LIAR: LENGTH-INTERVAL-BASED REWARD SHAPING FOR RELIABLE AND ACCURATE CHAIN-OF-THOUGHT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Reasoning Models (LRMs) have achieved remarkable success in complex problem solving through Reinforcement Learning with Verifiable Rewards (RLVR), which encourages them to generate chain-of-thought (CoT) reasoning traces. However, CoT length critically influences performance: excessively long traces increase inference cost and accumulate errors, while overly short ones break logical coherence and degrade accuracy. Although prior work suggests that the optimal CoT length varies with problem difficulty, directly targeting a single optimal length during policy learning remains impractical. In this paper, we propose the ***O**ptimal **R**easoning **I**nterval* (ORI), a statistically grounded interval of near-optimal reasoning lengths that contains the theoretical optimum and maintains high reliability across its range. Building on ORI, we introduce **L**ength-**I**nterval-b**A**sed **R**eward shaping (**LIAR**), which applies a piecewise-linear length penalty to correct trajectories outside a data-driven ORI estimate, thereby guiding reasoning toward a reliable and high-accuracy length regime. Extensive experiments across four base models show that steering generation into a *reliable* length interval—rather than compressing toward the shortest one—yields higher accuracy than prior CoT compression methods: in our primary DeepSeek-R1-Distill-Qwen-1.5B setting, LIAR improves over the strongest compression baseline LASER-D by **+1.7%** on average, with the largest gains on the hardest benchmarks (**+2.7%** on AIME 2026), where long reasoning is needed to stay reliable.

1 INTRODUCTION

Reinforcement Learning with Verifiable Rewards (RLVR) has achieved great success in evolving Large Language Models (LLMs) into powerful Large Reasoning Models (LRMs) (Lambert et al., 2025; Zhang et al., 2025b). By generating extended chains of thought (CoT), these LRMs perform strongly on mathematical, logical, and coding tasks (Guo et al., 2025; Team et al., 2025; OpenAI et al., 2024; Zeng et al., 2025). However, unconstrained generation can lead to *overthinking*: LRMs may produce tens of thousands of tokens for problems solvable in a few hundred, inflating inference cost and accumulating errors (Chen et al., 2025; Hou et al., 2025). This has motivated a growing line of work on CoT compression, aiming to reduce redundant reasoning while preserving accuracy.

Recent RL-based CoT compression methods (Team et al., 2025; Liu et al., 2025c; Aggarwal & Welleck, 2025; Hou et al., 2025; Arora & Zanette, 2025; Tu et al., 2025) address this through three distinct strategies. *Budget*-mode methods (Aggarwal & Welleck, 2025; Xu et al., 2025) specify a target token budget and penalize deviations from it, encouraging reasoning traces to match a prescribed length. *Adaptive*-mode methods (Tu et al., 2025; Fang et al., 2025) dynamically adjust the amount of reasoning according to the estimated difficulty, switching between more extensive and more concise reasoning. *Full*-mode methods (Team et al., 2025; Liu et al., 2025c; Arora & Zanette, 2025) incorporate length-aware objectives into general policy learning, encouraging concise reasoning across the training distribution without relying on a fixed budget or explicit mode switching.

Despite their differing designs, these paradigms rely on length-aware reward signals or auxiliary objectives to encourage efficient outputs. Yet they do not explicitly model the *accuracy–efficiency trade-off*: aggressive compression may induce *under-thinking*—premature halting or brittle shortcuts

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

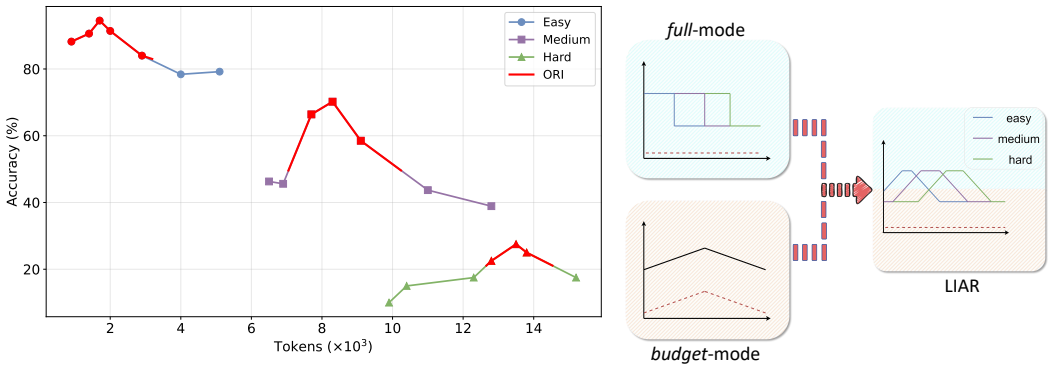


Figure 1: **Left:** Reasoning accuracy vs. CoT length for datasets of varying difficulty. Accuracy peaks at intermediate lengths before declining due to over-reasoning, forming the ORI (red), which shifts rightward with increasing difficulty. **Right:** LIAR combines the variable-length flexibility of full-mode compression with the length controllability of budget-mode shaping by replacing a single fixed budget with difficulty-dependent high-reward intervals.

on difficult problems (Zhang et al., 2025b; Su et al., 2025)—while insufficient compression leaves traces on easy problems unnecessarily verbose. In both cases, they lack a principled way to identify the range of reasoning lengths that sustains high accuracy for a given problem.

A key insight from recent theoretical work (Wu et al., 2025b; Zhang et al., 2025a) is that reasoning accuracy follows an inverted U-shaped curve as a function of CoT length: accuracy initially improves as additional steps enable finer decomposition, then degrades as error accumulation outweighs the benefit. This reveals a difficulty-dependent *Optimal CoT Length*—but it remains a post-hoc observation with no practical mechanism to target it during policy learning.

We bridge this gap by proposing the ***O**ptimal **R**easoning **I**nterval* (ORI): a target interval of CoT lengths that captures near-optimal reasoning behavior. Rather than pursuing a fragile single-point optimum, the ORI identifies the contiguous range of lengths that all achieve near-peak reliability (Theorem 3.1, Proposition 3.3). As shown in Figure 1 (left), this interval covers the plateau of the accuracy curve for each difficulty level, providing a robust target that tolerates length variation while maintaining high accuracy.

Building on the ORI, we propose **L**ength-**I**nterval-b**A**sed **R**eward shaping (**LIAR**), a reward design that steers policy learning toward this interval. As shown in Figure 1 (right), LIAR combines the flexibility of *full*-mode compression with the controllability of *budget*-mode shaping: correct trajectories inside the estimated ORI receive maximal reward, allowing reasoning lengths to vary within a reliable range, while correct trajectories that under- or over-reason outside the interval are softly penalized. Both interval endpoints are estimated from a data-driven *success-ratio curve* during training, requiring no ground-truth traces or manually specified budgets.

We validate LIAR across four models (DeepSeek-R1-Distill-Qwen-1.5B/7B, DeepScaleR-1.5B-Preview, and DeepSeek-R1-Distill-Llama-8B), eight mathematical benchmarks, and four out-of-domain tasks. In the primary DeepSeek-R1-Distill-Qwen-1.5B setting, LIAR achieves the strongest accuracy among compared CoT compression methods, improving over the strongest compression baseline LASER-D (Liu et al., 2025c) by +1.7% on average (and by +2.3/+4.1/+2.7% on AIME 2024/2025/2026, respectively). The advantage persists across models: LIAR also leads LASER-D by +2.5% on DeepSeek-R1-Distill-Qwen-7B, +1.2% on DeepScaleR-1.5B-Preview, and +1.8% on DeepSeek-R1-Distill-Llama-8B.

2 Preliminaries

We first introduce the notation in Section 2.1. We then formulate the CoT reasoning process in Section 2.2 and the unified framework of length-based reward shaping in Section 2.3.

2.1 NOTATION

We denote a reasoning query by $\mathbf{q}$ and its difficulty level by D, where larger D indicates harder problems. Let M denote the model's reasoning boundary (Chen et al., 2024), i.e., the maximum sub-problem complexity it can reliably solve in a single step. A CoT trace with N intermediate steps is written as $H_N = [q_1, a_1, \ldots, q_N, a_N]$, where each q_i is an intermediate sub-query and a_i is the corresponding answer. We use L to denote the output token count of the generated trace, and $A(N)$ to denote the expected accuracy of an N-step CoT. The policy model is denoted by π_θ, and k denotes the number of independent rollouts per query.

2.2 COT REASONING PROCESS

Following (Wu et al., 2025b), the expected accuracy of an N-step CoT is modeled as:

$$A(N) = \eta \left((1 - E(N, M, D))(1 - \sigma(D))\right)^N, \tag{1}$$

where $\sigma(D)$ denotes the sub-question generation error and $E(N, M, D)$ denotes the sub-answer error. Under this model, maximizing $A(N)$ yields the *Optimal CoT Length*:

$$N^*(M, D) = \frac{DZ}{M(Z+1)},$$

where $Z = W_{-1}\left(-\frac{1-\sigma(D)}{e}\right)$ and W_{-1} is the lower branch of the Lambert W function. This suggests that the optimal reasoning length grows with problem difficulty D and decreases with model capability M, balancing the benefit of finer decomposition against the risk of error accumulation.

In practice, directly targeting $N^*(M, D)$ is infeasible, since M, D, $\sigma(D)$, and $E(N, M, D)$ cannot be reliably estimated during training. Moreover, the reasoning step count N is a latent decomposition variable, whereas the output token count L is directly observable during generation. We therefore connect N and L through the following assumption:

Assumption 2.1. For a reasoning model θ and query $\mathbf{q}$, the output token count L grows approximately linearly with the number of reasoning steps N, i.e., $L = \gamma_\theta N + o(N)$ for some $\gamma_\theta > 0$.

This assumption formalizes the intuition that deeper multi-step reasoning requires proportionally more generated content, aligning with the Compute Law of (Zhang et al., 2025a) and the t-hop formulation of (Wu et al., 2025b). It justifies using the observable token count L as a surrogate for the latent reasoning step count N in reward design.

2.3 LENGTH-BASED REWARD SHAPING

Following (Liu et al., 2025c), many length-control objectives can be viewed as augmenting the verifiable correctness reward with a length-aware shaping term:

$$R(\mathbf{q}, H_N) = r_{\mathrm{acc}}(a) + \Phi_{\mathrm{len}}(\mathbf{q}, H_N, a),$$

where $r_{\mathrm{acc}}(a)$ verifies the final answer and $\Phi_{\mathrm{len}}(\mathbf{q}, H_N, a)$ assigns an auxiliary reward or penalty based on the generated reasoning trace, possibly conditioned on the query and correctness. While this abstraction injects efficiency preferences into RLVR, it does not determine which reasoning lengths should be preferred. Existing methods rely on fixed budgets, adaptive switching rules, or global conciseness objectives rather than identifying the problem-dependent length range that preserves high accuracy.

3 OPTIMAL REASONING INTERVAL

The key insight motivating our approach is that targeting a single optimal reasoning length is both theoretically fragile (sensitive to estimating latent quantities such as M and D) and practically unnecessary (a range of nearby lengths can achieve near-optimal reliability). In this section, we formalize this insight by reinterpreting the *Optimal CoT Length* through statistical reliability (Section 3.1) and defining the ORI as a level set of the reliability function (Section 3.2). We provide an empirical case study in Appendix C and defer all proofs to Appendix G.

3.1 Reinterpretation of the Optimal CoT Length

The limitations of the Optimal CoT Length—as a post-hoc statistic that cannot be directly leveraged in policy learning—motivate a shift in perspective. Instead of treating $N^*(M, D)$ merely as a single-point optimum, we reinterpret it from the perspective of statistical reliability.

Theorem 3.1. *Let $A(N)$ denote the success probability of an N-step CoT, and suppose $N^*(M, D) \in \arg\max_N A(N)$. For k independent rollouts, define the reliability of length N at correctness threshold C as*

$$R(N; k, C) = P(\#\{\textit{correct outcomes}\} \geq C) = \sum_{i=C}^{k} \binom{k}{i} (A(N))^i (1 - A(N))^{k-i}.$$

For any reliability level $\tau \leq 1-(1-A(N^*))^k$, $N^*(M, D)$ maximizes the largest feasible correctness threshold:

$$N^*(M, D) \in \arg\max_N \; C_\tau(N), \qquad C_\tau(N) = \max\big\{C \in [\![0, k]\!] : R(N; k, C) \geq \tau\big\},$$

where $R(N; k, 0) = 1$ by convention.

In other words, for a fixed reliability level τ, the optimal length $N^*(M, D)$ supports the largest correctness threshold C: with probability at least τ, at least C out of k rollouts are correct. This connects the theoretical optimum to practical evaluation protocols commonly used in stochastic language generation, such as pass@k, where performance is assessed through repeated sampling. Moreover, it suggests an interval relaxation: under the unimodality established below, fixing C and τ yields a contiguous set of reliable reasoning lengths, which we formalize next as the ORI.

3.2 Optimal Reasoning Interval

Despite its theoretical elegance, the Optimal CoT Length $N^*(M, D)$ is difficult to deploy in practice: neither the model capability M nor the instance-specific difficulty D can be reliably quantified, and the underlying correctness landscape—governed by the error rates $\sigma(D)$ and $E(N, M, D)$—is hard to model accurately. Even the rollout-based reinterpretation in Theorem 3.1, which avoids explicit dependence on M and D, still requires identifying a precise single-point optimum over candidate reasoning lengths. Such point estimation is sensitive to rollout noise and computationally inconvenient during training. Thus, N^* remains out of reach in practice: it is defined through latent quantities, and estimating it from rollouts is fragile.

To overcome these limitations, we propose the *Optimal Reasoning Interval* (ORI): a set of reasoning lengths that are sufficiently reliable under repeated stochastic decoding.

Definition 3.2 (Optimal Reasoning Interval)**.** Given rollout count k, correctness threshold C, and reliability level $\tau \leq \max_N R(N; k, C)$, the Optimal Reasoning Interval is defined as

$$\mathrm{ORI}(k, C, \tau) = [N_l, N_r],$$

where

$$N_l = \min\{N : R(N; k, C) \geq \tau\}, \qquad N_r = \max\{N : R(N; k, C) \geq \tau\},$$

and $R(N; k, C)$ is the reliability defined in Theorem 3.1.

The ORI provides an interval-based generalization of the pointwise Optimal CoT Length. Its theoretical soundness and practical utility rely on the following two properties.

Proposition 3.3 (Unimodality of Reliability)**.** *Under the accuracy model in Eq. 1, where $A(N)$ is unimodal in N, the reliability function $R(N; k, C)$ is unimodal in N for any fixed rollout count k and correctness threshold C.*

Corollary 3.4 (Inclusion of Optimal CoT Length)**.** *For any fixed rollout count k, correctness threshold C, and reliability level $\tau \leq \max_N R(N; k, C)$, it holds that $N^* \in \mathrm{ORI}(k, C, \tau)$.*

These results ensure that the ORI is well-defined as a single contiguous interval and contains the classical Optimal CoT Length whenever the reliability threshold is below the peak reliability. The practical advantage of ORI lies in its *estimability*: instead of requiring precise single-point optimization, it can be estimated from stochastic rollouts using only correctness feedback (Section 4.2). An empirical case study illustrating ORI on real-world benchmarks is provided in Appendix C.

4 LIAR: LENGTH-INTERVAL-BASED REWARD

In this section, we present the details of the proposed Length-Interval-based Reward shaping method (LIAR). First, we introduce the general form of the LIAR reward function using an adaptive target length interval in Section 4.1. Then, we describe how the target length interval is linked to the ORI and estimated during the training phase in Section 4.2.

4.1 REWARD FUNCTION DESIGN

Motivated by the Optimal Reasoning Interval (ORI), we propose LIAR (**L**ength-**I**nterval-b**A**sed **R**eward shaping), a correctness-conditioned reward design that guides successful reasoning trajectories toward a target length interval while discouraging wasteful incorrect generations. Following the abstraction in Section 2.3, we define:

$$R_{\mathrm{LIAR}}(\mathbf{q}, H_N) = r_{\mathrm{acc}}(a) + \big[r_{\mathrm{acc}}(a)\Phi_+(L; L_l, L_r) + \big(1 - r_{\mathrm{acc}}(a)\big)\Phi_-(L; L_r)\big],$$

where $r_{\mathrm{acc}}(a) = \mathbb{I}[a \text{ is correct}]$, L is the output token count, and $[L_l, L_r]$ is the target length interval.

For correct trajectories, the interval penalty is

$$\Phi_+(L; L_l, L_r) = \begin{cases} -0.5, & L_l - L > L_0, \\ -(L_l - L)/(2L_0), & 0 < L_l - L \le L_0, \\ 0, & L_l \le L \le L_r, \\ -(L - L_r)/(2L_0), & 0 < L - L_r \le L_0, \\ -0.5, & L - L_r > L_0. \end{cases}$$

Here L_0 controls the width of the soft-penalty region outside the target interval. We set $L_0 = \lambda L_r$ with $\lambda = 0.2$, so that the tolerated deviation scales with the target length rather than using a fixed token margin. This avoids applying the same absolute tolerance across difficulty groups, whose expected reasoning lengths can differ substantially.

For incorrect trajectories, LIAR only penalizes excessive length:

$$\Phi_-(L; L_r) = \begin{cases} 0, & L < L_r, \\ -(L - L_r)/(2L_0), & 0 \le L - L_r \le L_0, \\ -0.5, & L - L_r > L_0. \end{cases}$$

Thus, correct trajectories receive maximal reward inside the interval and are softly penalized for under- or over-reasoning, while incorrect trajectories are penalized only when they are excessively long. The lower bound of -0.5 preserves correctness as the primary objective: any correct trajectory receives a higher reward than any incorrect one.

Unlike L1-Exact (Aggarwal & Welleck, 2025), which enforces proximity to a single handcrafted target length L_T via $-\alpha|L - L_T|$, LIAR targets a length interval and applies correctness-conditioned shaping. This unifies the variable-length flexibility of full-mode compression with the length controllability of budget-mode shaping, while avoiding the pitfall of rewarding incorrect answers merely for being concise.

4.2 TARGET LENGTH INTERVAL

Assumption 2.1 transports the ORI from latent depth space to observable token space, so the target interval $[L_l, L_r]$ serves as a token-space surrogate of $\mathrm{ORI}(k, C, \tau)$ rather than an approximation of the latent quantities $[N_l, N_r]$ themselves. During training, however, we cannot actively control the model's output length or directly observe per-length accuracy curves. We therefore maintain a monitoring set $\mathcal{D}^M$, disjoint from policy-gradient minibatches, and periodically estimate tier-specific target intervals from its rollouts.

Online Difficulty Tiers. Before interval estimation, we partition monitoring queries into policy-dependent difficulty tiers $\mathcal{G} = \{\mathrm{low}, \mathrm{medium}, \mathrm{high}\}$ according to their current rollout correctness; details are provided in Appendix B. Let $\mathcal{D}_D^M$ denote the monitoring queries assigned to tier D.

Success-Ratio Curve. Given a difficulty tier D, we define the *success-ratio curve* using the k monitoring rollouts as

$$S_D(L) = \frac{1}{|\mathcal{D}_D^M|} \sum_{\mathbf{q} \in \mathcal{D}_D^M} \mathbb{I}[\exists j \in \{1, \ldots, k\} : r_{\text{acc}}(a_{\mathbf{q},j}) = 1 \text{ and } |H_{\mathbf{q},j}| \leq L] \,.$$

Thus, $S_D(L)$ is a tier-specific token-budgeted success rate: it measures the fraction of monitoring queries in difficulty tier D for which the current policy produces at least one correct rollout within budget L. By construction, $S_D(L)$ is non-decreasing in L.

We next estimate the two endpoints of the target interval from this curve. For readability, we describe the endpoint estimators for a fixed tier D and write L_l, L_r in place of $L_{l,D}, L_{r,D}$ unless the tier index is needed explicitly.

Estimation of L_l (Left Endpoint). The left endpoint L_l represents the minimum token budget required to reach the prescribed success threshold for difficulty tier D. We estimate it as the smallest length at which the success-ratio reaches a tier-specific threshold τ_D:

$$L_l = \min\{L : S_D(L) \geq \tau_D\},$$

where τ_D is the success threshold for tier D.

Estimation of L_r (Right Endpoint). The right endpoint L_r marks the onset of diminishing returns: beyond L_r, additional tokens yield only bounded residual gains. We estimate L_r via a *tail-mass criterion*:

$$L_r = \min\{L \geq L_l : S_D(L_{\max}) - S_D(L) \leq \epsilon_D\}, \tag{2}$$

where $L_{\max}$ is the maximum generation length and ϵ_D is a tier-specific tolerance. Thus, extending the token budget beyond L_r can improve the tier-specific success rate by at most ϵ_D, yielding a controlled residual gain; we justify the choice of this criterion below.

Stabilizer Mechanism. To prevent abrupt interval shifts during training, we apply the same step-size limit to both endpoints:

$$L_l^{(t)} = \max\big(L_l^{(\text{raw})},\ L_l^{(t-1)} - \Delta_{\max}\big), \quad L_r^{(t)} = \max\big(L_r^{(\text{raw})},\ L_r^{(t-1)} - \Delta_{\max}\big),$$

where $\Delta_{\max}$ is the maximum per-update downward movement. This stabilizer ensures smooth interval contraction as the model improves during training.

Estimation of the Target Interval $[L_l, L_r]$. Combining the two endpoint estimates, we obtain an adaptive, data-driven target interval:

$$[L_l, L_r] = \big[\min\{L : S_D(L) \geq \tau_D\},\ \min\{L \geq L_l : S_D(L_{\max}) - S_D(L) \leq \epsilon_D\}\big].$$

Both endpoints are *independently estimated from data* via the success-ratio curve, requiring no proportionality assumptions between left and right boundaries. The procedure is lightweight: it uses periodic monitoring rollouts, introduces only two tier-specific endpoint hyperparameters (τ_D and ϵ_D), and naturally adapts to task difficulty and model capability as the success-ratio curve evolves. The complete training procedure, including stabilized endpoint updates, is summarized in Algorithm 1 (Appendix A).

Why the success-ratio curve? It remains to justify the choice of statistic: in what sense is S_D a reliability curve, so that reading the endpoints off it is principled rather than merely convenient? The key point is that S_D is *cumulative*: $S_D(L)$ is the fraction of monitoring queries whose first successful rollout falls within budget L—the CDF of the first-success token count. Rollouts yield it almost for free: each supplies a length and a correctness label, so S_D can be evaluated on a grid of budgets with one count per point.

The reliability signal therefore lives in the *increment*, not the value. Under Assumption 2.1 ($N \approx L/\gamma_\theta$ to first order), the increment $S_D(L+\Delta) - S_D(L)$ is the new success that depth N adds over all shorter depths: the fixed-depth signal $R(N; k, 1)$, up to a survival factor (Remark D.2), large while N is reliable and vanishing once it is not. The right endpoint is where this accumulation stops, which is what the tail-mass criterion (2) tests; the left endpoint is where the accumulation first reaches τ_D.

Appendix D develops this argument in full: it decomposes the increment into the policy's depth-proposal density, the per-depth accuracy, and a survival factor, discusses the resulting ambiguity of a

flat increment, and proves under an explicit condition (verified empirically in Appendix E) that the estimated endpoint never overruns into the inaccurate region; Appendix F supports Assumption 2.1. We treat the estimator accordingly: L_r is a principled proxy for the reliability decline onset of the current policy, not an estimate of the latent optimum $N^*(M, D)$.

5 EXPERIMENTS

We evaluate LIAR on both in-domain mathematical reasoning and out-of-domain generalization tasks. Section 5.1 describes our experimental setup and Section 5.2 presents the main results, including cross-model, cross-scale, and out-of-domain results; a post-hoc analysis of learned reasoning patterns is deferred to Appendix J.

5.1 EXPERIMENTAL SETUP

Models and Datasets. Our primary setting trains DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) on the DeepScaleR-Preview-Dataset (Luo et al., 2025) (40K competition-level math QA pairs); we additionally fine-tune DeepScaleR-1.5B-Preview (Luo et al., 2025), DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025), and DeepSeek-R1-Distill-Llama-8B (Appendix H.7). Primary results cover five mathematical benchmarks (MATH500 (Hendrycks et al., 2021b), AIME 2026, AMC 2023, OlympiadBench (He et al., 2024), IMO-Acc-Bench (DeepMind, 2025)), with three further benchmarks in Appendix H and OOD evaluation on GPQA (Rein et al., 2024), LSAT (Zhong et al., 2023), MMLU (Hendrycks et al., 2021a), and LiveCodeBench (Section 5.3).

Baselines. We compare LIAR with strong CoT compression baselines covering the three strategies of Section 1: L1-Max (Aggarwal & Welleck, 2025) (*budget*; reported only at 7B, as no 1.5B checkpoint is available), AutoThink (Tu et al., 2025) (*adaptive*), and Thinkprune-4k (Hou et al., 2025), Efficient-Reasoning (Arora & Zanette, 2025), LAPO-I (Wu et al., 2025a), and LASER-D (Liu et al., 2025c) (*full*), with LASER-D the strongest baseline in our main evaluation. We also train DAPO-OS—GRPO with the overlong penalty of DAPO (Yu et al., 2025) applied verbatim (a one-sided penalty at the context limit)—on the same backbone and data as LIAR; all methods above are then evaluated under one unified protocol (Appendix B).

Implementation details. We implement LIAR with verl (Sheng et al., 2025) and Qwen-Math-Eval (Yang et al., 2024). For ORI estimation, each query is assigned online to the *high*/*medium*/*low* difficulty tier by its current-policy group correctness over $k = 8$ rollouts using thresholds $1/3$ and $2/3$. We set $\tau_{\text{low}}, \tau_{\text{medium}}, \tau_{\text{high}} = 0.80, 0.60, 0.30$, $\epsilon_{\text{low}}, \epsilon_{\text{medium}}, \epsilon_{\text{high}} = 0.10, 0.07, 0.05$, and $\Delta_{\max} = 512$; a one-at-a-time sensitivity sweep over these hyperparameters keeps accuracy within 1.2 pp of the default (Appendix H.1). For evaluation we use a unified avg@16 protocol; details on decoding, metric computation, and compute budgets are provided in Appendix B. All other hyperparameters remain identical to LASER-D (Liu et al., 2025c), and the model is optimized using Decoupled GRPO.

5.2 RESULTS

5.2.1 IN-DOMAIN MATHEMATICAL REASONING

As shown in Table 1, LIAR achieves the best average accuracy on DeepSeek-R1-Distill-Qwen-1.5B, reaching **50.5%** across the five primary mathematical benchmarks and outperforming LASER-D, AutoThink, and DAPO-OS by +1.7%, +1.8%, and +4.4%, respectively. The improvement is consistent across all individual benchmarks, with LIAR obtaining the highest accuracy on every dataset.

Difficulty-dependent gains. The gains are most evident against strong length-adaptive baselines on harder benchmarks: LIAR improves over LASER-D by +2.7% on AIME 2026 and +1.4% on IMO-Acc-Bench, and an n-gram audit finds no contamination of the training corpus in AIME 2026 (Appendix H.3). This supports the core thesis of LIAR—that interval-based rewards preserve the longer reasoning traces required for hard problems while still penalizing unnecessary verbosity.

Accuracy and length. LIAR averages 5,472 tokens per solution, versus 12,316 for the uncompressed model (-55.6%) and 3,740 for LASER-D ($+46\%$); we report accuracy and length as a pair throughout and claim dominance on neither axis. The gain is not an artifact of spending more tokens: under matched token budgets (Appendix H.2) it concentrates on AIME 2026 ($+2.7$ to $+4.6$ pp), where LIAR

Table 1: **Main results.** Accuracy (%) and average generation length (tokens) on five mathematical benchmarks, for DeepSeek-R1-Distill-Qwen-1.5B (upper block) and DeepSeek-R1-Distill-Qwen-7B (lower block). All methods are evaluated under our unified protocol (avg@16, Appendix B); "Original" denotes the unmodified base model. Highest accuracies within each block are in **bold**.

	Average		Individual Benchmarks									
			MATH500		AIME 2026		AMC 2023		OlympiadBench		IMO-Acc-Bench	
Method	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok
DeepSeek-R1-Distill-Qwen-1.5B												
Original	46.8	12316	83.9	5042	21.2	17932	71.6	8202	43.3	11510	14.0	18896
AutoThink	48.7	6055	84.0	2195	24.0	8863	74.1	4975	44.8	5559	16.4	8684
DAPO-OS	46.1	3084	81.8	1795	19.8	4336	70.9	2560	43.9	2841	14.0	3889
Thinkprune-4k	48.2	6026	83.8	2709	21.9	8682	73.6	4388	46.5	5529	15.3	8822
Effi-Reasoning	47.0	7632	81.9	2536	20.4	10927	72.8	5468	45.3	7784	14.4	11446
LAPO-I	47.6	5392	84.3	2354	21.5	8659	72.2	3098	44.5	5471	15.6	7376
LASER-D	48.8	3740	84.2	1872	21.5	5470	73.8	2981	47.3	3474	17.2	4905
LIAR (Ours)	**50.5**	5472	**85.6**	2449	**24.2**	8061	**76.1**	3949	**48.0**	4971	**18.6**	7930
DeepSeek-R1-Distill-Qwen-7B												
Original	62.6	10004	92.6	4017	45.0	14432	88.4	6433	58.9	8987	28.1	16153
AutoThink	63.4	5732	92.3	2427	45.6	8335	**91.2**	3825	59.0	5280	28.9	8793
L1-Max-4096	59.0	3066	91.0	2113	37.3	3901	85.8	2560	55.0	2904	25.8	3852
Effi-Reasoning	61.7	7666	91.0	2568	43.3	10936	89.8	4708	57.3	6997	27.2	13121
LASER-D	62.2	3855	91.6	1836	41.0	5975	89.1	2694	**60.4**	3350	28.7	5419
LIAR (Ours)	**64.7**	6006	**93.5**	2565	**50.0**	8789	90.3	3937	60.0	5463	**29.5**	9278

reaches an accuracy that LASER-D never attains at any budget cap. More aggressive compression does not help either: DAPO-OS cuts 75% of the original tokens (3,084) yet trails the uncompressed model in accuracy (46.1% vs. 46.8%), because its one-sided penalty truncates hard problems rather than reshaping them (AIME 2026: 19.8%).

Statistical significance. We quantify decoding uncertainty with paired bootstrap 95% confidence intervals (20,000 resamples) over the per-question evaluation replicates; full per-benchmark intervals are reported in Appendix I. The macro-average gain over LASER-D is $+1.7$ pp with 95% CI $[+1.0, +2.5]$, and the gain is positive with an interval excluding zero on *every* individual benchmark (Table 11). Importantly, the effect is not an artifact of a single small benchmark: excluding AIME 2026, the remaining four benchmarks still average $+1.5$ pp. At 7B, the per-benchmark intervals tell the same story: the macro gain is $+2.5$ pp with 95% CI $[+1.4, +3.6]$ (Appendix I, Table 12), and the two benchmarks without a statistically resolved difference have intervals containing zero—unresolved gaps rather than significant regressions.

5.2.2 Cross-Scale Generalization

To verify that LIAR is not specific to a single scale, we evaluate LIAR on DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) (lower block of Table 1). LIAR obtains the best average accuracy (**64.7%**) and improves over LASER-D by +2.5% on average, with a gain of +9.0% on AIME 2026; LASER-D retains a small lead on OlympiadBench ($+0.4\%$), while LIAR leads on IMO-Acc-Bench ($+0.8\%$), so LIAR extends—rather than dominates—the accuracy–efficiency frontier at this scale. The remaining two models are reported in the appendix: DeepScaleR-1.5B-Preview (Appendix H.6), where LIAR attains the best average accuracy (55.8%, $+1.2\%$ over LASER-D and $+5.2\%$ on AIME 2026), and the non-Qwen DeepSeek-R1-Distill-Llama-8B (Appendix H.7), where LIAR again attains the best macro accuracy (63.5 vs. 61.7 for LASER-D). Across all four models, LIAR attains the best average accuracy, suggesting that interval-based shaping is not tied to a particular model family or scale and remains especially beneficial on problems requiring longer reliable reasoning.

5.3 Out-of-Domain Reasoning

We evaluate OOD generalization on GPQA, LSAT, and MMLU—spanning scientific, logical, and multi-task knowledge domains—using DeepSeek-R1-Distill-Qwen-1.5B. LIAR attains the highest accuracy on GPQA (34.1%) and MMLU (47.4%) and a tied best on LSAT (26.3%), and lies on the

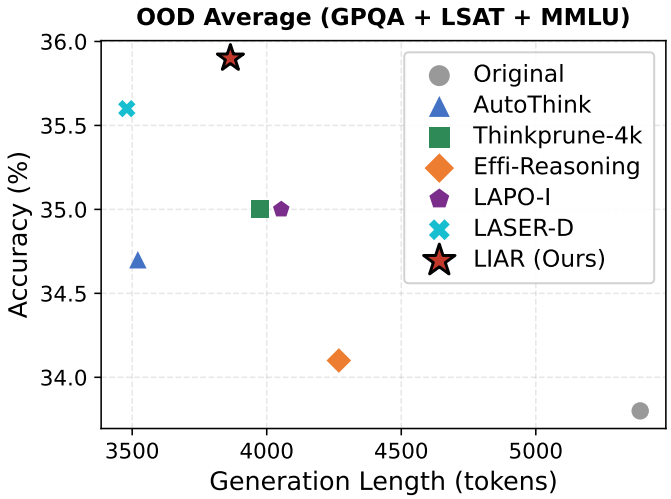


Figure 2: **Average accuracy vs. generation length across the three OOD benchmarks.** Points toward the upper-left corner represent better accuracy–efficiency trade-offs. LIAR achieves the highest average accuracy and lies on the frontier; its token count is below that of the uncompressed model but above those of the most aggressive compression baselines.

Table 2: **Out-of-domain code generation: LiveCodeBench (release_v1, 400 problems, average over n=4 samples).** Accuracy (%) and mean generated tokens per completion; Δ is the accuracy difference against the base model, with paired-by-question bootstrap 95% CIs (20,000 resamples). Highest accuracy in **bold**.

Method	Acc (%)	Tokens	Δ vs. base (95% CI)
Original (base)	24.0	12,428	—
LASER-D	28.5	5,941	$+4.5$ $[+2.1, +7.3]$
LIAR (Ours)	**29.6**	6,620	$+5.6$ $[+2.8, +8.4]$

accuracy–efficiency Pareto frontier on every benchmark (per-benchmark plots in Figure 3, Appendix H.5).

As a fourth, non-mathematical OOD task, we evaluate code generation on LiveCodeBench (release_v1, 400 problems) under the OpenCompass protocol of Liu et al. (2025c)—an average over n=4 samples, which differs from the unified avg@16 protocol of Table 1; Table 2 summarizes the results. Both RL-trained models improve substantially over the base model, and LIAR matches the strongest compression baseline ($+1.1$ pp over LASER-D, 95% CI $[-0.9, +4.1]$; Appendix H.7): shaping toward a reliable length interval carries over to code generation without loss.

Notably, the ORI is estimated exclusively from mathematical data in every case above, yet transfers to both knowledge and code domains—suggesting that the success-ratio curve captures transferable depth–reliability properties rather than domain-specific artifacts.

6 Conclusion

We introduced the Optimal Reasoning Interval (ORI), a statistically estimable interval derived from the success-ratio curve, whose endpoints mark reliable reasoning onset and the onset of diminishing returns. Building on it, LIAR performs length-interval-based reward shaping, penalizing correct traces only when they fall outside the estimated reliable range: it avoids brittle single-length targets and manual budgets while discouraging both under- and over-reasoning. Experiments across model scales, base models, eight mathematical benchmarks, and four out-of-domain tasks show that targeting a *reliable* length interval—rather than compressing hardest—yields the strongest accuracy among prior CoT compression methods.

AI USE STATEMENT

In this work, we used generative AI tools for none of the tasks with required disclosure: developing the conceptual framework (the ORI formalism), formulating the mathematical claims, proving them, designing the methodology and experiments, implementing the methods, preparing and cleaning the training and evaluation data, and interpreting the results were carried out entirely by the authors. Generating synthetic datasets, assisting with translation, and transcribing research material are not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure: editing the manuscript to improve readability—grammar, clarity, phrasing, and the organization of author-written draft passages—assisting with LaTeX markup of tables and figures, and formatting references. We have reviewed all AI-assisted work: every suggested edit was checked against the authors' original text, and every number, claim, and citation in this paper was verified against our own training and evaluation logs and against the source papers before being retained. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.04697.

Daman Arora and Andrea Zanette. Training language models to reason efficiently, 2025. URL https://arxiv.org/abs/2502.04463.

Abdelaziz Bounhar, Hadi Abdine, Evan Dufraisse, Ahmad Chamma, Amr Mohamed, Dani Bouch, Michalis Vazirgiannis, and Guokan Shang. Shorter but not worse: Frugal reasoning via easy samples as length regularizers in math rlvr, 2025. URL https://arxiv.org/abs/2511.01937.

Qiguang Chen, Libo Qin, Jiaqi WANG, Jingxuan Zhou, and Wanxiang Che. Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=pC44UMwy2v.

Sijia Chen and Baochun Li. Toward adaptive reasoning in large language models with thought rollback. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7033–7056. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chen24y.html.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*, 2025.

Google DeepMind. Towards robust mathematical reasoning, 2025. URL https://arxiv.org/abs/2511.01846.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *Advances in neural information processing systems*, 2025.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=yf1icZHC-l9.

Runquan Gui, Zhihai Wang, Jie Wang, Chi Ma, Huiling Zhen, Mingxuan Yuan, Jianye Hao, Defu Lian, Enhong Chen, and Feng Wu. Hypertree planning: Enhancing llm reasoning via hierarchical thinking. *arXiv preprint arXiv:2505.02322*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei

Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024. URL https://arxiv.org/abs/2402.14008.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021a. URL https://arxiv.org/abs/2009.03300.

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

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Ariel Kwiatkowski, Natasha Butt, Ismail Labiad, Julia Kempe, and Yann Ollivier. Likelihood-based reward designs for general llm reasoning, 2026. URL https://arxiv.org/abs/2602.03979.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

Xintong Li, Sha Li, Rongmei Lin, Hongye Jin, Linwei Li, Hejie Cui, Sarah Zhang, Chia-Yuan Chang, Kewei Cheng, Besnik Fetahu, Priyanka Nigam, Jingbo Shang, and Bing Yin. Stepwise penalization for length-efficient chain-of-thought reasoning, 2026. URL https://arxiv.org/abs/2603.00296.

Wenze Lin, Zhen Yang, Xitai Jiang, Pony Ma, and Gao Huang. Thickening-to-thinning: Reward shaping via human-inspired learning dynamics for llm reasoning, 2026. URL https://arxiv.org/abs/2602.04265.

Hanbing Liu, Lang Cao, Yuanyi Ren, Mengyu Zhou, Haoyu Dong, Xiaojun Ma, Shi Han, and Dongmei Zhang. Not all tokens matter: Towards efficient llm reasoning via token significance in reinforcement learning, 2025a. URL https://arxiv.org/abs/2506.08125.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. Dler: Doing length penalty right - incentivizing more intelligence per token via reinforcement learning, 2025b. URL https://arxiv.org/abs/2510.15110.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping, 2025c. URL https://arxiv.org/abs/2505.15612.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 3(5), 2025.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Alliot Nagle, Jakhongir Saydaliev, Dhia Garbaya, Michael Gastpar, Ashok Vardhan Makkuva, and Hyeji Kim. TERMINATOR: Learning optimal exit points for early stopping in chain-of-thought reasoning, 2026. URL https://arxiv.org/abs/2603.12529.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny

Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms, 2025. URL https://arxiv.org/abs/2505.00127.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. *arXiv preprint arXiv:2505.10832*, 2025.

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. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=yfcpdY4gMP.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Xingyu Wu, Yuchen Yan, Shangke Lyu, Linjuan Wu, Yiwen Qiu, Yongliang Shen, Weiming Lu, Jian Shao, Jun Xiao, and Yueting Zhuang. Lapo: Internalizing reasoning efficiency via length-adaptive policy optimization, 2025a. URL https://arxiv.org/abs/2507.15758.

Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in LLMs. In *Workshop on Reasoning and Planning for Large Language Models*, 2025b. URL https://openreview.net/forum?id=W8dxn7hBkO.

Yuhui Xu, Hanze Dong, Lei Wang, Doyen Sahoo, Junnan Li, and Caiming Xiong. Scalable chain of thoughts via elastic reasoning. *arXiv preprint arXiv:2505.05315*, 2025.

Kaiwen Yan, Xuanqing Shi, Hongcheng Guo, Wenxuan Wang, Zhuosheng Zhang, and Chengwei Qin. Drqa: Dynamic reasoning quota allocation for controlling overthinking in reasoning large language models, 2025. URL https://arxiv.org/abs/2508.17803.

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. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate problem solving with large language models, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. NeurIPS, 2025.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

Junyu Zhang, Yifan Sun, Tianang Leng, Jingyan Shen, Liu Ziyin, Paul Pu Liang, and Huan Zhang. When reasoning meets its laws. *arXiv preprint arXiv:2512.17901*, 2025a.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, Fangfu Liu, Xiang Xu, Jiaze Ma, Xuekai Zhu, Ermo Hua, Yihao Liu, Zonglin Li, Huayu Chen, Xiaoye Qu, Yafu Li, Weize Chen, Zhenzhao Yuan, Junqi Gao, Dong Li, Zhiyuan Ma, Ganqu Cui, Zhiyuan Liu, Biqing Qi, Ning Ding, and Bowen Zhou. A survey of reinforcement learning for large reasoning models, 2025b. URL https://arxiv.org/abs/2509.08827.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models, 2023. URL https://arxiv.org/abs/2304.06364.

A ALGORITHM

Algorithm 1 summarizes the LIAR training procedure. The procedure has two coupled loops: a monitoring loop periodically estimates tier-specific target intervals, and a policy-optimization loop uses the most recent intervals to compute LIAR rewards for GRPO. The monitoring set $\mathcal{D}^M$ is disjoint from policy-gradient minibatches, so interval estimation does not directly contribute rollout samples to the policy update.

Algorithm 1 LIAR Training Procedure

Require: Policy π_θ, training set $\mathcal{D}$, monitoring set $\mathcal{D}^M$, training horizon T, tiers $\mathcal{G} = \{\text{low}, \text{medium}, \text{high}\}$, rollout count k

Require: Thresholds $\{\tau_D\}_{D\in\mathcal{G}}$, tolerances $\{\epsilon_D\}_{D\in\mathcal{G}}$, maximum length $L_{\max}$, update period t_{update}, stabilizer limit $\Delta_{\max}$, penalty-scale coefficient λ

Generate k monitoring rollouts per query in $\mathcal{D}^M$ using the initial policy π_θ
Assign each monitoring query to an online difficulty tier by group correctness, yielding $\{\mathcal{D}_D^M\}_{D\in\mathcal{G}}$
for each difficulty tier $D \in \mathcal{G}$ **do**
    Estimate the success-ratio curve $S_D(L)$ on $\mathcal{D}_D^M$
    Initialize $L_{l,D}^{(0)} \leftarrow \min\{L : S_D(L) \geq \tau_D\}$
    Initialize $L_{r,D}^{(0)} \leftarrow \min\{L \geq L_{l,D}^{(0)} : S_D(L_{\max}) - S_D(L) \leq \epsilon_D\}$
end for
for training step $t = 1, 2, \ldots, T$ **do**
    Sample batch $\mathcal{B} \sim \mathcal{D}$ and generate k rollouts per query using π_θ
    Assign each training query $\mathbf{q} \in \mathcal{B}$ to tier $D(\mathbf{q})$ by group correctness over its k rollouts
    Set $[L_{l,D}^{(t)}, L_{r,D}^{(t)}] \leftarrow [L_{l,D}^{(t-1)}, L_{r,D}^{(t-1)}]$ for all $D \in \mathcal{G}$
    if $t \bmod t_{\text{update}} = 0$ **then**
        Generate k monitoring rollouts per query in $\mathcal{D}^M$ using the current policy π_θ
        Reassign monitoring queries to tiers, yielding updated sets $\{\mathcal{D}_D^M\}_{D\in\mathcal{G}}$
        for each difficulty tier $D \in \mathcal{G}$ **do**
            Estimate $S_D(L)$ on the current $\mathcal{D}_D^M$
            $L_{l,D}^{(\text{raw})} \leftarrow \min\{L : S_D(L) \geq \tau_D\}$
            $L_{r,D}^{(\text{raw})} \leftarrow \min\{L \geq L_{l,D}^{(\text{raw})} : S_D(L_{\max}) - S_D(L) \leq \epsilon_D\}$
            if both raw endpoints are available **then**
                $L_{l,D}^{(t)} \leftarrow \max\big(L_{l,D}^{(\text{raw})}, L_{l,D}^{(t-1)} - \Delta_{\max}\big)$
                $L_{r,D}^{(t)} \leftarrow \max\big(L_{r,D}^{(\text{raw})}, L_{r,D}^{(t-1)} - \Delta_{\max}\big)$
            else
                Keep $[L_{l,D}^{(t)}, L_{r,D}^{(t)}] \leftarrow [L_{l,D}^{(t-1)}, L_{r,D}^{(t-1)}]$ for this tier
            end if
        end for
    end if
    for each generated training rollout $H_{\mathbf{q},j}$ **do**
        Retrieve $[L_{l,D(\mathbf{q})}^{(t)}, L_{r,D(\mathbf{q})}^{(t)}]$ and set $L_{0,D(\mathbf{q})}^{(t)} \leftarrow \lambda L_{r,D(\mathbf{q})}^{(t)}$
        Compute $R_{\text{LIAR}}(\mathbf{q}, H_{\mathbf{q},j})$ using the retrieved interval, and $L_{0,D(\mathbf{q})}^{(t)}$
    end for
    Update π_θ via GRPO with rewards $\{R_{\text{LIAR}}\}$
end for

B IMPLEMENTATION DETAILS AND COMPUTE BUDGET

Difficulty assignment. LIAR uses three online, policy-dependent difficulty tiers to select tier-specific target intervals. Whenever a query $\mathbf{q}$ is used for interval estimation or reward computation, we generate $k = 8$ rollouts with the current policy and let $c(\mathbf{q})$ be the number of correct responses. We assign $\mathbf{q}$ to the *high*-difficulty tier if $c(\mathbf{q})/k < 1/3$, to the *low*-difficulty tier if $c(\mathbf{q})/k \geq 2/3$, and to the *medium*-difficulty tier otherwise; equivalently for $k = 8$, *high* means $c \leq 2$, *medium* means $c \in \{3, 4, 5\}$, and *low* means $c \geq 6$. All k rollouts of the same query share this query-level tier when selecting the target interval for reward computation, and the tier is recomputed from fresh rollouts whenever the query is sampled again. The monitoring set uses the same assignment rule at each

interval update, so its tier membership changes with the policy rather than with source labels or benchmark categories.

Monitoring and reward hyperparameters. Training uses the 40K-question DeepScaleR-Preview-Dataset, with a held-out monitoring set $\mathcal{D}^M$ of 500 queries disjoint from GRPO minibatches. Before training, we run one monitoring pass to initialize the tier-specific intervals; during training, we update these intervals every 20 training steps. At each interval update, we sample $k = 8$ rollouts per monitoring query, reassign monitoring queries to tiers, and estimate the success-ratio curve $S_D(L)$ for each tier. In our runs, each tier contains at least 11% of the monitoring queries; if a raw interval cannot be estimated for a tier during training, we keep the previous interval for that tier while updating the others normally. We set $\tau_{\text{low}}, \tau_{\text{medium}}, \tau_{\text{high}} = 0.80, 0.60, 0.30$, $\epsilon_{\text{low}}, \epsilon_{\text{medium}}, \epsilon_{\text{high}} = 0.10, 0.07, 0.05$, $\Delta_{\max} = 512$, $\lambda = 0.2$ for $L^{(t)}_{0,D} = \lambda L^{(t)}_{r,D}$.

Evaluation protocol. All benchmark results are computed with Qwen-Math-Eval using stochastic decoding at temperature 0.6. We use a unified protocol across all methods and benchmarks: **16 rollouts per question (avg@16)** for every dataset, including AIME 2024/2025/2026, AMC 2023, MATH500, OlympiadBench, IMO-Acc-Bench, GSM8K, GPQA, LSAT, and MMLU; the only exception is LiveCodeBench, which follows the OpenCompass protocol of Liu et al. (2025c) (n=4; Section 5.3). We report average correctness and average generation length over all generated responses; this is not a pass@k or majority-vote metric. All baselines, including LASER-D and DAPO-OS, are evaluated under this identical protocol; where an original publication reports a different rollout budget or benchmark version, we evaluate its released or re-trained checkpoint under our protocol for comparability. Unless otherwise specified, the remaining decoding settings and GRPO hyperparameters follow LASER-D.

Compute budget. Each 1.5B LIAR training run uses 8 NVIDIA A100 GPUs for approximately 5 days. The 7B LIAR training run uses 8 NVIDIA A100 GPUs for approximately 2 weeks. Periodic monitoring rollouts add overhead relative to standard GRPO, but they reuse the same policy and evaluation stack and do not require training an additional model.

C A Case Study of ORI

To empirically illustrate the ORI, we conduct an offline case study on MATH-style problems from MATH500 (Hendrycks et al., 2021b) and AIME 2024 using DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025), as shown in Figure 1 (left). For this case study only, we construct three illustrative difficulty groups—*easy* (MATH level 1), *medium* (MATH level 5), and *hard* (AIME 2024)—and follow the length-control protocol of (Fu et al., 2023; Wu et al., 2025b) to elicit CoTs of varying lengths. We set the reliability level to $\tau = 0.85$ and use $k = 8$ rollouts, with group-specific correctness thresholds $C = 6, 3, 1$ for the easy, medium, and hard groups, respectively, reflecting their decreasing solvability. The resulting curves show that: **(i)** a contiguous high-reliability region emerges for each difficulty group, supporting the ORI structure in Proposition 3.3; **(ii)** the estimated ORI shifts rightward as difficulty increases, consistent with the intuition that harder problems require longer reasoning; **(iii)** the empirical accuracy peak lies within the estimated ORI, consistent with Corollary 3.4.

D Formal Bridge Between the Success-Ratio Curve and Reliability

This appendix develops the increment argument of Section 4.2 in full: we state the identity behind it, decompose the increment into interpretable factors, isolate the condition under which the ϵ_D-rule provably does not overrun into the inaccurate region, and state explicitly what we do not claim. Let $\bar{n} = L_{\max}/\gamma_\theta$ denote the depth corresponding to the maximum budget.

Lemma D.1 (Success ratio as budgeted pass@k)**.** *Grant Assumption 2.1 and suppose the k monitoring rollouts of each query are independent. Define the per-rollout budgeted success probability*

$$p_{\mathbf{q}}(L) \;=\; P_{\mathbf{q}}\big(r_{\text{acc}}(a) = 1 \textit{ and } N \le L/\gamma_\theta\big).$$

Then

$$S_D(L) \;=\; \frac{1}{|\mathcal{D}_D^M|} \sum_{\mathbf{q}\in\mathcal{D}_D^M} \Big[1 - \big(1 - p_{\mathbf{q}}(L)\big)^k\Big]. \tag{3}$$

Lemma D.1 shows that S_D is exactly the tier-averaged k-rollout *budgeted pass rate*: the token-space empirical analogue of the reliability $R(N; k, 1)$ in depth space, marginalized over the policy's depth proposal. In this sense S_D is not an ad-hoc statistic but the budgeted-reliability curve of the current policy. What its *value* does not establish is *where* reliable depths lie relative to its shape; since S_D is monotone, that information lives in its increment.

Remark D.2 (Where the increment can mislead). Differentiating (3) and writing $n = L/\gamma_\theta$ gives, up to the $o(\cdot)$ term of Assumption 2.1,

$$\frac{d}{dL} S_D(L) \;=\; \frac{1}{\gamma_\theta\,|\mathcal{D}_D^M|} \sum_{\mathbf{q}\in\mathcal{D}_D^M} k\big(1 - p_{\mathbf{q}}(L)\big)^{k-1}\, A_{\mathbf{q}}(n)\, f_{\mathbf{q}}(n),$$

where $A_{\mathbf{q}}(n)$ is the per-depth accuracy of the policy at depth n and $f_{\mathbf{q}}(n)$ the density with which it proposes that depth for query $\mathbf{q}$. The marginal gain of budget is thus a product of three factors—proposal density, per-depth accuracy, and the survival factor $k(1-p_{\mathbf{q}})^{k-1}$—so it vanishes for exactly two reasons: deep traces are inaccurate ($A_{\mathbf{q}} \approx 0$), or such depths are rarely proposed ($f_{\mathbf{q}} \approx 0$). These are the only two ways in which a flat increment can mislead the endpoint rule. We impose a condition on the accuracy factor (Assumption D.3 below) and make no claim about the proposal-density factor: if the policy rarely reaches a depth, the endpoint simply tracks what the policy can reach, which is the intended behavior of a policy-dependent interval.

Assumption D.3 (Tail collapse)**.** For difficulty tier D there exist a depth $\hat{n}_D$ and a constant $\eta_D \in [0, 1)$ such that, for the current policy,

$$P_{\mathbf{q}}\big(r_{\mathrm{acc}}(a) = 1 \;\big|\; N \geq \hat{n}_D\big) \;\leq\; \eta_D \qquad \forall \mathbf{q} \in \mathcal{D}_D^M;$$

that is, beyond depth $\hat{n}_D$ traces are almost surely incorrect (the over-reasoning regime).

Proposition D.4 (Right endpoint as a safe surrogate)**.** *Grant Assumptions 2.1 and D.3. Then the estimator* (2) *satisfies:*

(i) (near-sufficiency) $S_D(L_r) \;\geq\; S_D(L_{\max}) - \epsilon_D$*: stopping at L_r forfeits at most ϵ_D of the achievable success mass, by construction of the rule;*

(ii) (no overrun) *if $k\,\eta_D \leq \epsilon_D$, then $L_r \leq \gamma_\theta \hat{n}_D$: the interval never extends into the over-reasoning regime.*

Proposition D.4 grounds the estimator in the ORI reliability view rather than treating it as an ad-hoc length heuristic. Item (i) is unconditional—it is the operational content of the rule—while item (ii) requires the tail-collapse condition, whose constant η_D is a property of the current policy and is measurable from monitoring rollouts; we verify it on AIME 2026 in Appendix E. This is a condition for the *guarantee*, not a requirement for running the algorithm: LIAR is well defined whether or not it holds. We do *not* claim the stronger depth-space alignment $L_r \geq \gamma_\theta n_{r,D}$: that would additionally require the policy to keep proposing depths inside the reliable window (a coverage condition on the proposal-density factor of Remark D.2), which we neither assume nor verify. Throughout, the estimator is a *principled proxy* for the reliability decline onset of the current policy—which is what reward shaping needs—and not an estimate of the latent optimum $N^*(M, D)$, which depends on the inaccessible $E(N, M, D)$ and $\sigma(D)$. Assumption 2.1 is validated empirically in Appendix F; proofs are given in Appendix G.

E Within-Query Evidence for the Tail-Collapse Condition

This appendix reports the experiment that empirically tests the tail-collapse condition of Assumption D.3, on which Proposition D.4(ii) rests.

Setup. We use the full evaluation records of the LIAR-trained DeepSeek-R1-Distill-Qwen-1.5B model on AIME 2026 under the unified protocol (30 questions $\times$ 16 rollouts at temperature 0.6,

480 rollouts in total; aggregate accuracy 24.2%, matching Table 1). For every rollout we record its generation length and correctness. Token counts are calibrated so that the population mean equals the reported mean generation length (8,061 tokens); per-response lengths are measured from the raw completion text.

Why naive length binning is confounded. Pooling all 480 rollouts and binning by length, accuracy falls sharply as length grows (from 52.7% below 4k tokens to 0% beyond 12k). However, this pooled trend conflates two effects: question difficulty (harder questions elicit both longer and less accurate traces) and the within-question effect of over-reasoning. In our sample, 12 of 30 questions produce zero correct rollouts and have a mean length of 9.5k tokens, which alone drives the long-length buckets to 0% accuracy regardless of any within-question effect.

Within-query design. We therefore restrict to the 17 questions with mixed outcomes (at least one correct and one incorrect rollout among the 16) and compare lengths *within* each question, which differences out question difficulty.

Table 3: **Within-query length–accuracy analysis on AIME 2026 (LIAR, 1.5B):** accuracy by within-query length quintile over the 17 mixed-outcome questions (272 rollouts).

Within-query length quintile	Rollouts	Accuracy (%)
Q1 (shortest)	54	68.5
Q2	54	46.3
Q3	54	33.3
Q4	54	37.0
Q5 (longest)	56	0.0

Findings. As summarized in Table 3: **(i)** in 15 of the 17 mixed-outcome questions (88%), the mean length of incorrect rollouts exceeds that of correct rollouts; **(ii)** incorrect rollouts are on average 2,876 tokens longer than correct rollouts within the same question (median $+773$); **(iii)** accuracy declines from 68.5% in the shortest within-query quintile to 0% in the longest quintile (11k–45k tokens), even though these rollouts answer the *same* questions. Typical examples: on question 4, correct rollouts average 4.5k tokens versus 9.6k for incorrect ones; on question 11 the gap is $+9.8$k tokens.

Interpretation w.r.t. Assumption D.3. The within-query analysis shows that, holding the question fixed, reliability collapses as traces exceed the question's reliable length range: beyond roughly 11k tokens on this benchmark, *no* rollout is correct. This is precisely the empirical content of the tail-collapse condition with $\eta_D \approx 0$ beyond $\hat{n}_D \approx 11$k tokens for the hardest tier, supporting the no-overrun guarantee of Proposition D.4(ii). We caution that this evidence covers one benchmark and one model; verifying the condition across benchmarks and along training is left as future work (see Limitations).

F Empirical Validation of Assumption 2.1

Assumption 2.1 posits that the output token count grows approximately linearly with reasoning depth, $L = \gamma_\theta N + o(N)$. We test this by fitting response length L against text-based step proxies $\hat{N}$ on unconstrained outputs, using the reasoning-step-transition proxy ($5{,}180$ outputs per model).

Table 4: **Assumption 2.1: overall fit** of L vs. step proxy $\hat{N}$ (per model).

Model	γ (tok/step)	R^2	Pearson	Spearman	log-log slope
LIAR	21.8	0.82	0.91	0.96	1.03
base	20.9	0.78	0.88	0.96	1.00

As shown in Tables 4 and 5, the fit is strong ($R^2 \geq 0.78$ overall) and the two quantities are highly rank-correlated (Spearman ≈ 0.96), and the log-log slope being close to 1 indicates that the relationship is approximately linear rather than super- or sub-linear, supporting Assumption 2.1. The slope γ is positive and model-specific, as the assumption posits; sentence- and block-count proxies show the same trend. The per-tier fits show that the length–proxy relationship remains strong in every tier

Table 5: **Assumption 2.1: per-difficulty-tier fit** of $L = \gamma \hat{N}$ for LIAR and the pre-RL base model.

Model	Tier	γ (tok/step)	R^2	Pearson	Spearman	log-log slope
LIAR	simple	22.3	0.84	0.92	0.87	0.88
LIAR	medium	21.4	0.83	0.91	0.93	0.85
LIAR	hard	21.7	0.72	0.85	0.94	0.88
base	simple	21.3	0.78	0.88	0.91	0.87
base	medium	21.1	0.64	0.80	0.91	0.84
base	hard	20.3	0.65	0.81	0.88	0.86

($R^2 = 0.72$–0.84 for LIAR), and that γ varies little across tiers (21.4–22.3 for LIAR and 20.3–21.3 for the base model)—consistent with the stability that the per-tier token intervals require. We are transparent about the limitation: $\hat{N}$ is a *text proxy*, not the latent true reasoning-step count, and such proxies are themselves length-correlated, so this constitutes empirical support rather than a strict proof; the true latent step count is not directly observable (see Limitations).

G Proof

In this section, we prove Theorem 3.1, Proposition 3.3, Corollary 3.4, Lemma D.1, and Proposition D.4. For brevity, we abbreviate the Optimal CoT Length $N^*(M, D)$ by N^*.

We first show that any maximizer of the expected accuracy also maximizes rollout reliability.

Lemma G.1. *For any correctness threshold $C \in [\![1, k]\!]$, the Optimal CoT Length N^* maximizes the reliability function $R(N; k, C)$, i.e.,*

$$R(N^*; k, C) \geq R(N; k, C), \quad \forall N.$$

Proof. Let

$$f_C(p) = \sum_{i=C}^{k} \binom{k}{i} p^i (1-p)^{k-i}, \quad p \in [0, 1].$$

Then $R(N; k, C) = f_C(A(N))$. For $C \in [\![1, k]\!]$, differentiating the binomial tail gives

$$\begin{aligned}
f'_C(p) &= \sum_{i=C}^{k-1} \binom{k}{i} \left[i p^{i-1}(1-p)^{k-i} - (k-i) p^i (1-p)^{k-i-1} \right] + k p^{k-1} \\
&= \sum_{i=C}^{k-1} \left[k \binom{k-1}{i-1} p^{i-1}(1-p)^{k-i} - k \binom{k-1}{i} p^i (1-p)^{k-i-1} \right] + k p^{k-1} \\
&= k \left[p^{k-1} + \sum_{i=C}^{k-1} \binom{k-1}{i-1} p^{i-1}(1-p)^{(k-1)-(i-1)} - \sum_{i=C}^{k-1} \binom{k-1}{i} p^i (1-p)^{(k-1)-i} \right] \\
&= k \left[\sum_{i=C}^{k} \binom{k-1}{i-1} p^{i-1}(1-p)^{(k-1)-(i-1)} - \sum_{i=C}^{k-1} \binom{k-1}{i} p^i (1-p)^{(k-1)-i} \right] \\
&= k \left[\sum_{j=C-1}^{k-1} \binom{k-1}{j} p^j (1-p)^{(k-1)-j} - \sum_{i=C}^{k-1} \binom{k-1}{i} p^i (1-p)^{(k-1)-i} \right] \\
&= k \binom{k-1}{C-1} p^{C-1}(1-p)^{k-C} \geq 0,
\end{aligned}$$

where the second equation is obtained by the well-known identities:

$$\binom{k}{i} i = k \binom{k-1}{i-1}, \ \binom{k}{i}(k-i) = k \binom{k-1}{i}.$$

Thus, f_C is non-decreasing on $[0, 1]$. Since $N^* \in \arg\max_N A(N)$, we have $A(N^*) \geq A(N)$ for all N. Therefore,

$$R(N^*; k, C) = f_C(A(N^*)) \geq f_C(A(N)) = R(N; k, C),$$

which proves the claim. □

G.1 Proof of Theorem 3.1

Proof. For any N, recall

$$C_\tau(N) = \max\{C \in [\![0, k]\!] : R(N; k, C) \geq \tau\},$$

with $R(N; k, 0) = 1$ by convention.

We show that $C_\tau(N) \leq C_\tau(N^*)$ for all N. If $C_\tau(N) = 0$, the inequality is immediate since $C_\tau(N^*) \geq 0$. Otherwise, $C_\tau(N) \in [\![1, k]\!]$ and

$$R(N; k, C_\tau(N)) \geq \tau.$$

By Lemma G.1,

$$R(N^*; k, C_\tau(N)) \geq R(N; k, C_\tau(N)) \geq \tau.$$

Hence $C_\tau(N)$ is feasible for N^*, implying

$$C_\tau(N) \leq C_\tau(N^*).$$

Since this holds for all N, N^* maximizes $C_\tau(N)$.

Finally, the condition $\tau \leq 1 - (1 - A(N^*))^k = R(N^*; k, 1)$ ensures that the feasible set for N^* contains at least $C = 1$, so the statement is non-vacuous. □

G.2 Proof of Proposition 3.3

Proof. As shown in the proof of Lemma G.1, for any fixed k and $C \in [\![1, k]\!]$, the binomial-tail function

$$f_C(p) = \sum_{i=C}^{k} \binom{k}{i} p^i (1 - p)^{k-i}$$

is non-decreasing in p. Since $R(N; k, C) = f_C(A(N))$, and $A(N)$ is unimodal in N under the accuracy model in Eq. 1, the composition of the unimodal function $A(N)$ with the non-decreasing function f_C is also unimodal in N. Therefore, $R(N; k, C)$ is unimodal in N. Consequently, every superlevel set $\{N : R(N; k, C) \geq \tau\}$ is contiguous on the ordered length grid, so whenever this set is nonempty the two endpoints of Definition 3.2 recover it exactly. □

G.3 Proof of Corollary 3.4

Proof. By Lemma G.1, N^* maximizes $R(N; k, C)$ for any fixed k and C, so

$$R(N^*; k, C) = \max_N R(N; k, C).$$

Since $\tau \leq \max_N R(N; k, C)$ by assumption, we have $R(N^*; k, C) \geq \tau$. Let N_l and N_r be the endpoints in Definition 3.2. Then N^* belongs to the reliable set $\{N : R(N; k, C) \geq \tau\}$, and hence $N_l \leq N^* \leq N_r$. Therefore, $N^* \in [N_l, N_r] = \mathrm{ORI}(k, C, \tau)$. □

G.4 Proof of Lemma D.1

Proof. Under Assumption 2.1, the token-budget event for the j-th rollout of query $\mathbf{q}$ satisfies $\{|H_{\mathbf{q},j}| \leq L\} = \{N_{\mathbf{q},j} \leq L/\gamma_\theta\}$ up to an $o(\cdot)$ term; we absorb that term into the budget window by taking L/γ_θ to the nearest integer depth and suppress it for notational clarity. Hence $\mathbb{I}\big[r_{\mathrm{acc}}(a_{\mathbf{q},j}) = 1 \text{ and } |H_{\mathbf{q},j}| \leq L\big]$ is a Bernoulli indicator with success probability $p_{\mathbf{q}}(L)$, and the k indicators are independent by assumption. The existence event in the definition of S_D therefore has probability $1 - (1 - p_{\mathbf{q}}(L))^k$, and averaging over $\mathcal{D}_D^M$ yields (3). □

G.5 Proof of Proposition D.4

For two budgets $L < L'$, define the per-query residual success mass

$$x_{\mathbf{q}}(L) \;=\; p_{\mathbf{q}}(L_{\max}) - p_{\mathbf{q}}(L) \;=\; P_{\mathbf{q}}\big(r_{\mathrm{acc}}(a) = 1 \text{ and } L/\gamma_\theta < N \le \bar{n}\big),$$

and note that, by (3),

$$\begin{aligned} S_D(L_{\max}) - S_D(L) &= \frac{1}{|\mathcal{D}_D^M|} \sum_{\mathbf{q}} \Big[\big(1 - p_{\mathbf{q}}(L)\big)^k - \big(1 - p_{\mathbf{q}}(L_{\max})\big)^k \Big] \\ &= \frac{1}{|\mathcal{D}_D^M|} \sum_{\mathbf{q}} \big(1 - p_{\mathbf{q}}(L)\big)^k \Big[1 - \big(1 - x_{\mathbf{q}}(L)\big)^k \Big]. \end{aligned} \tag{4}$$

We use the elementary bound $1 - (1-x)^k \le kx$, valid for $x \in [0,1]$ and integers $k \ge 1$.

Proof of (i). Immediate from (2): since S_D is monotone and L_r is the first budget at which the tail mass falls to at most ϵ_D, we have $S_D(L_r) \ge S_D(L_{\max}) - \epsilon_D$ (if the set in (2) is empty at $L_{\max}$, the endpoint is capped at $L_{\max}$ and equality holds). □

Proof of (ii). Fix any $L \ge \gamma_\theta \hat{n}_D$. By Assumption D.3, $x_{\mathbf{q}}(L) \le P_{\mathbf{q}}(r_{\mathrm{acc}} = 1 \mid N \ge \hat{n}_D) \cdot P_{\mathbf{q}}(N \ge \hat{n}_D) \le \eta_D$ for every $\mathbf{q}$. Applying the upper bound in (4) gives $S_D(L_{\max}) - S_D(L) \le k\,\eta_D \le \epsilon_D$, so the criterion in (2) is met at $L = \gamma_\theta \hat{n}_D$, and by minimality $L_r \le \gamma_\theta \hat{n}_D$. □

Remark G.2. Proposition D.4 claims only that L_r is a *safe surrogate* for the onset of the reliability decline under the stated condition; it does not identify the latent optimum $N^*(M, D)$, nor does it align L_r with the depth-space right edge $n_{r,D}$. The constant η_D (tail collapse) is a property of the current policy and is measurable from the monitoring rollouts of Section 4.2; Appendix E verifies the collapse condition on AIME 2026. We do not claim the converse direction: empirically, the estimated interval did not truncate the reliable window in our runs, which we attribute to the stabilizer and to the τ_D-rule anchoring the left endpoint, but we have no bound that rules it out.

H Additional Experimental Results

This section collects the supporting experiments referenced in Section 5: a hyperparameter sensitivity sweep, a matched-token-budget comparison, a contamination audit, three additional mathematical benchmarks, per-benchmark out-of-domain plots, and two further base models.

H.1 Hyperparameter Sensitivity

We run a one-at-a-time sensitivity sweep around the default hyperparameters ($\tau_D = 0.80/0.60/0.30$, $\epsilon_D = 0.10/0.07/0.05$, $\lambda = 0.2$, $\Delta_{\max} = 512$) on DeepSeek-R1-Distill-Qwen-1.5B, reporting average accuracy and token usage across the five main benchmarks. The last row of each group in Table 6 also includes the degenerate settings that remove a component entirely (no right endpoint, uniform thresholds without tiers, or no stabilizer).

As shown in Table 6, accuracy stays within 1.2 pp of the default across all fifteen settings while tokens move more, indicating that LIAR is robust to these choices rather than dependent on careful tuning. Two observations stand out. **(1) ϵ_D acts mainly as a length knob:** halving or doubling it moves token usage by $\pm 20\%$ at a cost of at most 0.9 pp accuracy, and removing the right endpoint altogether costs $+32\%$ tokens for -0.7 pp—consistent with the right endpoint suppressing unnecessary continuation without acting as a hard generation cap. **(2) The largest accuracy drop comes from removing difficulty-aware tiers** (uniform τ, -1.2 pp), indicating that the gain stems from the difficulty-adaptive design itself rather than from a tuned threshold.

H.2 Matched-Token-Budget Comparison

To test whether LIAR's advantage is merely an artifact of allowing longer outputs, we run a *token budget-forcing* sweep: both models decode naturally, and at a shared thinking-token budget we inject a wrap-up prompt, then compare accuracy at matched budgets. Table 7 reports accuracy, actual

Table 6: **Hyperparameter sensitivity.** One-at-a-time sweeps around the defaults on DeepSeek-R1-Distill-Qwen-1.5B, averaged over the five main benchmarks. Accuracy stays within 1.2 pp of the default in every setting, while token usage responds mainly to ϵ_D.

Setting	Avg Acc (%)	ΔAcc	Avg Tok	ΔTok
Right-endpoint tolerance ϵ_D (larger $\epsilon_D \Rightarrow$ earlier $L_r \Rightarrow$ shorter outputs)				
$\epsilon_D = 0.10/0.07/0.05$ (default)	50.5	—	5472	—
$\epsilon_D = 0.05/0.035/0.025$ ($\times 0.5$)	50.1	-0.4	6617	$+21\%$
$\epsilon_D = 0.20/0.14/0.10$ ($\times 2$)	49.6	-0.9	4383	-20%
no right endpoint ($L_r = L_{\max}$)	49.8	-0.7	7214	$+32\%$
Per-tier thresholds τ_D				
$\tau_D = 0.80/0.60/0.30$ (default)	50.5	—	5472	—
$\tau_D = 0.60/0.40/0.10$ (lower)	49.7	-0.8	4885	-11%
$\tau_D = 1.00/0.80/0.50$ (higher)	50.2	-0.3	6009	$+10\%$
uniform $\tau = 0.6$, no tiers	49.3	-1.2	5102	-7%
Length-reward coefficient λ ($L_0 = \lambda L_r$)				
$\lambda = 0.2$ (default)	50.5	—	5472	—
$\lambda = 0.1$	50.2	-0.3	5341	-2%
$\lambda = 0.4$	50.1	-0.4	5496	$+0\%$
Endpoint stabilizer $\Delta_{\max}$				
$\Delta_{\max} = 512$ (default)	50.5	—	5472	—
$\Delta_{\max} = 256$	50.4	-0.1	5811	$+6\%$
$\Delta_{\max} = 1024$	49.7	-0.8	4920	-10%
$\Delta_{\max} = \infty$ (no stabilizer)	49.6	-0.9	4893	-11%

average generated tokens, truncation-trigger rate, and the LIAR–LASER accuracy gap under each budget cap; the unconstrained reference points are MATH500 85.6 vs. 84.2 ($+1.4$), OlympiadBench 48.0 vs. 47.3 ($+0.7$), and AIME 2026 24.2 vs. 21.5 ($+2.7$).

Table 7: **Matched-token-budget comparison** via budget forcing on DeepSeek-R1-Distill-Qwen-1.5B. Each row shares a budget cap; "trig%" is the fraction of generations truncated by the cap. Bold marks LIAR–LASER gaps of at least $+2$ pp.

Benchmark	Budget	LIAR			LASER-D			
		Acc (%)	Tok	Trig%	Acc (%)	Tok	Trig%	LIAR−LASER
MATH500	2048	79.4	1562	24	79.6	1408	19	-0.2
MATH500	4096	82.9	1798	11	83.2	1683	8	-0.3
MATH500	6144	84.0	1951	6	83.7	1787	3	$+0.3$
MATH500	8192	85.4	2108	4	83.2	1845	1	$\mathbf{+2.2}$
OlympiadBench	6144	46.3	3321	23	46.5	3144	12	-0.2
OlympiadBench	8192	47.4	3818	15	47.6	3307	5	-0.2
OlympiadBench	10240	47.6	4062	10	47.2	3370	2	$+0.4$
OlympiadBench	12288	47.9	4171	6	47.4	3413	0	$+0.5$
AIME 2026	10240	23.8	6499	24	19.2	5435	14	$\mathbf{+4.6}$
AIME 2026	12288	23.8	6951	18	20.0	5482	8	$\mathbf{+3.8}$
AIME 2026	14336	24.2	7300	15	21.5	5504	1	$\mathbf{+2.7}$
AIME 2026	16384	24.2	7550	10	21.5	5519	1	$\mathbf{+2.7}$

Every row imposes the same cap on both methods, and LIAR naturally reasons longer (it uses more tokens than LASER-D in every row), so on hard problems it spends a larger share of that shared cap: this extra depth—not a more permissive cap—is what produces the AIME 2026 gain. At short caps both methods are truncated (LIAR more often, see trig%), so accuracy there reflects truncation rather than efficiency; the meaningful rows are those whose cap covers each model's natural length. On easy or saturated benchmarks (MATH500, OlympiadBench) the two methods track each other closely; LIAR's clear advantage is on the hardest benchmark (AIME 2026, $+2.7$ to $+4.6$ pp), where LIAR reaches an accuracy (24.2%) that LASER-D never attains at *any* cap (19.2–21.5%). LIAR thus extends the frontier where depth matters while matching LASER-D elsewhere; we do not claim Pareto dominance.

H.3 Contamination Audit for AIME 2026

We run an n-gram overlap audit between the DeepScaleR RL training set (40,315 questions) and the 30 AIME 2026 questions. The maximum per-question overlap is 0.192 (13-gram) and 0.316 (8-gram), both from a single question (AIME 2026 Q13), and there the shared n-grams are entirely AIME answer-format boilerplate (e.g., "expressed as $m\sqrt{n}$; ... find $m + n$") with no shared mathematical content: Q13 is an equiangular-pentagon problem, and its closest training match is an unrelated rolling-cone problem. We find no evidence of problem-level reuse. Just as importantly, LIAR and LASER-D train on the *identical* DeepScaleR RL corpus (only the length reward differs), so any contamination would benefit both methods equally and could not explain LIAR's *relative* advantage on AIME 2026; the gain is attributable to the method, not to data leakage.

H.4 Additional Benchmark Results

Table 8 reports results on three additional benchmarks—AIME 2024, AIME 2025, and GSM8K—across all model families, complementing the main results in Table 1.

Table 8: **Additional benchmark results across model families.** Accuracy (%) and generation length (tokens) on AIME 2024, AIME 2025, and GSM8K. LIAR achieves the strongest AIME accuracy in five of six model–benchmark settings and remains competitive on GSM8K.

	Accuracy (%)			Generation Length (tokens)		
Method	AIME24	AIME25	GSM8K	AIME24	AIME25	GSM8K
DeepSeek-R1-Distill-Qwen-1.5B						
Original	28.9	29.2	85.8	15956	15029	2473
AutoThink	**34.6**	28.3	85.8	9514	6792	903
Thinkprune-4k	29.0	28.7	86.2	8301	7394	1157
Effi-Reasoning	29.0	27.9	85.6	10552	11522	1028
LAPO-I	29.3	28.3	86.2	8318	7609	1005
LASER-D	31.9	26.7	**86.4**	5750	5024	914
LIAR	34.2	**30.8**	86.3	8348	6779	1011
DeepScaleR-1.5B-Preview						
Original	40.3	37.5	87.9	9550	9685	2157
AutoThink	41.9	37.1	89.0	9033	9434	1686
L1-Max-4096	28.1	26.2	88.6	2941	2768	1560
Thinkprune-4k	37.2	33.3	88.0	6366	5707	1106
LAPO-I	38.1	36.7	88.2	5371	5323	1179
LASER-D	42.5	36.3	**89.3**	5802	5169	1028
LIAR	**43.8**	**38.3**	88.3	6176	5332	1227
DeepSeek-R1-Distill-Qwen-7B						
Original	54.2	42.1	93.1	13132	13840	1670
AutoThink	50.4	43.3	**93.2**	8137	9641	1144
L1-Max-4096	41.7	31.2	93.1	4207	3986	1669
Effi-Reasoning	52.9	42.1	92.5	10554	10164	1057
LASER-D	54.4	39.2	93.1	5624	5586	938
LIAR	**55.4**	**43.8**	93.1	8762	9109	1191

On DeepSeek-R1-Distill-Qwen-1.5B, LIAR achieves the highest accuracy on AIME 2025 (30.8%), outperforming the second-best method (29.2%) by +1.6%. On AIME 2024, LIAR attains 34.2%, trailing AutoThink by 0.4% and outperforming LASER-D by +2.3%. The comparison with LASER-D is instructive: LASER-D performs well on AIME 2024 but drops on AIME 2025 (26.7% vs. LIAR's 30.8%), suggesting that single-point length targets can be sensitive to benchmark-specific length requirements. On GSM8K, all methods perform comparably (85.6–86.4%), suggesting that LIAR does not over-compress simpler problems where shorter reasoning is already sufficient.

On DeepScaleR-1.5B-Preview, LIAR achieves the highest accuracy on both AIME 2024 (43.8%) and AIME 2025 (38.3%), outperforming LASER-D by +1.3% and +2.0%, respectively. The gains over LASER-D are consistent with the DeepSeek-R1-Distill-Qwen-1.5B results, suggesting that the advantage of interval-based targeting is not tied to a particular 1.5B model family. Aggressive budgeted compression can be brittle on hard benchmarks: L1-Max-4096 drops to 28.1% on AIME 2024 (vs. 40.3% for the Original model), a -12.2% degradation consistent with under-allocating reasoning budget for multi-step decomposition. In contrast, LIAR *improves* over the same uncompressed model by +3.5% on AIME 2024 while using 35% fewer tokens than it.

On DeepSeek-R1-Distill-Qwen-7B, LIAR achieves the highest accuracy on both AIME 2024 (55.4%) and AIME 2025 (43.8%), outperforming LASER-D by +1.0% and +4.6%, respectively. It also remains competitive on GSM8K (93.1%), within 0.1% of the best result, and stays on the accuracy–efficiency frontier across all three additional benchmarks.

H.5 Out-of-Domain Per-Benchmark Results

Figure 3 shows the accuracy–efficiency trade-off on each individual OOD benchmark; the averaged Pareto plot over GPQA, LSAT, and MMLU is given in Figure 2 in Section 5.3.

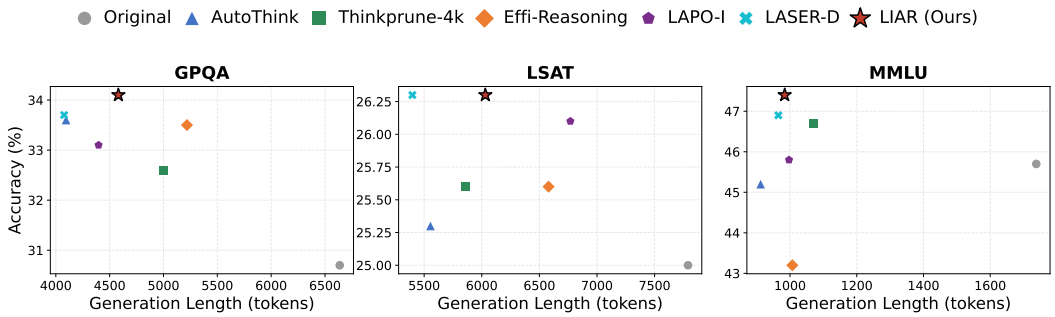


Figure 3: **Per-benchmark accuracy vs. generation length on three OOD tasks.** Points toward the upper-left corner represent better accuracy–efficiency trade-offs. LIAR achieves Pareto-optimal or near-Pareto-optimal performance on all three benchmarks: highest accuracy on GPQA (34.1%) and MMLU (47.4%), and tied-best on LSAT (26.3%).

GPQA (graduate-level science) represents problems requiring domain knowledge combined with multi-step reasoning. LIAR achieves the highest accuracy (34.1%) among all methods and lies on the accuracy–efficiency frontier. This suggests that intervals estimated from mathematical monitoring data can still provide a useful reasoning-length prior for knowledge-intensive problems, without requiring domain-specific recalibration.

LSAT (logical reasoning) tests formal deductive reasoning with relatively short but precise inference chains. LIAR achieves tied-best accuracy (26.3%) with competitive token efficiency. The margin on LSAT is modest, consistent with the possibility that many logical reasoning problems have a narrower useful length range than competition-style mathematics does. Nevertheless, LIAR does not degrade performance, suggesting that interval-based shaping does not impose harmful length constraints on this non-mathematical domain.

MMLU (multi-task knowledge) spans diverse subjects from humanities to STEM. LIAR achieves the highest accuracy (47.4%) and lies on the accuracy–efficiency frontier. The strong performance across MMLU's heterogeneous task distribution suggests that the learned interval structure captures a transferable depth–reliability pattern rather than only a mathematical-domain artifact.

Across all three OOD benchmarks, LIAR consistently occupies the Pareto-optimal or near-Pareto-optimal region of the accuracy–efficiency space. In these comparisons, no baseline simultaneously achieves higher accuracy and lower token usage, indicating that interval-based reward shaping transfers favorably to unseen reasoning domains without domain-specific adaptation of the ORI estimation procedure.

H.6 Additional Base Model: DeepScaleR-1.5B-Preview

To further test that LIAR is not tied to a single fine-tuning recipe, we evaluate LIAR on DeepScaleR-1.5B-Preview (Luo et al., 2025) under the same unified avg@16 protocol (Table 9). LIAR achieves the best average accuracy (**55.8%**), outperforming LASER-D by $+1.2\%$, with a particularly large gain on AIME 2026 ($+5.2\%$). LASER-D retains the lead on AMC 2023, and the advantage of LIAR is concentrated on the harder benchmarks, consistent with the Qwen results.

Table 9: **Cross-model results on DeepScaleR-1.5B-Preview.** Accuracy (%) and average generation length (tokens) on five benchmarks. "Original" denotes the unmodified base model. Highest accuracies are in **bold**.

	Average		Individual Benchmarks									
			MATH500		AIME 2026		AMC 2023		OlympiadBench		IMO-Acc-Bench	
Method	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok
Original	52.8	6805	87.8	3268	27.1	10228	80.0	5277	50.0	6376	19.2	8874
AutoThink	53.8	5796	87.6	2762	26.5	8346	82.7	4795	50.1	5521	**22.2**	7557
L1-Max-4096	48.9	2409	86.0	1903	19.0	2911	74.8	2270	48.3	2381	16.5	2578
Thinkprune-4k	52.4	5155	87.1	2212	24.2	6437	80.2	3516	51.4	4055	19.2	9553
LAPO-I	52.4	4389	86.3	2168	25.8	6424	79.8	3522	50.0	4011	20.0	5822
LASER-D	54.6	4051	88.5	2100	27.3	5519	**83.3**	3978	52.8	3349	21.1	5311
LIAR (Ours)	**55.8**	4491	**88.9**	2730	**32.5**	5944	82.7	4490	**53.7**	3575	21.2	5716

H.7 Transfer to a Non-Qwen Model Family and a Non-Mathematical Task

Non-Qwen model family (Llama-8B). To test that the advantage of LIAR is not specific to the Qwen model lineage, we fine-tuned DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025) with LIAR and evaluated Original, LASER-D, and LIAR under the same unified avg@16 protocol as the main tables (temperature 0.6; k=16 on every benchmark).

Table 10: **Cross-family results on DeepSeek-R1-Distill-Llama-8B.** Accuracy (%) / generation length (tokens). Highest macro accuracy is in **bold**.

Model	MATH500	AIME 2026	AMC 2023	OlympiadBench	IMO-Acc-Bench	Macro acc / tok
Original	89.4 / 4412	43.1 / 13425	90.0 / 6636	55.4 / 8970	27.1 / 15396	61.0 / 9768
LASER-D	90.8 / 3937	41.7 / 6519	90.3 / 4768	**56.8** / 5121	28.7 / 6062	61.7 / 5281
LIAR (Ours)	**91.9** / 4130	**49.4** / 8069	**90.5** / 5777	56.4 / 7876	**29.3** / 8418	**63.5** / 6854

As shown in Table 10, LIAR attains the best macro accuracy on the Llama family as well (63.5 vs. 61.7 for LASER-D and 61.0 for the original model), with a $+7.7$ pp gain on AIME 2026 and positive gains on MATH500, AMC 2023, and IMO-Acc-Bench, while LASER-D retains a small lead on OlympiadBench (-0.4 pp). As in the Qwen results, LIAR generates longer traces than LASER-D on all five benchmarks, so the result supports transfer beyond the Qwen family while retaining the same accuracy–efficiency trade-off rather than dominance.

Non-mathematical task (code generation). To probe generalization beyond mathematics, we evaluated the 1.5B models on LiveCodeBench (release_v1, 400 problems) under the OpenCompass protocol used by Liu et al. (2025c) (average over n=4 samples; this differs from the unified avg@16 protocol of the main tables). Accuracy and length are tabulated in Table 2 (Section 5.3); we report the significance analysis here. Both RL-trained models improve significantly over the base model (paired-by-question bootstrap 95% CIs: LIAR $+5.6$ $[+2.8, +8.4]$; LASER-D $+4.5$ $[+2.1, +7.3]$), while LIAR and LASER-D are statistically on par ($+1.1$ $[-0.9, +4.1]$). In terms of length, LIAR uses 11.4% more tokens per completion than LASER-D, so neither method dominates the other, while both remain far more efficient than the base model. Mathematics-only interval estimation therefore transfers to code generation without loss, with LIAR matching the strongest compression baseline; we view broader non-mathematical training domains as future work (see Limitations).

I Statistical Significance Analysis

We report paired bootstrap 95% confidence intervals (20,000 resamples) computed directly on the per-question evaluation replicates: each sample position is one decoding replicate, and resampling is over replicate-level LIAR–LASER-D differences on fixed questions. The intervals quantify decoding (run-to-run) uncertainty; they do not remove question-set uncertainty on small benchmarks such as AIME (30 questions) and AMC (40 questions).

All intervals in Table 11 are computed against the LASER-D values of Table 1 (including AMC 2023 at 73.8 and a macro of 48.8). Every individual benchmark gap is positive with an interval excluding zero, and the macro-average gain is $+1.7$ pp (95% CI $[+1.0, +2.5]$). Crucially, the effect is not an artifact of a single small benchmark: excluding AIME 2026, the remaining four benchmarks still average $+1.5$ pp in favor of LIAR. We nonetheless report intervals rather than point estimates because AIME 2026 (30 questions) and AMC 2023 (40 questions) remain small sets, and because the intervals capture decoding uncertainty only—not training-run variance.

Table 11: **Per-benchmark significance on DeepSeek-R1-Distill-Qwen-1.5B** (avg@16, paired bootstrap 95% CI, 20,000 resamples).

Benchmark	LIAR	LASER-D	Diff (pp)	95% CI
MATH500	85.6	84.2	+1.4	[+0.9, +1.9]
OlympiadBench	48.0	47.3	+0.7	[+0.1, +1.5]
AIME 2026	24.2	21.5	+2.7	[+0.6, +5.2]
AMC 2023	76.1	73.8	+2.3	[+0.2, +4.8]
IMO-Acc-Bench	18.6	17.2	+1.4	[+0.5, +2.6]
Macro average (5)	**50.5**	48.8	+1.7	[+1.0, +2.5]

Table 12: **Per-benchmark significance on DeepSeek-R1-Distill-Qwen-7B** (avg@16, all five benchmarks; paired bootstrap 95% CI, 20,000 resamples).

	Accuracy (%)		Diff	95% CI	Sig.	Tokens	
Benchmark	LIAR	LASER-D	(pp)			LIAR	LASER-D
MATH500	93.5	91.6	+1.9	[+1.5, +2.3]	yes	2565	1836
OlympiadBench	60.0	60.4	-0.4	$[-0.9, +0.1]$	no	5463	3350
AIME 2026	50.0	41.0	+9.0	[+4.6, +13.8]	yes	8789	5975
AMC 2023	90.3	89.1	+1.2	[+0.5, +2.2]	yes	3937	2694
IMO-Acc-Bench	29.5	28.7	+0.8	$[-0.3, +2.0]$	no	9278	5419
Macro (5)	**64.7**	62.2	**+2.5**	[+1.4, +3.6]	yes	6006	3855

On 7B, AIME 2026 contributes the largest gain, but the macro improvement ($+2.5$ pp, CI $[+1.4, +3.6]$) is not confined to AIME: LIAR also improves on MATH500 ($+1.9$ pp) and AMC 2023 ($+1.2$ pp), both with intervals excluding zero, and the four non-AIME benchmarks still average $+0.9$ pp. LIAR generates longer traces than LASER-D on every benchmark (6,006 vs. 3,855 mean tokens, Table 1; about 56% more), so we do not claim accuracy–efficiency dominance; three benchmarks improve significantly while no decrease is statistically resolved on the other two.

J Reasoning Pattern Analysis

This appendix complements the main experiments with a post-hoc analysis of the reasoning behavior induced by LIAR. To understand how LIAR shapes reasoning behavior, we analyze high-entropy keyword frequencies following (Wang et al., 2025; Aggarwal & Welleck, 2025), normalizing each keyword's occurrence rate on AIME 2026 in the LIAR-trained model against that of the original model.

As shown in Figure 4, LIAR induces a structured redistribution of reasoning tokens rather than uniformly increasing verbosity. Self-verification tokens such as "Wait" and "But" decrease markedly,

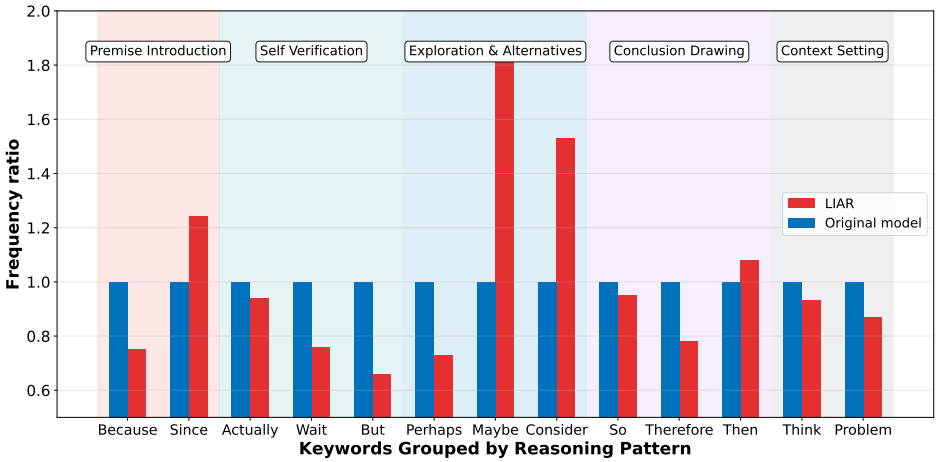


Figure 4: **Relative frequency of high-entropy tokens organized by reasoning pattern.**

indicating fewer backtracking and correction steps in LIAR-generated traces. Meanwhile, exploratory tokens such as "Maybe" and "Consider" increase, suggesting that LIAR preserves useful exploration while avoiding unproductive trial-and-error. Conclusion-related transitions become more compact, with fewer "So" and "Therefore" tokens and a slight increase in "Then". Together with higher accuracy and outputs shorter than those of the uncompressed model, these shifts suggest that LIAR promotes selective exploration, compact conclusion drawing, and reduced corrective backtracking. We caution that this analysis is post-hoc: it does not enter the reward or the interval estimator, and it should be read as descriptive evidence rather than causal support.

K Related Work

Chain-of-Thought and Overthinking. Chain-of-thought (CoT) prompting (Wei et al., 2023) established multi-step reasoning traces as a standard paradigm for complex LLM reasoning, later extended by search, rollback, and hierarchical planning methods such as Tree of Thoughts (Yao et al., 2023), Thought Rollback (Chen & Li, 2024), and HyperTree Planning (Gui et al., 2025). With test-time scaling in recent LRMs (OpenAI et al., 2024), however, models often produce unnecessarily long traces on simple instances (Zhang et al., 2025b; Chen et al., 2025; Yan et al., 2025). Prior analyses show that correct answers may appear before generation ends (Chen et al., 2025), and that accuracy can follow an inverted-U trend with reasoning length due to a trade-off between decomposition and error accumulation (Wu et al., 2025b).

Reward Shaping for Length Control. Length-control methods improve the accuracy–efficiency trade-off through budget constraints, full-mode efficient reasoning, or length-aware objectives, including L1-Max (Aggarwal & Welleck, 2025), Elastic Reasoning (Xu et al., 2025), LASER-D (Liu et al., 2025c), Thinkprune (Hou et al., 2025), Efficient-Reasoning (Arora & Zanette, 2025), and LAPO (Wu et al., 2025a). Among RL recipes, DAPO (Yu et al., 2025) contributes an overlong reward-shaping component alongside three training-stability designs (clip-higher, dynamic sampling, token-level policy-gradient loss); its overlong penalty is the only length-related part, applying a one-sided truncation-style penalty at the context limit, and existing length-shaping works typically cite it without an isolated comparison. We therefore train its length component under our unified protocol (DAPO-OS, Section 5.1) and position LIAR against it: LIAR's penalty is two-sided (penalizing both under- and over-reasoning of *correct* traces), difficulty-adaptive via online tiers, and bounded by an estimated reliable interval rather than the context limit. Recent reward- or advantage-shaping methods refine length incentives more directly: BINGO (Liu et al., 2025a) uses token significance, T2T (Lin et al., 2026) switches between exploration- and conciseness-favoring rewards, and SWAP (Li et al., 2026) allocates penalties by step importance. Likelihood-based reward design (Kwiatkowski et al., 2026) is a broader reward-design approach that replaces sparse verifier rewards with reference-answer log-likelihood rewards, but it is not length-specific because it does not explicitly reward or penalize

reasoning length. LIAR complements these methods by asking *where* length penalties should apply, using reliable length intervals to avoid uniformly discouraging long reasoning.

Orthogonal Efficiency Interventions. A complementary line reduces reasoning cost without primarily redesigning the length reward. Inference-time methods adaptively decide how much reasoning to use or when to stop, including s1 (Muennighoff et al., 2025), which enforces a token budget by appending "Wait" during decoding, as well as AutoThink (Tu et al., 2025), Thinkless (Fang et al., 2025), and TERMINATOR (Nagle et al., 2026). Training-side interventions include DLER (Liu et al., 2025b), which stabilizes length-penalized GRPO, and Shorter but not Worse (Bounhar et al., 2025), which uses moderately easy samples as implicit length regularizers. These approaches are complementary to LIAR's data-driven interval estimation for selective length regularization.

Theoretical Analyses. Recent works (Wu et al., 2025b; Zhang et al., 2025a) have formalized the non-monotonic relationship between accuracy and reasoning length, showing that intermediate reasoning depth can maximize efficacy as a function of task complexity and model capability. However, existing practical methods remain only loosely connected to this theoretical characterization: they either enforce rigid constraints without modeling the optimal range, or apply heuristic penalties without estimating where reliable reasoning begins and ends. LIAR bridges this gap by coupling statistical interval estimation with reward design: rather than only asking *how* to penalize length, it estimates *where* penalties should apply through data-driven ORI bounds.

L Limitations and Future Work

While LIAR attains the highest accuracy among the length-control methods we compare, several limitations remain.

First, the target intervals are estimated from rollouts on a finite monitoring set $\mathcal{D}^M$ collected during training. Although our OOD results are encouraging, intervals calibrated on mathematical data may not transfer optimally to domains with different reasoning structures, such as code generation, theorem proving, or multi-hop factual reasoning. A natural direction is domain-adaptive or task-adaptive interval estimation that updates calibration when the deployment distribution shifts.

Second, the success-ratio curve is an empirical token-budget surrogate for the latent reliability landscape. Its endpoints depend on the monitoring-query diversity, the number of rollouts, and the granularity of candidate length values. The stabilizer reduces abrupt interval changes but does not eliminate estimation noise, and periodic monitoring rollouts introduce additional training overhead.

Third, LIAR currently uses coarse online difficulty tiers and shared intervals within each tier. This design keeps the estimator lightweight, but it may miss fine-grained variation among problems with similar rollout correctness but different reasoning structures. Future work could explore finer-grained or continuous difficulty conditioning, as well as step-level process signals for more accurate interval estimation.

Fourth, our experiments cover multiple 1.5B-scale models and one 7B-scale model. The observed trends suggest that interval-based shaping remains useful across these settings, but broader validation on larger models, non-mathematical training domains, and different RL optimization recipes remains future work.

Fifth, the guarantees of Proposition D.4 (Appendix D) are conditional: the no-overrun guarantee requires the tail-collapse condition (Assumption D.3), whose constant is a property of the current policy and may drift during training, and we make no claim of depth-space alignment, which would additionally require a coverage condition on the policy's depth proposal. We verify the collapse condition empirically on AIME 2026 (Appendix E) and monitor rollout length distributions during training, but a complete verification across benchmarks, training horizons, and proposal distributions remains open. The right endpoint L_r should therefore be read as an empirically safe surrogate for the reliability decline onset, not as an estimator of the latent optimum $N^*(M, D)$.

Sixth, we instantiate LIAR with binary verifiable correctness rewards, where $r_{\mathrm{acc}} \in \{0, 1\}$ cleanly separates the Φ_+/Φ_- branches. With graded or non-binary rewards (e.g., soft rubric scores or process rewards), the correctness-conditioned structure requires a decision rule for splitting trajectories into reward branches; natural candidates include thresholding, expectation-weighted mixing of Φ_+ and Φ_-, or applying only the correctness-independent overlong penalty Φ_-. A failure mode of naive

thresholding is that trajectories near the threshold receive inconsistent shaping across GRPO group members; designing and validating these variants is future work.

M BROADER IMPACT

LIAR is a reward shaping method that steers reasoning traces into a reliable length interval; shorter traces are a by-product of that objective, not the objective itself. A secondary positive impact is computational: by discouraging unnecessary continuation while preserving reliable reasoning, LIAR can reduce inference latency, serving cost, and energy use for reasoning-heavy applications. Lowering the cost of reasoning models may also make advanced educational, scientific, and analytical tools more accessible in resource-constrained settings.

At the same time, these gains should be interpreted carefully. LIAR does not introduce new model capabilities, but cheaper reasoning can lower the barrier to deploying existing capabilities at scale, including in high-stakes settings where human oversight and domain-specific evaluation remain necessary. Moreover, encouraging shorter reasoning traces may reduce the amount of intermediate text available for inspection, even when final-answer accuracy is preserved. Thus, LIAR should be paired with standard safety evaluation, transparency practices, and task-specific validation rather than treated as a substitute for them.