwen3-4B-Base	66.0	33.2	36.6	8.5/6.9	30.2	66.9	26.3	30.9	41.4
GSPO	85.2	51.7	62.7	26.7/20.5	49.4	88.4	48.5	61.5	66.1
GSPO + LINE *with different random seeds*									
Run 1	**88.4**	57.2	66.2	30.5/**26.7**	53.8 (+4.4)	91.4	47.5	63.8	67.6 (+1.5)
Run 2	87.2	57.0	66.0	30.9/24.1	53.0 (+3.6)	91.0	**51.0**	**65.2**	**69.1** (+3.0)
Run 3	87.2	**59.0**	**68.7**	**33.3**/25.2	**54.7** (+5.3)	**91.8**	43.4	63.7	66.3 (+0.2)
Mean ± Std	$87.6_{\pm 0.7}$	$57.7_{\pm 1.1}$	$67.0_{\pm 1.5}$	$31.6_{\pm 1.5}/25.3_{\pm 1.3}$	$53.8_{\pm 0.9}$ (+4.4)	$91.4_{\pm 0.4}$	$47.3_{\pm 3.8}$	$64.2_{\pm 0.8}$	$67.7_{\pm 1.4}$ (+1.6)

F.4. Pitfalls of Direct State Coverage Maximization

As discussed in Section 4.1, a natural way to incentivize in-context exploration is to directly optimize an empirical state-coverage metric, such as the In-Context Distinct State Count $C_{\text{context}}(\tau)$. In principle, this objective is closely aligned with the count-based exploration perspective: if a trajectory visits more distinct abstract states, it should cover a broader reasoning space. However, in practice, directly using such a metric as an RL reward is fragile because it depends on a hand-designed state abstraction function ϕ, for which no perfect choice exists. We instantiate this direct-optimization baseline using distinct n-gram count as a concrete example of ϕ.

As shown in Figure 9, the model initially increases the measured state coverage rapidly. However, this improvement is not accompanied by stable reasoning gains. Instead, training soon collapses: validation accuracy drops, the distinct n-gram ratio deteriorates, and the generated responses become increasingly semantically vacuous. The collapse case in Case F.4 illustrates this failure mode: the model produces long and superficially varied text, but the content does not correspond to meaningful mathematical exploration.

This behavior reflects a reward-hacking problem. Because the state abstraction is based on surface-level textual diversity, the policy can increase the reward by generating unusual or weakly related token patterns rather than by developing useful solution trajectories. In other words, maximizing the proxy itself can decouple state coverage from reasoning quality.

In contrast, LINE avoids directly optimizing a brittle coverage metric. Instead, it uses length as a coarse proxy for exploration capacity and combines it with a redundancy penalty to suppress degenerate loops. This design does not guarantee semantic coverage, but it provides a more stable surrogate: it expands the available reasoning horizon while discouraging the most common low-information solution to the length incentive.

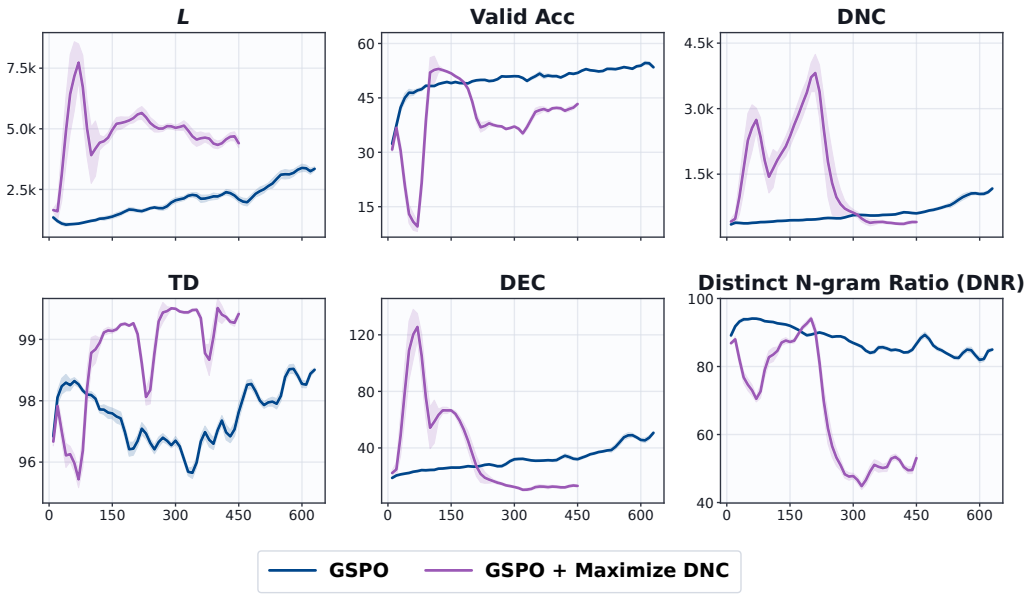


Figure 9. Directly optimizing state coverage leads to reward hacking and training collapse. Maximizing distinct n-gram count increases superficial diversity but degrades reasoning quality, resulting in reduced accuracy and unstable training dynamics.

Collapse Case

Alright, I'm going to tackle this problem systematically. It seems quite complex for a straightforward approach without deeper *context or specific techniques not immediately clear without direct context matching possibly better suited needs might require more detailed methods or perhaps tools or further specific insight provided might be needed or links to more detailed methods or perhaps further explained or provided or may need specific tools or other methods or specific possible resources or more context or better specific direction or specific further details or hints or further steps needed possibly need some better context or more direct approach or some better resources or needing maybe more detailed methods or explanation or given context or given further specific context but without further without more details without such providing helping or some or a better or way or context or might need some more direct like a better or help or like more context or something to provide some more specific something that might the context better or or might need some specific help or way that might consider more way that can help or things that might can or able provide to may able help others like or to some such or a such way that can needed can way can that might like that or can or can also may some may or can some some ...*

F.5. SFT as a Complementary

We further study how supervised fine-tuning (SFT) interacts with LINE. We aim to understand whether SFT provides signals that are difficult to acquire through RL alone.

Setup. We construct an SFT dataset by randomly sampling 4k problems from the training set and generating Chain-of-Thought (CoT) solutions using GPT-OSS-120B (Agarwal et al., 2025). We fine-tune Qwen3-4B-Base for 3 epochs with a context length of 4,096 tokens. We then apply RL training with and without LINE on top of the SFT-initialized model.

Results. The training dynamics in Figure 10 show the following trends:

1. **SFT improves initialization.** Models initialized with SFT start with higher accuracy and state coverage compared to the base model, indicating that SFT provides a stronger starting point for RL training.
2. **LINE enables continued growth.** While SFT+GSPO improves over the base model, trajectory length and state coverage tend to plateau early. Adding LINE leads to more sustained growth in both metrics, suggesting that it helps the model explore beyond the behaviors induced by supervised data.
3. **A plausible complementary effect.** The combination of SFT and LINE achieves the best performance in our setting. One possible interpretation is that SFT provides useful reasoning patterns that are not easily discovered through RL alone,

while LINE helps utilize them through extended exploration. However, we leave a more detailed investigation of this interaction to future work.

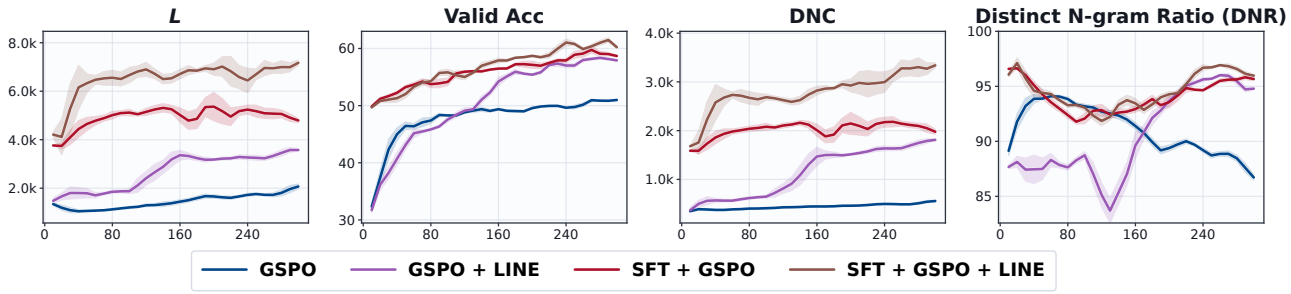


Figure 10. Interaction between supervised fine-tuning (SFT) and LINE on Qwen3-4B-Base. We compare GSPO, GSPO + LINE, SFT + GSPO, and SFT + GSPO + LINE in terms of trajectory length, validation accuracy, state coverage (DNC), and diversity (DNR). SFT improves initial performance and state coverage, while LINE further extends trajectory length and exploration beyond the supervised horizon. Their combination achieves the better performance.

F.6. Hyperparameter Sensitivity Analysis

We conduct an ablation study to verify that our method's improvements are robust to hyperparameter choices and to understand the impact of constraints.

Impact of Repetition Threshold (Θ). The threshold Θ controls the tolerance for repeated local patterns. A smaller threshold imposes a stricter constraint, while a larger threshold allows more repetition before the penalty is triggered. As shown in Table 9, $\Theta = 10$ achieves the best average performance, outperforming both the stricter setting $\Theta = 6$ and the looser setting $\Theta = 15$.

When Θ is too small, the model can be over-penalized. Mathematical reasoning often contains necessary local repetition, such as restating variables, reusing equations, or applying the same transformation across multiple cases. A strict threshold may incorrectly penalize these valid reasoning patterns, disrupting the logical flow needed for hard problems. This is reflected by the weaker AIME 2025 performance under $\Theta = 6$.

When Θ is too large, the constraint becomes less effective. The model can satisfy the length incentive by producing longer but less informative trajectories with repeated local patterns. This weakens the role of the redundancy penalty as a guardrail against degenerate length expansion. The lower average performance under $\Theta = 15$ suggests that some repetition control is necessary for converting additional tokens into useful exploration.

Table 9. The Ablation study of Θ.

	MATH	Olympiad	AIME	AMC	AIME25	Avg.
GRPO w/Clip-Higher	86.4	54.1	25.2	61.8	22.2	49.9
+LINE ($\Theta = 6$)	86.0	55.4	29.2	65.4	21.5	51.5
+LINE ($\Theta = 10$)	**88.8**	54.1	**30.4**	**65.5**	**24.5**	**52.7**
+LINE ($\Theta = 15$)	86.8	**56.0**	28.5	63.9	22.4	51.5

Impact of Repetition Magnitude β We further examine the effect of the penalty magnitude β in Eq. 7. Figure 11 compares $\beta = 0.3$ and $\beta = 0.6$. The final accuracy is relatively robust to this choice: both settings converge to similar validation performance. This suggests that the effectiveness of LINE does not rely on a narrowly tuned penalty weight.

The main difference lies in the efficiency of the resulting trajectories. A larger penalty weight encourages more concise reasoning, leading to shorter trajectories and a faster entropy decrease. In our main experiments, we use $\beta = 0.6$ as a default because it preserves the performance gains while reducing unnecessary repetition.

Impact of redundancy n-gram The redundancy penalty in LINE can be viewed as operating on a state abstraction ϕ, where repeated n-grams serve as a proxy for revisiting similar reasoning states. Different choices of n-gram size therefore

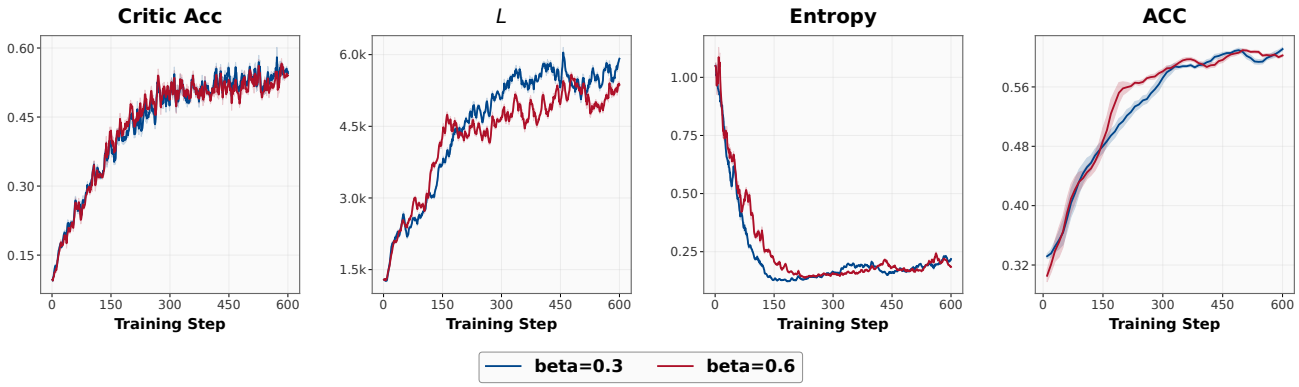


Figure 11. Training dynamics of different β.

Table 10. **Ablation of redundancy n-gram size on Qwen3-4B-Base. Bold** indicates the best result. Gains over the GSPO baseline are marked in blue.

Model	In-Domain Performance					Out-of-Domain Performance			
	MATH	Olympiad	AMC	AIME 24/25	Avg.	ARC-c	GPQA	MMLU-Pro	Avg.
Qwen3-4B-Base	66.0	33.2	36.6	8.5/6.9	30.2	66.9	26.3	30.9	41.4
GSPO	85.2	51.7	62.7	26.7/20.5	49.4	88.4	48.5	61.5	66.1
GSPO + LINE *with different redundancy n-gram sizes*									
10-gram	87.2	**59.0**	68.7	33.3/25.2	54.7 (+5.3)	91.8	43.4	63.7	66.3 (+0.2)
100-gram	**88.8**	58.7	**70.6**	30.4/**26.3**	**55.0** (+5.6)	89.6	**50.0**	65.3	**68.3** (+2.2)
512-gram	88.6	57.3	70.2	**35.8**/22.8	54.9 (+5.5)	**92.0**	46.5	**66.2**	68.2 (+2.1)
GSPO + R_{len}	88.4	54.5	66.8	29.5/21.8	52.2 (+2.8)	91.0	40.9	63.9	65.2 (-0.9)

correspond to different granularities of this abstraction.

As shown in Table 10, all n-gram settings consistently improve over the GSPO baseline on in-domain performance. While larger n-grams (e.g., 100 or 512) achieve slightly higher accuracy than smaller ones, the overall performance remains stable across a wide range of choices.

This result suggests that the effectiveness of LINE does not depend on a precise or carefully tuned abstraction. Instead, it is sufficient to employ a coarse proxy that can detect and suppress degenerate repetition patterns. In other words, the abstraction ϕ need not be lossless: approximate representations that capture task-relevant redundancy are sufficient to support improved exploration, consistent with prior findings (Abel et al., 2016).

Different n-gram sizes illustrate a trade-off between sensitivity and permissiveness. Smaller n-grams more aggressively penalize local repetition, while larger n-grams preserve more flexibility but may miss short loops. However, this trade-off has only a minor impact on overall performance, further reinforcing that the key factor is the presence of an explicit redundancy constraint rather than the exact form of ϕ.

F.7. Semantic repetition

To further investigate the limitations of n-grams in capturing semantic redundancy, we conduct an additional experiment where the n-gram-based redundancy penalty is replaced with a semantic embedding-based alternative. Specifically, we split each trajectory into chunks, extract embeddings using Qwen3-Embedding-0.6B, and compute redundancy based on embedding similarity rather than surface-level n-gram overlap.

As shown in Table 11, we make the following observations:

- Semantic redundancy still improves over the baseline. Using semantic embeddings consistently outperforms the GSPO baseline on mathematical reasoning tasks (Avg: 51.4 vs. 49.4).
- Semantic rewards are difficult to optimize in RL. In practice, semantic-based rewards are highly sensitive to implementation details, including chunking strategy, chunk size, and similarity thresholds. More importantly, semantic similarity is prone to reward hacking: as training progresses, the model can learn to circumvent the penalty without improving reasoning quality, leading to degraded performance. In contrast, n-gram-based penalties provide a more stable and

Table 11. **Comparison between n-gram and semantic redundancy penalties on Qwen3-4B-Base.** AIME 24/25 denotes AIME 2024 / AIME 2025, reported as x/y. **Bold** indicates the best result. Gains over the GSPO baseline are marked in blue.

Model	In-Domain Performance					Out-of-Domain Performance			
	MATH	Olympiad	AMC	AIME 24/25	Avg.	ARC-c	GPQA	MMLU-Pro	Avg.
Qwen3-4B-Base	66.0	33.2	36.6	8.5/6.9	30.2	66.9	26.3	30.9	41.4
GSPO	85.2	51.7	62.7	26.7/20.5	49.4	88.4	**48.5**	61.5	66.1
Redundancy penalty variants									
GSPO + LINE	88.4	**57.2**	**66.2**	**30.5/26.7**	**53.8** (+4.4)	**91.4**	47.5	**63.8**	**67.6** (+1.5)
Semantic penalty	**88.8**	54.5	61.9	28.6/23.2	51.4 (+2.0)	90.6	44.9	63.1	66.2 (+0.1)

computationally efficient signal for RL training.

These findings suggest that while semantic redundancy modeling is promising, simple surface-level signals such as n-grams currently offer a more robust and reliable optimization target for reinforcement learning.

F.8. Ablation on Length Reward

We further evaluate the cosine length reward proposed by (Yang et al., 2025b) as a representative length-based reward shaping baseline. This reward provides a smooth length-dependent signal that encourages the model to adjust its reasoning length according to the correctness of the final answer:

$$R(C, L_{\text{gen}}) = \begin{cases} \text{CosFn}(L_{\text{gen}}, L_{\max}, r_0^c, r_L^c), & \text{if } C = 1, \\ \text{CosFn}(L_{\text{gen}}, L_{\max}, r_0^w, r_L^w), & \text{if } C = 0, \\ r_e, & \text{if } L_{\text{gen}} = L_{\max}, \end{cases} \tag{16}$$

where

$$\text{CosFn}(t, T, \eta_{\min}, \eta_{\max}) = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \left(1 + \cos\left(\frac{t\pi}{T}\right)\right). \tag{17}$$

In our implementation, we set $r_0^c = 0.6$, $r_L^c = 1.0$, $r_0^w = 0$, and $r_L^w = -0.5$. This configuration ensures that correct trajectories receive higher rewards than incorrect ones across different generation lengths, while still providing a length-dependent shaping signal.

Empirically, we find that the cosine length reward performs poorly in our setting. It achieves a validation accuracy of 45.2, which is substantially lower than 57.0 for GSPO and 61.5 for GSPO + LINE. Moreover, it causes policy entropy to increase dramatically from 0.04 to 6.19, indicating unstable and overly stochastic optimization.

We hypothesize that this degradation occurs because the cosine reward does not account for the reference policy's length distribution. As a result, it provides a global length-dependent signal but does not reliably guide the model toward expanding useful in-context state coverage.

G. Case Study

To analyze the internal reasoning dynamics, we follow the principle of (Feng et al., 2025) and employ Claude-3.7-Thinking to extract the underlying reasoning graph structure from the model's generated responses. This allows us to visualize and quantify the "thought process" beyond simple token sequences or behaviors.

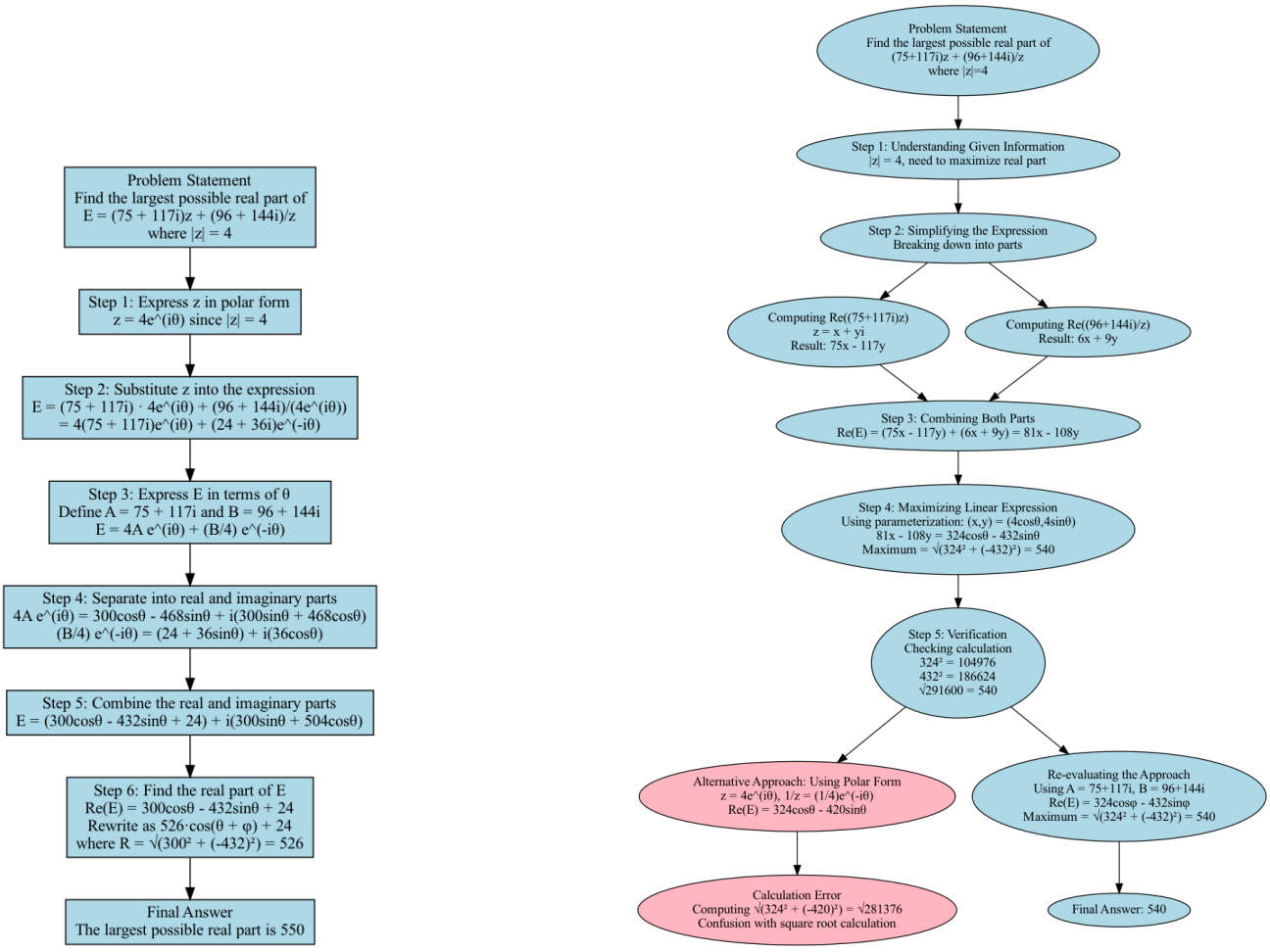
Qualitative Analysis. We select a representative problem from the AIME benchmark where the baseline fails while our method succeeds. As visualized in Figure 12a, the **GSPO baseline** (Left) follows a linear and shallow reasoning path. It attempts a direct derivation but fails to cross-check its intermediate steps, quickly converging to an incorrect answer (324).

In contrast, the model trained with our LINE recipe (Figure 12b) exhibits a significantly richer reasoning topology. Crucially, the length incentive activates *in-context exploration* behaviors: the model spawns alternative hypotheses ("Alternative Approach: Using Polar Form"), performs explicit self-verification ("Step 5: Verification"), and successfully identifies and corrects a calculation error ("Calculation Error"). This capacity to branch out and backtrack allows the model to recover from initial pitfalls and ultimately derive the correct solution (540).

Quantitative Structural Metrics To verify if this observation holds statistically, we compute the average **Depth** (maximum path length in the reasoning graph) and **Width** (average branching factor) across the 40 samples from AIME. As shown in Table 12, our method consistently expands the reasoning structure:

- **Increased Depth** ($13.8 \rightarrow 14.75$)**:** The model constructs longer logical chains, enabling deeper decomposition of complex problems.
- **Increased Width** ($2.15 \rightarrow 2.30$)**:** More importantly, the increased width indicates that the model is not merely "padding" the response with empty tokens. Instead, it engages in broader exploration by considering multiple parallel hypotheses or verification paths.

These structural metrics confirm that explicitly incentivizing response length effectively translates into a broader search horizon, enabling the model to navigate the state space more thoroughly.



(a) GSPO *(b)* GSPO + LINE

Figure 12. A case from AIME on GSPO and our recipe models.

Table 12. Exploration width and depth metrics.

Model	Depth	Width
GSPO	13.8	2.15
GSPO + LINE	14.75	2.3