MITIGATING LENGTH-SCALING TAX WITH ONLINE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Length scaling during reinforcement-learning (RL) post-training is often viewed as a sign of improved reasoning ability, especially on difficult problems, but may also make responses to already-solved problems unnecessarily verbose. We quantify this side effect as the *length-scaling tax* (LST): excess response length on already-solved queries without a commensurate accuracy gain. To mitigate LST, we propose *Length Self-Distillation* (LSD), which routes solved prompts to on-policy distillation and retains the original RL objective for unsolved prompts. LSD uses an exponential moving average of the online policy as its teacher, requiring no external model. We find that LSD achieves comparable or better performance than RL across multiple variants, while substantially curbing response-length growth on easy queries. LSD reduces LST from 19.0% to -3.7% on single-turn reasoning and from 31.4% to 13.7% on multi-turn agentic tasks, demonstrating that LSD effectively preserves concise response patterns on easy queries while supporting efficient exploration on difficult queries during RL post-training.

1 INTRODUCTION

Scaling the rollout budget is a common way to improve the performance of large language models (LLMs) on difficult reasoning tasks. At inference time, prompting models to think longer, sampling multiple responses, and applying verifier-guided search can translate additional computation into higher accuracy (Wei et al., 2022; Wang et al., 2022; Lightman et al., 2024; Snell et al., 2024). Yet the value of this computation depends on problem difficulty. Extended deliberation and self-correction can help on difficult problems, but offer little benefit once a problem is already solved. Therefore, adaptively allocating budgets has become an important design axis for modern reasoning models (Muennighoff et al., 2025; Aggarwal & Welleck, 2025; Yang et al., 2025; OpenAI, 2025; Anthropic, 2025; Google, 2025).

In parallel, reinforcement learning with verifiable rewards (RLVR) has become a central post-training mechanism for eliciting LLMs' reasoning and agentic capabilities (Shao et al., 2024; Wan et al., 2026a; Jin et al., 2025). Since DeepSeek-R1(Guo et al., 2025), the spontaneous growth of response length during RL has often been viewed as a behavioral signature of improving reasoning ability (Yeo et al., 2025). However, a standard RLVR objective jointly optimizes prompts of varying difficulty. Updates that promote longer and more elaborate reasoning on hard problems can also alter the policy's continuation distribution on easy ones. Under group-relative objectives, this spillover is difficult to correct. Once every response in an easy rollout group is correct, its relative advantages collapse toward zero. The easy group therefore provides no gradient for preserving a concise solution, while difficult groups continue to reshape the shared policy.

Moreover, this failure mode may be reinforced by common data-selection strategies. Dynamic sampling treats all-correct groups as uninformative and discards them from training (Yu et al., 2025), while difficulty-aware curricula downweight easy problems and concentrate the training distribution near the policy's competence frontier (Bae et al., 2026; Wan et al., 2026b; Qu et al., 2026).

Despite the rapidly growing literature on reasoning efficiency, most existing work still characterizes efficiency using coarse-grained aggregate statistics, most notably average response length. A straightforward strategy is to apply stronger length control to easier problems (Shen et al., 2025; Xu et al., 2026). However, such methods do not explicitly preserve the concise behavior that the policy already exhibits on solved queries. Existing evaluations lack a systematic metric for quantifying the

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.

unintended lengthening imposed on easy queries by subsequent RL updates. Another line of work focuses on the super-long CoT, especially for difficult problems, because these responses exhibit the most pronounced overthinking behaviors (Yuan et al., 2026; Yi et al., 2026; Xiang et al., 2025; Chen et al., 2024; Luo et al., 2026). However, imposing length control in this regime often incurs an accuracy cost. Although much of a long trajectory may appear redundant, exploratory branches within that trajectory can still uncover the reasoning path that ultimately leads to the correct solution. Response-level compression may remove not only redundant computation but also reasoning steps necessary to solve the problem.

Complementary to reward shaping, on-policy distillation (OPD) provides dense token-level supervision on states visited by the student (Agarwal et al., 2024). Because this supervision is evaluated on student-generated prefixes, it directly regularizes the evolving policy along its own state distribution. Recent work has adapted self- and contrastive OPD to reasoning compression, showing that distribution-level supervision can substantially shorten reasoning while retaining accuracy (Sang et al., 2026; Ruan et al., 2026). However, OPD is primarily used as a general compression objective. This does not address the asymmetric learning problem considered here. We believe that easy queries need a token-level preservation signal precisely because their relative RL advantage vanishes, whereas difficult queries that remain unsolved should continue to be governed by RLVR.

Motivated by this gap, our objective is to preserve concise behavior where the policy is already successful without restricting exploration elsewhere. We formalize this training-induced inefficiency as the *length-scaling tax* (LST): excess response length that emerges on already-solved queries as a shared policy undergoes RL post-training, without a commensurate gain in accuracy. Conditioning on a fixed easy query set distinguishes LST from global measures of overthinking while avoiding the survivorship bias that arises from repeatedly redefining the easy set.

To better understand LST, we first conduct several empirical studies to characterize its behavior and identify its drivers. We find that LST persists across multiple easy sets defined at different checkpoints, rather than arising from a particular model snapshot. Moreover, we find that training distributions concentrated on hard prompts further amplify the tax.

Based on this, we propose *length self-distillation* (LSD). At each training step, LSD uses the current rollout accuracy to route solved prompt groups to an OPD objective, while retaining the original RLVR objective for unsolved groups. In practice, LSD requires neither a stronger external teacher nor a separately prompted concise model. Instead, its teacher is a delayed version of the same policy lineage, instantiated as a rolling exponential moving average checkpoint. Within LSD, we compare supervised-gradient forward- and reverse-KL objectives with a sampled-action policy-gradient estimator of reverse KL, and analyze how these objectives constrain the policy at different levels of granularity.

Our study makes three contributions.

First, we introduce LST, a query-conditional metric that measures excess response length on already-solved prompts relative to an accuracy-qualified reference. We also establish the prevalence of LST, characterize its behavioral signatures, and identify its key drivers.

Second, we propose LSD, a mixed RL and distillation algorithm that routes solved rollout groups to self-distillation while retaining the original RLVR objective for unsolved groups. We develop three complementary implementations and analyze their different levels of policy-control granularity.

Third, we evaluate LSD on single-turn reasoning and multi-turn agentic tasks, showing that LSD with SG-FKL matches or improves average Pass@1 over RL while reducing LST from 19.0% to -3.7% on single-turn reasoning and from 31.4% to 13.7% on multi-turn agentic tasks. We also analyze how the distillation objective, teacher half-life, and routing threshold affect the trade-off between preserving efficient behavior and acquiring new capabilities.

2 The Length-Scaling Tax

Let $\mathcal{X}_{\text{eval}}$ denote the evaluation query set, and let $x \in \mathcal{X}_{\text{eval}}$ be a query. At each checkpoint k during RL post-training, we sample N responses

$$y_{k,i}(x) \sim \pi_k(\cdot \mid x), \qquad i = 1, \ldots, N, \tag{1}$$

using a fixed decoding configuration and response budget. Let $r(x, y) \in \{0, 1\}$ denote the correctness reward and $\ell(y)$ the number of response tokens. The empirical solve rate and mean response length of policy π_k on query x are

$$\widehat{R}(x; \pi_k) = \frac{1}{N} \sum_{i=1}^{N} r(x, y_{k,i}(x)), \qquad \widehat{L}(x; \pi_k) = \frac{1}{N} \sum_{i=1}^{N} \ell(y_{k,i}(x)). \tag{2}$$

At an RL anchor checkpoint b, we define the easy-query set induced by the anchor policy π_b using an evaluation threshold τ, set to 1 unless otherwise specified:

$$\mathcal{E}_b = \left\{ x \in \mathcal{X}_{\text{eval}} : \widehat{R}(x; \pi_b) \geq \tau \right\}. \tag{3}$$

Thus, whether a query belongs to $\mathcal{E}_b$ is determined exclusively by rollouts from π_b. At the default threshold $\tau = 1$, every sampled rollout for each selected query is correct at the anchor checkpoint.

To avoid survivorship bias, $\mathcal{E}_b$ is frozen after its construction. At every later checkpoint, we evaluate the same queries in this fixed easy set. For the frozen set $\mathcal{E}_b$, its mean accuracy and response length under an evaluated policy π_k are

$$R_b(k) = \frac{1}{|\mathcal{E}_b|} \sum_{x \in \mathcal{E}_b} \widehat{R}(x; \pi_k), \qquad L_b(k) = \frac{1}{|\mathcal{E}_b|} \sum_{x \in \mathcal{E}_b} \widehat{L}(x; \pi_k). \tag{4}$$

Here, the subscript b specifies which anchor policy defines the query set, whereas k specifies which policy is being evaluated on that set.

Let $\mathcal{K}_b$ denote all RL checkpoints evaluated on the fixed easy set $\mathcal{E}_b$ over the full training trajectory, including checkpoints before anchor b. We define an accuracy-constrained reference that captures the smallest mean token cost observed while maintaining the easy-query accuracy criterion:

$$L_b^\star = \min_{\substack{j \in \mathcal{K}_b: \\ R_b(j) \geq \tau}} L_b(j). \tag{5}$$

For fair comparisons, all methods use the same RL-frozen set $\mathcal{E}_b$ and shared reference $L_b^\star$: the shortest mean response length among RL checkpoints whose accuracy on this set is at least τ. We define the normalized length-scaling tax as

$$\mathrm{LST}_b(k) = \frac{L_b(k) - L_b^\star}{L_b^\star}. \tag{6}$$

A positive $\mathrm{LST}_b(k)$ measures the percentage of excess tokens used by π_k relative to this accuracy-qualified reference. With $\tau = 1$, the reference has perfect empirical accuracy, so additional length cannot correspond to higher observed accuracy than the reference. When $R_b(k) = 1$ as well, LST compares lengths at identical empirical accuracy. For explicitly reported settings with $\tau < 1$, LST instead measures excess length under an accuracy threshold; it does not by itself establish waste at identical accuracy.

LST exists under standard RLVR. We study a standard group-relative RLVR baseline initialized from Qwen3-4B-Base (Yang et al., 2025). We post-trained the model on the deduplicated DAPO-Math-17K dataset Yu et al. (2026) with a maximum response budget of 4096 tokens and evaluate checkpoints on AMC 2023, AIME 2025, and AIME 2026. For every query and checkpoint, we sample 32 responses using the same decoding configuration and response budget.

We first examine the dynamics of entire evaluation set in Figure 1. As expected, RLVR improves aggregate accuracy, while mean response length grows steadily throughout training.

We next examine whether the length-scaling tax emerges during RLVR by applying the fixed-easy-set protocol in Figure 2. Specifically, at each anchor checkpoint $b \in \{0, 25, 50, 75, 100\}$, we select queries with an empirical solve rate of at least $\tau = 0.875$, freeze the resulting easy set, and track its accuracy and response length at all subsequent checkpoints. Regardless of which anchor defines the easy set, accuracy remains largely stable, whereas response length continues to increase throughout RLVR.

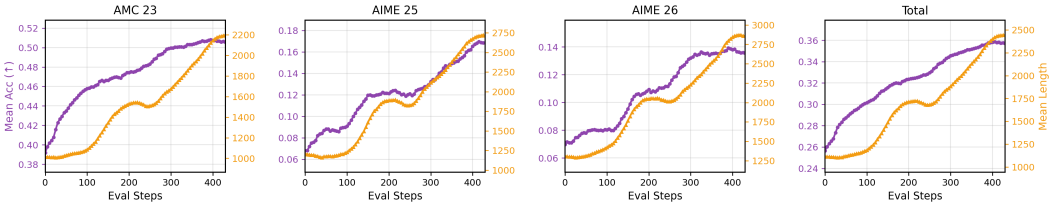


Figure 1: **Global validation accuracy and response length during RLVR.** Purple circles show mean accuracy, and orange squares show mean response length on three benchmarks and their aggregate over training.

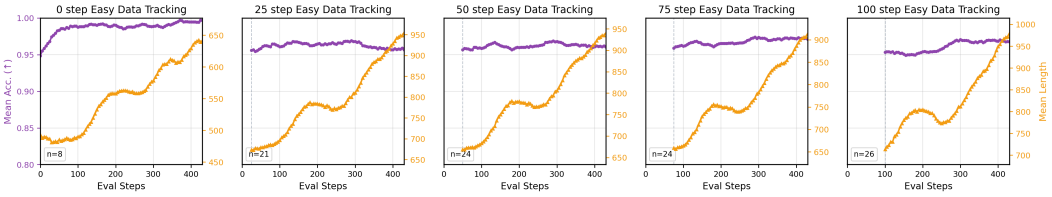


Figure 2: **Easy-query accuracy saturates while response length keeps growing during RLVR.** At each anchor $b \in \{0, 25, 50, 75, 100\}$, we select queries with solve rate at least $\tau = 0.875$ and freeze the resulting easy set.

3 What Amplifies the Tax?

Intuitively, the rollout budget, curriculum design, and training-data difficulty can alter which trajectories contribute gradient signals and how those signals shape the shared policy, thereby affecting the severity of LST. We therefore conduct a series of controlled experiments to test these hypotheses. We refer to the configuration used in Section 2 as ALL-4K, which trains on the full DAPO-Math-17K mixture with a 4k rollout budget. ALL-8K uses the same training mixture but doubles the rollout budget to 8k. ALL-4K-8K first trains with a 4k budget and then continues training the resulting RL checkpoint with an 8k budget. Finally, HARD-4K retains the 4k budget but restricts training to hard prompts that the initial base model solves in at most four out of eight rollouts.

Table 1 reports five anchor-based LST scores and the overall Pass@1 improvement under four training settings.

① **HARD-4K produces the highest LST across all easy sets.** It suggests that training on harder data causes stronger behavioral spillover to already-solved prompts.

② **A larger rollout budget amplifies LST when used from the start, but mitigates it when introduced later in training.** At step 240, ALL-8K has a much higher LST than ALL-4K across all anchors. A larger budget therefore accelerates LST early in training. The later trend is different. At step 400, ALL-4K-8K has a lower LST than continued 4k training and achieves a larger Pass@1 improvement. This does not mean that the 8k setting produces shorter responses overall. Its average responses and hard-query responses remain longer, but length growth on easy queries becomes slower.

Why outcome-only RL does not correct the drift. We explain LST through the policy-gradient signal produced by RLVR. Consider a rollout group $\{y_i\}_{i=1}^{G}$. Its group-relative advantages vanish when all responses receive the same correct reward:

$$R_1 = \cdots = R_G \quad \Longrightarrow \quad A_1 \approx \cdots \approx A_G \approx 0. \tag{7}$$

Let $g_{\mathcal{E}}$ and $g_{\mathcal{H}}$ denote the expected update directions from easy and hard prompts. Let $\rho_{\mathcal{E}}$ and $\rho_{\mathcal{H}}$ denote their sampling weights. The update on the mixed training distribution is

$$g_{\text{mix}} = \rho_{\mathcal{E}} g_{\mathcal{E}} + \rho_{\mathcal{H}} g_{\mathcal{H}} \approx \rho_{\mathcal{H}} g_{\mathcal{H}}. \tag{8}$$

Thus, hard prompts dominate the update after easy groups become saturated.

However, a zero gradient from easy prompts does not keep their output distributions fixed. All prompts share the same policy parameters. An update from hard prompts can therefore change the

The extra tokens are not valuable. We further analyze 5,824 responses from the 26 queries in the step-100 easy set $\mathcal{E}_{100}$. Following (Xu et al., 2026), we construct a lexicon of reflection words that capture explicit hesitation, verification, and self-correction. We additionally compute the repeated bigram rate to quantify local phrase repetition within each response. Figure 3 shows that, as training proceeds, the growth in response length is accompanied by a substantial increase in reflection words and repeated bigrams, which is not a desirable behavior for easy queries.

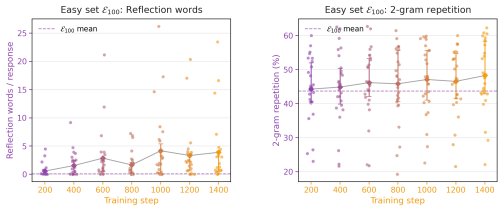


Figure 3: **Observable behavior on the fixed easyset.** Reflection word frequency and repeated-bigram rate in responses generated by subsequent checkpoints for the same queries in $\mathcal{E}_{100}$.

Table 1: **Controlled comparison of budget and training-data effects.** LST is computed with the ALL-4K fixed-easy reference. Gray rows are the corresponding ALL-4K anchor baselines. Each non-anchor LST entry reports the value at $t^\star$, with the colored arrow showing its change from the corresponding anchor baseline. Red denotes LST increase and green denotes decrease. ΔPass@1 is full-test-set Pass@1 improvement over the base model.

Setting	Budget	$t^\star$	$\mathrm{LST}_0(t^\star)$	$\mathrm{LST}_{25}(t^\star)$	$\mathrm{LST}_{50}(t^\star)$	$\mathrm{LST}_{75}(t^\star)$	$\mathrm{LST}_{100}(t^\star)$	ΔPass@1
ALL-4K	4096	240	21.0	18.1	16.7	13.7	16.3	+9.1 pp
ALL-8K	8192	240*	25.5 ↑4.5	40.9 ↑22.8	38.0 ↑21.3	36.5 ↑22.8	37.4 ↑21.1	+9.7 pp
ALL-4K	4096	400	42.9	44.6	43.0	40.0	43.4	+11.8 pp
ALL-4K-8K	4096→8192	100†	33.7 ↓9.2	42.2 ↓2.4	38.8 ↓4.2	37.2 ↓2.8	39.2 ↓4.2	+12.6 pp
ALL-4K-8K	4096→8192	400	31.4 ↓11.5	42.6 ↓2.0	39.3 ↓3.7	34.8 ↓5.2	39.3 ↓4.1	**+15.0** pp
HARD-4K	4096	400	**53.9** ↑11.0	**58.0** ↑13.4	**54.7** ↑11.7	**49.7** ↑9.7	**54.7** ↑11.3	+11.8 pp

*For ALL-8K, we only report step 240 because the run begins to collapse around step 250. †ALL-4K-8K continues for 100 steps from the ALL-4K step-300 checkpoint, making it comparable to ALL-4K at step 400.

token probabilities at an easy prefix $s^{\mathcal{E}}$. To first order,

$$\Delta \log \pi_\theta(a \mid s^{\mathcal{E}}) \approx \eta \rho_{\mathcal{H}} \nabla_\theta \log \pi_\theta(a \mid s^{\mathcal{E}})^\top g_{\mathcal{H}}, \tag{9}$$

which is generally nonzero even when $g_{\mathcal{E}} \approx 0$.

4 LENGTH SELF-DISTILLATION

In this section, we introduce Length Self-Distillation (LSD). It adds two components to the original RL trainer: an online difficulty router and a temporal self-teacher.

Online routing. Online routing introduces a practical challenge. Ideally, if the prompt in the training batch can be reliably solved by the earlier teacher policy, it should be routed to the OPD objective to preserve the teacher's concise behavior. However, applying this rule directly would require additional teacher rollouts and would roughly double the rollout cost. For efficient training, we use the empirical solve rate of the student's on-policy rollout group as a lightweight routing signal. To examine the validity of this proxy, we select the easy set at step 500 with thresholds $\tau \in \{0.8, 0.9, 1.0\}$, and trace the same prompts back through previous checkpoints to check whether earlier policies would also classify them as easy. The result in Figure 4 suggests that most of these prompts already have high historical accuracy regardless of the chosen easy-set threshold. Even when using the step-250 checkpoint as the teacher for the step-500 policy, over 87% of the prompts classified as easy at step-500 are also classified as easy at step-250.

Temporal self-teacher. The simplest self-teacher is the pre-RL policy π_0. It provides a clean behavioral anchor because its easy-query responses have not yet accumulated LST. However, distillation from this fixed teacher can impede further capability acquisition. Since the router uses the current student's solve rate, it may route newly solved prompts to OPD even when π_0 cannot solve them reliably. Figure 4 illustrates the underlying historical mismatch. Using π_0 as the teacher creates a

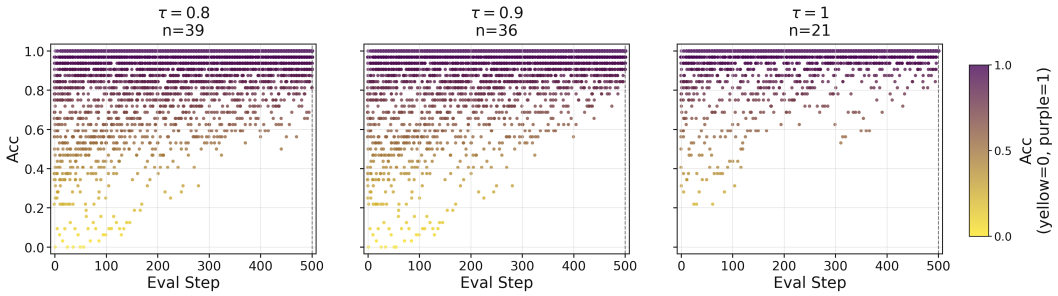


Figure 4: Historical accuracy of prompts selected as easy at step 500. Each point is one prompt at one previous checkpoint. Yellow indicates $A(x) = 0$ and purple indicates $A(x) = 1$.

substantial mismatch between the prompts routed to OPD and those the teacher can solve. Strong distillation toward this fixed policy can therefore limit benchmark improvement.

We address this problem with an **exponential moving average (EMA) teacher** $\bar{\pi}_k$:

$$\bar{\theta}_k \leftarrow \beta\bar{\theta}_{k-1} + (1-\beta)\theta_k, \tag{10}$$

where θ_k and $\bar{\theta}_k$ denote the online policy and the EMA teacher policy's parameters in training step k. The coefficient $\beta \in [0, 1]$ controls the temporal lag of the EMA teacher. A larger β keeps the teacher closer to past policies and provides a stronger behavioral anchor, while a smaller β lets it track the online policy more quickly. In our implementation, we set β through a half-life parameter H:

$$\beta = 2^{-1/H}. \tag{11}$$

Thus, the contribution of a past online policy decays exponentially, and its weight is halved after roughly H EMA updates. By adjusting H, we control how far the teacher lags behind the online policy.

Routed LSD objective. Based on the temporal self-teacher, LSD combines online routing with two different optimization objectives. At training step k, the current policy π_k generates G responses for each prompt in the rollout batch $\mathcal{B}_k$. We compute $\widehat{R}(x; \pi_k)$ using these responses and partition the batch into an easy set $\mathcal{E}_k$ and a hard set $\mathcal{H}_k$:

$$\mathcal{E}_k = \left\{ x \in \mathcal{B}_k : \widehat{R}(x; \pi_k) \geq \tau \right\}, \qquad \mathcal{H}_k = \mathcal{B}_k \setminus \mathcal{E}_k. \tag{12}$$

The original RLVR objective is applied to responses from $\mathcal{H}_k$. Responses from $\mathcal{E}_k$ instead receive an OPD objective defined by the temporal self-teacher.

We instantiate the OPD objective for easy groups in three ways, yielding three LSD variants that differ in the direction of the Kullback–Leibler (KL) divergence and how its gradient is computed. ***Supervised-gradient forward KL (SG-FKL)*** directly minimizes the teacher-to-student KL using the teacher's top-K tokens augmented with stop tokens, with the teacher distribution normalized over this support. ***Supervised-gradient reverse KL (SG-RKL)*** instead minimizes the student-to-teacher KL, normalizing both distributions over the same augmented support. Both supervised-gradient variants differentiate the loss directly through the student logits while treating the teacher distribution as fixed. ***Policy-gradient reverse KL (PG-RKL)*** uses the teacher-minus-rollout-policy log-probability difference at each sampled token as a detached advantage in a proximal policy optimization (PPO)-style objective. Thus, SG-FKL and SG-RKL provide supervision over the full retained support, whereas PG-RKL updates the policy through sampled actions. All three variants retain the original group-relative RL objective for hard groups. Appendix C gives the full losses, stop-token treatment, and importance-ratio definitions.

Let $\bar{\ell}^{\mathrm{RL}}$ and $\bar{\ell}^{\mathrm{OPD}}$ be sequence-mean losses on the hard and easy routes, with $n_{\mathcal{H}}$ and $n_{\mathcal{E}}$ sampled sequences, respectively. The number of sequences in each route naturally determines its contribution to the loss. We therefore weight the two route-level losses by their respective numbers of response sequences and get the final objective of LSD:

$$\mathcal{L}_{\mathrm{LSD}} = \frac{n_{\mathcal{H}}}{n_{\mathcal{H}} + n_{\mathcal{E}}} \bar{\ell}^{\mathrm{RL}} + \frac{n_{\mathcal{E}}}{n_{\mathcal{H}} + n_{\mathcal{E}}} \bar{\ell}^{\mathrm{OPD}}. \tag{13}$$

5 EXPERIMENTS

5.1 SETUP AND COMPARISON PROTOCOL

Unless otherwise specified, all main LSD experiments use the training-time routing threshold $\tau = 1$. Full configurations are in Appendix F.

Single-Turn Reasoning Task. We post-train Qwen3-4B-Base on deduplicated DAPO-Math-17K and evaluate AMC 2023 and AIME 2025–2026 with 32 responses per query and a 4k response budget. We compare RL, the three LSD variants, CRISP (Sang et al., 2026), and Fixed SG-FKL. CRISP distills a periodically refreshed, concise-prompted teacher using reverse KL on all rollouts. Fixed uses a frozen π_0 teacher, a fixed routing map with threshold 1, and an LSD coefficient of 1. SG-FKL and SG-RKL use $K = 32$; EMA updates start after four actor updates. Fixed-easy-set evaluation thresholds are separate from the training routing threshold.

Multi-Turn Agentic Task. We follow Wu et al. (2025) and post-train Qwen3-8B-Base on the CutTheBill training split, evaluating it on BrowseComp-Plus (Chen et al., 2025). We use a 20,000-token response budget, at most 48 turns, and a separate Qwen3-8B refinement agent. We compare RL, the three LSD variants, and RL + Length Penalty under the same environment and evaluation configuration. Following AdapThink (Xu et al., 2026), RL + Length Penalty applies stronger length penalties to easier queries. We report Pass@1, average turns, and LST_{50}. For the agent setting, response length in Eq. 6 is the sum of policy-generated tokens across turns; tool observations and refinement-model generation are separate cost components.

5.2 MAIN RESULTS

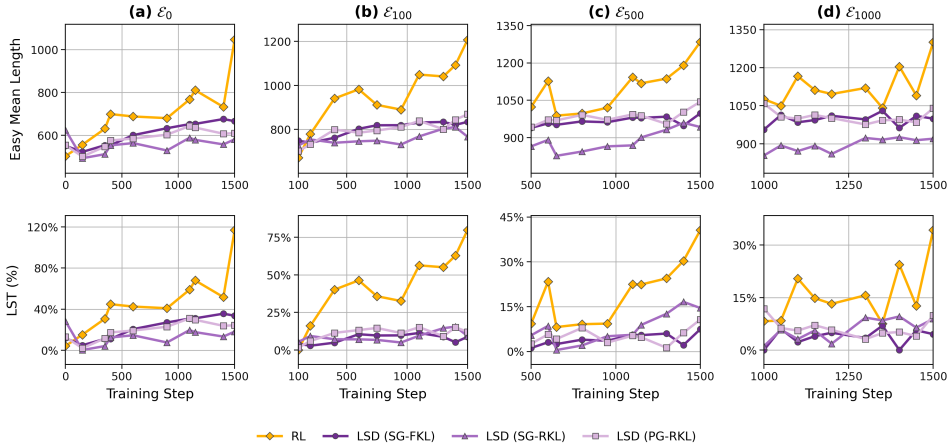
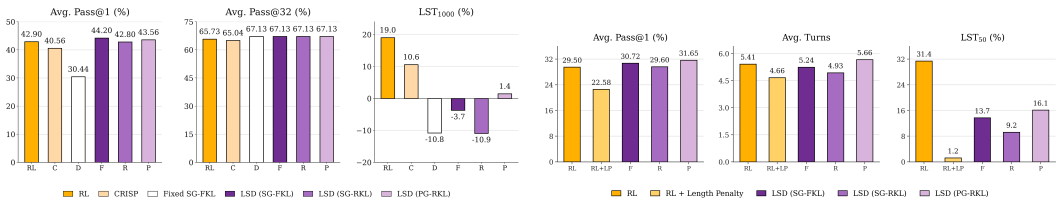


Figure 5: **Length growth on fixed easy queries.** Across four anchor-defined sets, all three LSD variants have lower easy-query length and LST than RL through most of training. The upper panels report mean length and the lower panels report LST. Different anchors select different cohorts.

Maintaining concise reasoning on easy queries. Figure 5 shows that LSD curbs easy-query length growth across anchors. Appendix D jointly reports accuracy and length on identical frozen query sets. Quantitatively, Figure 6a and Table 2 show that, on single-turn reasoning, LST_{1000} decreases from 19.0% under RL to -3.7%, -10.9%, and 1.4% under SG-FKL, SG-RKL, and PG-RKL, respectively. The same pattern extends to multi-turn agentic tasks. As reported in Figure 6b and Table 3 in Appendix A, LST_{50} decreases from 31.4% under RL to 13.7%, 9.2%, and 16.1% under SG-FKL, SG-RKL, and PG-RKL, respectively.

Encouraging more reasoning on hard queries. The reduction in easy-query length does not extend to the hard-query cohort. Table 2 reports results on a fixed easy-query set and its hard-query complement. Average hard-query length increases from 2329 tokens under RL to 2511, 2395, and



(a) Single-turn reasoning tasks.

(b) Multi-turn agentic tasks.

Figure 6: **Capability and length-scaling tax across tasks.** (a) Single-turn average Pass@1, Pass@32, and LST_{1000}. (b) BrowseComp-Plus average Pass@1, turns, and LST_{50}. C and D denote CRISP and Fixed SG-FKL; F, R, and P denote LSD with SG-FKL, SG-RKL, and PG-RKL. RL+LP denotes RL + Length Penalty.

2393 tokens under SG-FKL, SG-RKL, and PG-RKL, respectively. All three variants therefore produce shorter responses on easy queries while allowing longer responses on hard queries.

The training allocation shows a complementary pattern. As shown in Figure 12, during training after step 500, the easy route accounts for only 12.17%, 10.58%, and 10.94% of tokens under SG-FKL, SG-RKL, and PG-RKL, respectively. The hard route therefore retains 87.83–89.42% of the logged token share on average. This allocation keeps training primarily focused on hard queries while preserving concise response patterns on easy queries.

Comparing efficiency baselines. CRISP yields 40.56% Pass@1; Fixed SG-FKL yields 30.44% Pass@1 and -10.8% LST_{1000} (Figure 6a; Table 2). On multi-turn tasks, RL + Length Penalty reduces LST_{50} to 1.2% and average turns to 4.66, but lowers Pass@1 to 22.58% (Table 3).

Comparing the three KL objectives. Overall, the three KL objectives differ in how strongly they preserve existing behavior and how they distribute probability across candidate responses. Their Pass@1 scores vary slightly, while their Pass@32 scores are nearly identical. Among the three EMA variants, SG-RKL achieves the lowest LST, but its stronger easy-query compression accompanies lower Pass@1.

The loss definitions offer a possible explanation. SG-FKL weights discrepancies by teacher probabilities, whereas SG-RKL penalizes student mass on tokens assigned low teacher probability. With a lagged teacher, the latter may more strongly preserve established behavior. As shown in Figure 11 and Table 6, SG-RKL has lower mean actor entropy (0.0367) than SG-FKL (0.0743) and PG-RKL (0.0542), together with the smallest mean absolute teacher–rollout log-probability difference before updates. Meanwhile, PG-RKL updates sampled actions using the original token probabilities, whereas SG-RKL directly optimizes distributions renormalized on the retained support. This distinction may also contribute to their different outcomes.

5.3 Ablations

In this section, we ablate two additional hyperparameters introduced by LSD: the EMA half-life H and the routing threshold τ. All ablations use the SG-FKL variant, which achieves the highest average Pass@1 on single-turn reasoning among the three LSD variants.

EMA half-life. Figure 7 compares $H \in \{2, 4, 8\}$ with $\tau = 1$. A longer half-life generally suppresses LST but slows capability acquisition. Among the tested settings, $H = 4$ achieves the highest average Pass@1 (41.85%) at step 1750 (Table 4), indicating that keeping the teacher closer to the online policy is not always beneficial. As shown in Figure 9 and Table 5, we compare teacher–student parameter lag, distillation loss, and token routing across half-lives. Larger H produces a greater parameter lag and higher distillation loss, while a smaller share of tokens is routed to OPD. The intermediate lag at $H = 4$ may provide a stable behavioral reference while allowing the teacher to track improvements in the online policy.

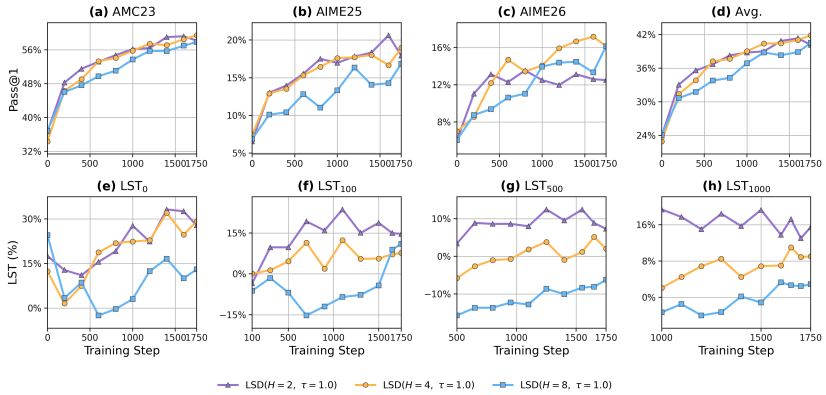


Figure 7: **EMA half-life ablation.** We compare $H = 2, 4, 8$ at $\tau = 1.0$ through step 1750. The upper row shows benchmark and aggregate Pass@1; the lower row shows LST for anchors $b = 0, 100, 500, 1000$.

Scope of difficulty routing. We examine whether preservation should extend to partially solved queries by varying $\tau \in \{1.0, 0.85, 0.75\}$ with SG-FKL and $H = 4$. Lowering τ routes more partially solved groups to distillation. At step 1750, average Pass@1 decreases from 41.85% to 40.91% and 39.31%, respectively (Figure 8; Table 4). Meanwhile, the mean distillation token share increases from 10.31% to 19.00% (Figure 10; Table 5). These results support restricting preservation to fully solved rollout groups: such groups lack a group-relative reward signal, whereas partially solved groups still provide reward variation for continued RL learning. Extending preservation before rollout accuracy saturates can therefore compromise capability acquisition. Moreover, to assess the role of query selection, we compare LSD with count-matched random routing under the same EMA and distillation settings (Appendix D, Table 9). See Appendix B for the interpretation of negative LST values.

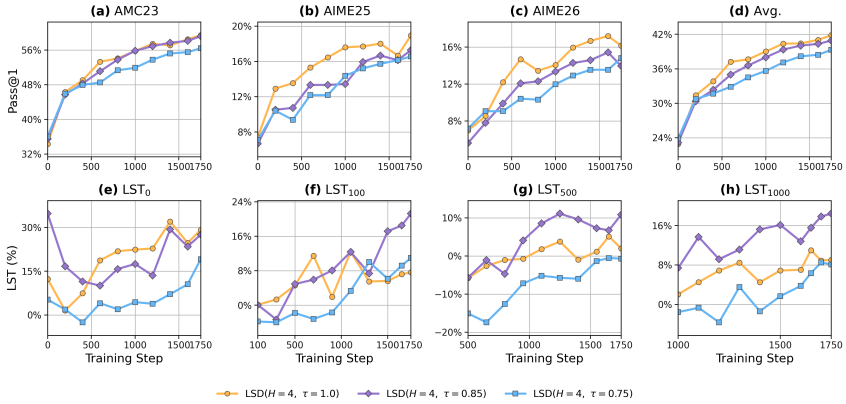


Figure 8: **Routing-threshold ablation.** We compare $\tau = 1.0, 0.85, 0.75$ at $H = 4$ through step 1750. Panels follow Figure 7.

6 Conclusion

In this paper, we identify LST as a conditional cost of RL post-training that arises when responses to already-solved queries lengthen without corresponding accuracy gains. LSD restores a token-level preservation signal on solved rollout groups while retaining RL on unsolved groups, using online accuracy-based routing and an EMA self-teacher. We instantiate LSD with three objectives and demonstrate reduced easy-query cost with strong benchmark performance on single-turn mathematical reasoning and multi-turn agentic tasks. Future work will evaluate LSD on larger models and explore how to extract denser learning signals from sampled data for more effective supervision.

AI USE STATEMENT

We used generative AI tools for translation, language editing, literature synthesis, analysis and plotting code, statistical checks, and discussions of methodology, experimental design, and result interpretation. The authors reviewed the AI-assisted outputs and take responsibility for the final content, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

Training configurations, objectives, and evaluation protocols are documented in Section 5 and Appendices A–D and F. We will release our code after paper acceptance.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

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

Anthropic. Claude's extended thinking. https://www.anthropic.com/news/visible-extended-thinking, 2025.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 700–719, 2026.

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 o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, et al. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*, 2025.

Google. Gemini 2.5 thinking model updates. https://developers.googleblog.com/gemini-2-5-thinking-model-updates/, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Haotian Luo, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Xiaochun Cao, Dacheng Tao, Naiqiang Tan, and Li Shen. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14242–14257, 2026.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

OpenAI. Introducing openai o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/, 2025.

Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Weijie Liu, Clive Bai, Kai Yang, Yangkun Chen, Saiyong Yang, and Xiangyang Ji. Small generalizable prompt predictive models can steer efficient rl post-training of large reasoning models. *arXiv preprint arXiv:2602.01970*, 2026.

Jiacheng Ruan, Jun Tang, Wenzhen Yuan, Ting Liu, Shuai Bai, Dayiheng Liu, Zhibo Yang, and Yuzhuo Fu. Contrastive on-policy distillation. *arXiv preprint arXiv:2607.19046*, 2026.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026.

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

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2322–2331, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Xu Wan, Yansheng Wang, Wenqi Huang, and Mingyang Sun. Buffer matters: Unleashing the power of off-policy reinforcement learning in large language model reasoning. *arXiv preprint arXiv:2602.20722*, 2026a.

Xu Wan, Yansheng Wang, Wenqi Huang, and Mingyang Sun. Buffer matters: Unleashing the power of off-policy reinforcement learning in large language model reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Xu Wan, Speed Zhu, Jianwei Cai, Guang Chen, XiMing Huang, Wiggin Zhou, and Mingyang Sun. The shadow price of reasoning: Economic perspective on optimal budget allocation for LLMs. *arXiv preprint arXiv:2606.03092*, 2026c.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Jiahao Wu, Zhongwen Xu, Qiang Fu, and Wei Yang. Cut the bill, keep the turns: Affordable multi-turn search RL. Tencent TEG AIPD Technical Report, December 2025. URL https://agate-slipper-ef0.notion.site/Cut-the-Bill-Keep-the-Turns-Affordable-Multi-Turn-Search-RL-003f78214a4d451fb06f453d084e666c. Accessed: 2026-09-04.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*, 2025.

Wenyue Xu, Xu Wan, Wei Wang, Wenqi Huang, Wotao Yin, Shengjie Zhao, and Mingyang Sun. Adapthink: Adaptive thinking preferences for reasoning language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 9808–9825, 2026.

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

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

Jingyang Yi, Jiazheng Wang, and Sida Li. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *Advances in Neural Information Processing Systems*, 38: 39011–39043, 2026.

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. *arXiv preprint arXiv:2503.14476*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Danlong Yuan, Tian Xie, Shaohan Huang, Huishuai Zhang, Zhuocheng Gong, Chong Luo, Furu Wei, and Dongyan Zhao. Shorten after you're right: Lazy length penalties for reasoning rl. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12864–12877, 2026.

A ADDITIONAL BENCHMARK RESULTS

A.1 SINGLE-TURN REASONING

Table 2: **Evaluation results for single-turn reasoning.** Panel (a) reports accuracy in percent; panel (b) reports the mean token counts, and LST. Avg. Acc. weights benchmarks by their numbers of evaluation prompts. Avg. Len. covers all 143 prompts. Easy and Hard Len. use the fixed RL step-100 easy set ($A(x) \geq 0.9$) and its hard-query complement. LST_{1000} uses the $\tau = 1$ RL step-1000 cohort in Table 8. On this fixed easy set, RL step 885 has the lowest mean response length among RL checkpoints satisfying the accuracy constraint (100%). Its mean length, 853.716 tokens, is the shared reference for all methods.

(a) Task performance (%)

	AMC		AIME25		AIME26		Avg. Acc.	
Method	Pass@1	Pass@32	Pass@1	Pass@32	Pass@1	Pass@32	Pass@1	Pass@32
RL	60.09	84.34	22.19	**43.33**	16.04	36.67	42.90	65.73
CRISP	58.48	86.75	18.36	36.67	13.18	33.33	40.56	65.04
Fixed SG-FKL	45.07	**87.95**	11.46	40.00	8.96	36.67	30.44	**67.13**
LSD (SG-FKL)	**61.30**	86.75	**22.92**	**43.33**	**18.15**	36.67	**44.20**	**67.13**
LSD (SG-RKL)	60.87	**87.95**	19.65	36.67	15.94	**40.00**	42.80	**67.13**
LSD (PG-RKL)	60.70	**87.95**	21.96	**43.33**	17.76	33.33	43.56	**67.13**

(b) Response length and LST

Method	Avg. Len.	Easy Len.	Hard Len.	LST_0 (%)	LST_{1000} (%)
RL	2124	997	2329	67.2	19.03
CRISP	2150	972	2364	61.2	10.60
Fixed SG-FKL	**1226**	**691**	**1323**	11.0	−10.81
LSD (SG-FKL)	2257	859	2511	31.5	−3.72
LSD (SG-RKL)	2143	757	2395	**9.1**	**−10.92**
LSD (PG-RKL)	2148	802	2393	14.0	1.45

The fixed hard-query cohort is the complement of the evaluation easy set; it is distinct from the dynamically routed hard groups used during training.

A.2 MULTI-TURN AGENTIC TASKS

Table 3: **Evaluation results on multi-turn agentic tasks.** Values correspond to Figure 6b. Pass@1 and LST are in percent; lengths are in tokens.

Method	BrowseComp-Plus						
	Pass@1	Avg. Turns	Easy Len.	Hard Len.	LST_0	LST_{50}	LST_{100}
RL	29.50	5.41	3531.9	4065.3	42.3	31.4	11.8
RL + Length Penalty	22.58	**4.66**	2708	3426		**1.2**	
LSD (SG-FKL)	30.72	5.24	2706.0	4252.0	24.9	13.7	1.2
LSD (SG-RKL)	29.60	4.93	**2146.5**	4034.8	**13.6**	9.2	**−6.2**
LSD (PG-RKL)	**31.65**	5.66	2715.2	4139.5	31.4	16.1	2.4

SG-FKL and SG-RKL reduce both LST and average interaction turns while increasing Pass@1. PG-RKL attains the highest Pass@1 but uses more turns than RL, showing that lower LST does not necessarily imply fewer interactions. RL + Length Penalty obtains the lowest LST_{50} and fewest turns, but its Pass@1 falls to 22.58% from RL's 29.50%.

B ABLATION REPORTING DETAILS

The ablation trajectories are shown in Figures 7 and 8 in Section 5.3.

The ablations use a fixed, accuracy-qualified reference length from aligned RL, shared across configurations. Negative LST indicates responses shorter than this reference; accuracy preservation is evaluated separately on the same fixed easy set.

Table 4: **Ablation endpoint summary at step 1750.** Avg. Pass@1 is weighted over all 143 evaluation prompts. LST is computed against the fixed aligned-RL accuracy-qualified reference. The $H = 4, \tau = 1.0$ run is shared by both comparisons.

H	τ	Avg. Pass@1 (%)	LST_0 (%)	LST_{100} (%)	LST_{500} (%)	LST_{1000} (%)
2	1.00	40.14	27.79	14.64	7.29	15.49
4	1.00	41.85	29.20	7.63	2.01	9.06
8	1.00	40.49	13.01	11.00	-6.21	2.91
4	0.85	40.91	27.55	21.24	10.84	18.42
4	0.75	39.31	19.15	10.97	-0.68	8.11

We compare all five configurations over the shared steps 1–1750. Table 5 reports arithmetic means of logged per-step scalars; the Pass@1 column instead gives the unsmoothed evaluation at step 1750. The fresh $H = 4, \tau = 1.0$ run is used in both ablations.

Table 5: **LSD training signals in the H and τ ablations.** Sequence and token shares refer to the easy route. Parameter gap is the logged maximum absolute teacher–student parameter difference before the EMA update. Training statistics average 1750 steps per run.

H	τ	Pass@1 (%)	Easy seq. (%)	Easy tok. (%)	Effective OPD coef.	Param. gap ($\times 10^{-5}$)	OPD loss ($\times 10^{-3}$)
2	1.00	40.14	18.87	10.79	0.252	0.690	0.614
4	1.00	41.85	18.54	10.31	0.247	1.087	0.693
8	1.00	40.49	15.24	7.90	0.192	1.690	0.841
4	0.85	40.91	24.67	14.87	0.354	1.077	0.715
4	0.75	39.31	29.50	19.00	0.450	1.078	0.689

B.1 TEACHER LAG AND DISTILLATION EXPOSURE

Increasing H produces a larger teacher parameter lag and a larger logged SG-FKL loss, while reducing the fraction of tokens sent to OPD. Thus, the stronger preservation observed for $H = 8$ does not require greater distillation exposure. Instead, the delayed teacher can impose a stronger constraint on the tokens it supervises. The maximum parameter gap is a parameter-space diagnostic, not a KL divergence; its ordering need not match the teacher–rollout log-probability gap.

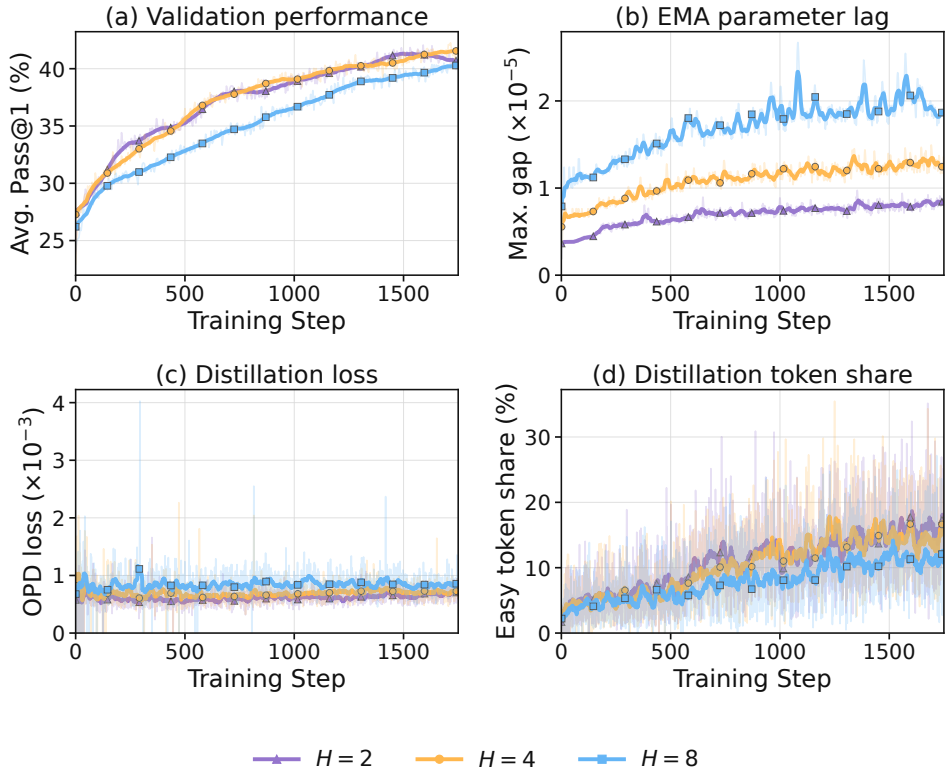


Figure 9: **Training signals behind the EMA half-life ablation.** All runs use SG-FKL and $\tau = 1$. Faint lines show raw values; marked curves show centered 21-point moving means, using available points at the boundaries. Evaluation points are five training steps apart; the other metrics are logged every training step.

B.2 Routing threshold and preservation pressure

With eight responses per group, thresholds $\tau = 1$, 0.85, and 0.75 admit groups with at least eight, seven, and six correct responses, respectively. Lowering the threshold therefore moves more partially solved groups to distillation. As shown in Figure 10, both the easy-route sequence share and token share increase, along with the logged effective OPD coefficient. From $\tau = 1$ to 0.75, the mean token share rises from 10.31% to 19.00%, while the effective coefficient rises from 0.247 to 0.450. The lower Pass@1 at step 1750 is consistent with more preservation pressure on prompts that still admit incorrect responses. These coupled changes do not isolate which training signal causes the performance difference.

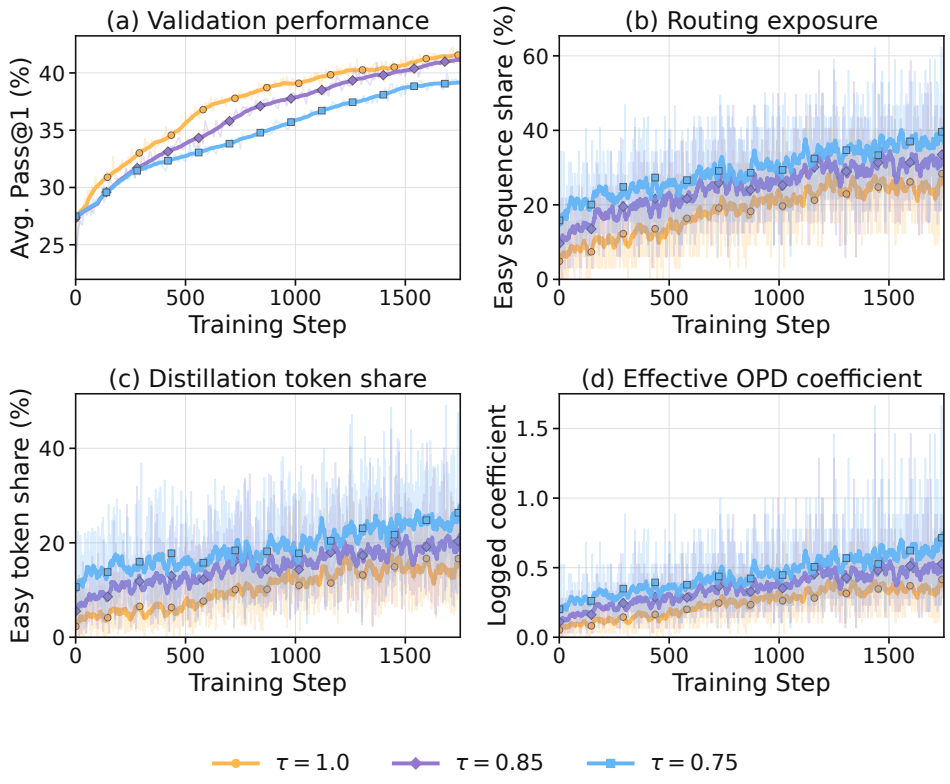


Figure 10: **Training signals behind the routing-threshold ablation.** All runs use SG-FKL and $H = 4$, over steps 1–1750. Raw and smoothed curves follow the convention in Figure 9. Lower thresholds increase the number of sequences and fraction of tokens routed to distillation, as well as the effective OPD coefficient.

C Detailed RL and distillation objectives

The notation and losses below expand the routed objective in Section 4.

For each hard prompt x, let $\{y_j\}_{j=1}^{G}$ denote its rollout group and let $r_j = r(x, y_j)$ be the reward of response y_j. We compute the group-relative advantage as

$$A_i = \frac{r_i - \bar{r}_x}{\sigma_x + \epsilon_{\text{adv}}}, \qquad \bar{r}_x = \frac{1}{G}\sum_{j=1}^{G} r_j, \qquad \sigma_x = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \bar{r}_x)^2}. \tag{14}$$

For a response y_i of length T_i, let $s_{i,t} = (x, y_{i,<t})$ denote its prefix at token t. The importance ratio is

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid s_{i,t})}{\pi_{\text{old}}(y_{i,t} \mid s_{i,t})}. \tag{15}$$

Its sequence-level RL loss is

$$\ell_i^{\mathrm{RL}} = -\frac{1}{T_i}\sum_{t=1}^{T_i} \min\left(\rho_{i,t}(\theta)A_i,\ \mathrm{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)A_i\right). \tag{16}$$

For an easy response y_i, we consider three OPD variants. Two directly optimize a top-K KL objective, while the third estimates reverse KL through policy gradients.

Supervised-gradient forward KL (SG-FKL). Let $\mathcal{V}_{i,t}^K$ denote the teacher's top-K tokens at prefix $s_{i,t}$. To ensure that termination behavior is supervised even when a stop token does not appear in the teacher's top-K predictions, we augment this set with the stop-token set $\mathcal{V}_{\mathrm{stop}}$:

$$\widetilde{\mathcal{V}}_{i,t}^K = \mathcal{V}_{i,t}^K \cup \mathcal{V}_{\mathrm{stop}}. \tag{17}$$

We normalize the teacher distribution over the augmented support:

$$\bar{\pi}_k^K(v \mid s_{i,t}) = \frac{\bar{\pi}_k(v \mid s_{i,t})}{\sum_{u\in\widetilde{\mathcal{V}}_{i,t}^K} \bar{\pi}_k(u \mid s_{i,t})}, \qquad v \in \widetilde{\mathcal{V}}_{i,t}^K. \tag{18}$$

Then, the forward-KL OPD objective is

$$\ell_i^{\text{SG-FKL}} = \frac{1}{T_i}\sum_{t=1}^{T_i}\sum_{v\in\widetilde{\mathcal{V}}_{i,t}^K} \bar{\pi}_k^K(v \mid s_{i,t})\left[\log \bar{\pi}_k^K(v \mid s_{i,t}) - \log \pi_\theta(v \mid s_{i,t})\right]. \tag{19}$$

Gradients are taken directly through the student logits, while the teacher distribution is detached.

Supervised-gradient reverse KL (SG-RKL). For the reverse direction, we also normalize the student distribution over the same augmented support:

$$\pi_\theta^K(v \mid s_{i,t}) = \frac{\pi_\theta(v \mid s_{i,t})}{\sum_{u\in\widetilde{\mathcal{V}}_{i,t}^K} \pi_\theta(u \mid s_{i,t})}, \qquad v \in \widetilde{\mathcal{V}}_{i,t}^K. \tag{20}$$

We then directly minimize

$$\ell_i^{\text{SG-RKL}} = \frac{1}{T_i}\sum_{t=1}^{T_i}\sum_{v\in\widetilde{\mathcal{V}}_{i,t}^K} \pi_\theta^K(v \mid s_{i,t})\left[\log \pi_\theta^K(v \mid s_{i,t}) - \log \bar{\pi}_k^K(v \mid s_{i,t})\right]. \tag{21}$$

Policy-gradient reverse KL (PG-RKL). The third objective only uses the teacher probability of the sampled token. For each token, we define the detached OPD advantage

$$A_{i,t}^{\mathrm{OPD}} = \mathrm{sg}\left[\log \bar{\pi}_k(y_{i,t} \mid s_{i,t}) - \log \pi_{\mathrm{old}}(y_{i,t} \mid s_{i,t})\right], \tag{22}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient. We then apply the PPO-style clipped objective:

$$\ell_i^{\text{PG-RKL}} = -\frac{1}{T_i}\sum_{t=1}^{T_i} \min\left(\rho_{i,t}(\theta)A_{i,t}^{\mathrm{OPD}}, \mathrm{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)A_{i,t}^{\mathrm{OPD}}\right). \tag{23}$$

Let ℓ_i^{OPD} denote the OPD loss used by a particular LSD variant. Under sequence-mean–token-mean aggregation, we first average the token losses within each response. We then average the resulting sequence-level losses within their assigned routes:

$$\bar{\ell}^{\mathrm{RL}} = \frac{1}{n_{\mathcal{H}}}\sum_{i\in\mathcal{Y}_{\mathcal{H}}} \ell_i^{\mathrm{RL}}, \qquad \bar{\ell}^{\mathrm{OPD}} = \frac{1}{n_{\mathcal{E}}}\sum_{i\in\mathcal{Y}_{\mathcal{E}}} \ell_i^{\mathrm{OPD}}, \tag{24}$$

where $\mathcal{Y}_{\mathcal{H}}$ and $\mathcal{Y}_{\mathcal{E}}$ contain the response sequences routed to RLVR and OPD, respectively, and $n_{\mathcal{H}} = |\mathcal{Y}_{\mathcal{H}}|$ and $n_{\mathcal{E}} = |\mathcal{Y}_{\mathcal{E}}|$.

C.1 Logged optimization diagnostics

We examine the logged training histories of the three single-turn LSD runs. The shared training interval is steps 501–1628: SG-RKL's available training history begins at step 501, and PG-RKL's ends at step 1628. All three configurations record an EMA half-life of four updates, an EMA update interval of one, and an online routing threshold of one.

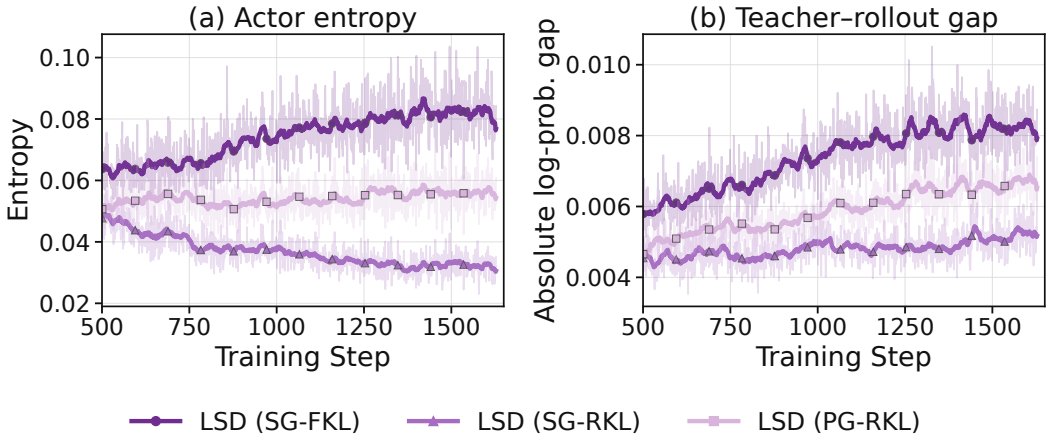


Figure 11: **Training dynamics of the three LSD objectives.** Actor entropy (left) and the mean absolute teacher–rollout log-probability difference before updates (right) over the shared steps 501–1628. Faint traces show raw per-step values; marked curves show centered 21-step means, using available points at the boundaries. SG-RKL exhibits lower entropy and a smaller teacher–rollout discrepancy over this interval. Each trajectory uses its own run's training batches.

Table 6: **Optimization diagnostics over a shared training interval.** Values are arithmetic means of the logged per-step scalars over all 1128 shared steps, without smoothing. These training steps are not independent experimental replicates.

Variant	Actor entropy	Teacher–rollout gap	Easy sequences (%)
SG-FKL	0.0743	0.00738	21.51
SG-RKL	0.0367	0.00481	20.23
PG-RKL	0.0542	0.00590	20.67

C.2 Easy and hard token allocation during training

The training histories directly record the fraction of sequences and tokens assigned to each route. Figure 12 shows their evolution and distribution over the 1128 shared steps 501–1628. The two route fractions sum to one at every recorded step, up to numerical precision.

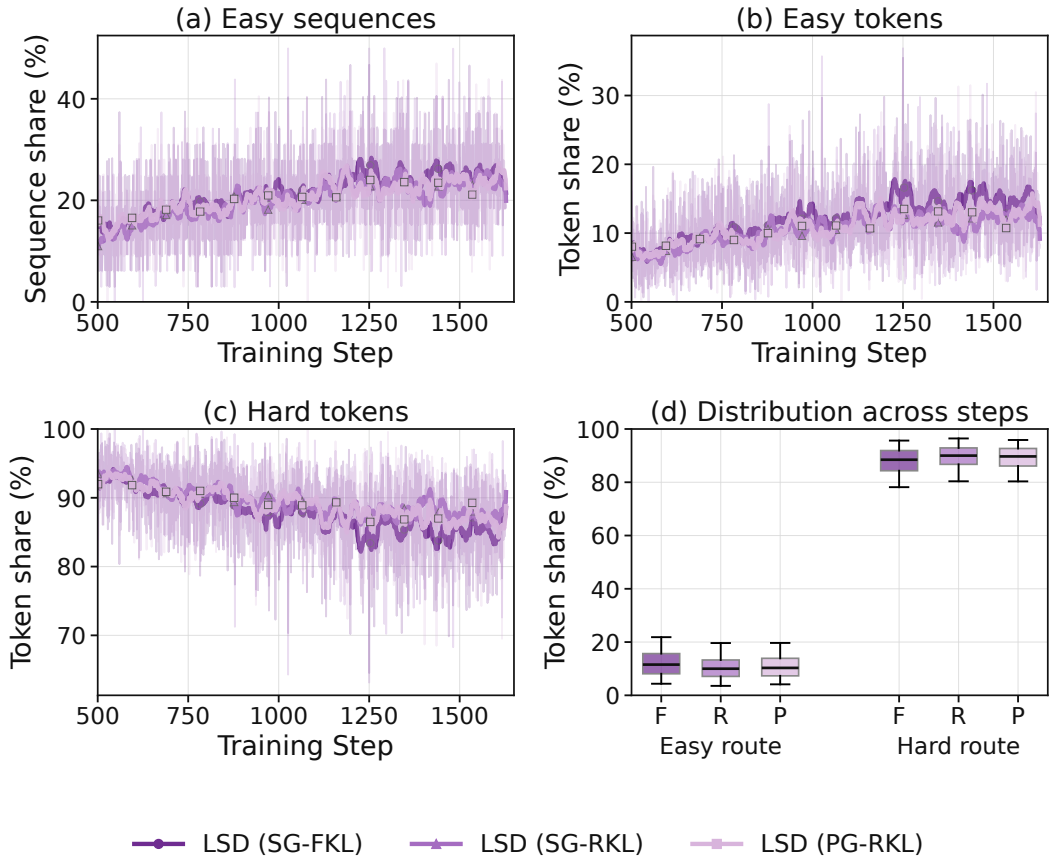


Figure 12: **Training token allocation between the easy and hard routes.** Panels (a)–(c) show raw per-step fractions and centered 21-step means. Panel (d) summarizes the distribution of token shares across training steps: boxes show the interquartile range and median, and whiskers show the 5th and 95th percentiles. F, R, and P denote SG-FKL, SG-RKL, and PG-RKL. These are distributions of per-step shares, not individual response lengths or uncertainty intervals.

Table 7: **Training-route allocation over steps 501–1628.** The first three numeric columns are means of logged per-step fractions, expressed in percent. The last reports the median and interquartile range of the easy-token share.

Variant	Easy seq. (%)	Easy tok. (%)	Hard tok. (%)	Easy tok. median [IQR]
SG-FKL	21.51	12.17	87.83	11.54 [8.09, 15.67]
SG-RKL	20.23	10.58	89.42	9.99 [7.11, 13.27]
PG-RKL	20.67	10.94	89.06	10.29 [7.31, 13.90]

The easy route accounts for a smaller share of tokens than of sequences, while the hard route retains about 88–89% of the token share on average. The values summarize each run's own dynamic routing, not the fixed easy and hard evaluation cohorts in Table 2. The available aggregate logs do not provide individual response lengths by route, so they do not determine a per-response length histogram or a token-count-weighted total across the full training run.

D PAIRED ACCURACY AND LENGTH ON FIXED EASY SETS

Evaluation on identical queries. We freeze query IDs using RL anchors $b \in \{100, 500, 1000\}$ and $\tau = 1$: every selected query has 32/32 correct anchor rollouts.

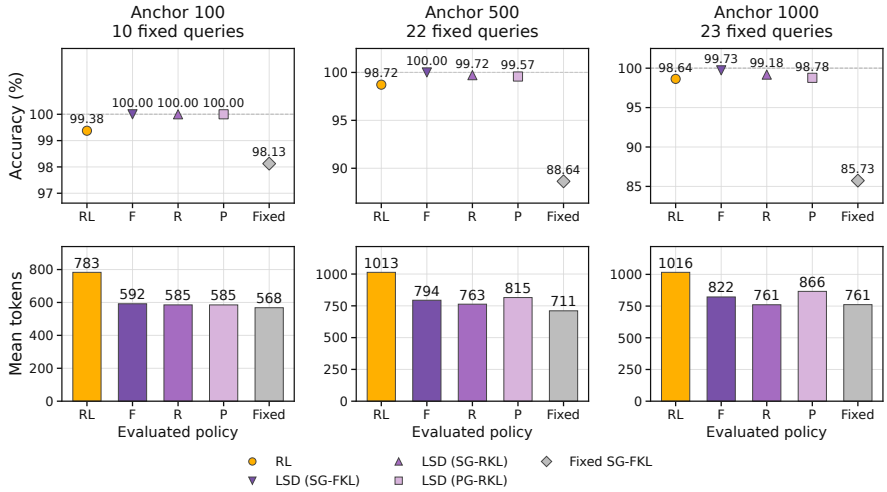


Figure 13: **Accuracy and length evaluated on the same fixed queries.** Each column uses one RL-frozen cohort with $\tau = 1$. The top row reports empirical accuracy; the bottom row reports mean response length. The dashed line marks the 100% anchor accuracy. F, R, and P denote SG-FKL, SG-RKL, and PG-RKL, respectively. Every method is evaluated on the identical query IDs within a column.

Table 8: **Paired evaluation on RL-frozen easy sets ($\tau = 1$).** ΔA is the accuracy difference from RL (pp). For $b = 1000$, the shared reference is RL step 885: 853.716 tokens at 100% accuracy, the minimum over 345 RL checkpoints.

Anchor	Method	Correct / total	Acc. (%)	Mean tokens	ΔA (pp)
$b = 100$	RL	318/320	99.38	783.1	+0.00
	LSD (SG-FKL)	320/320	100.00	592.0	+0.63
	LSD (SG-RKL)	320/320	100.00	585.3	+0.63
	LSD (PG-RKL)	320/320	100.00	585.1	+0.63
	Fixed SG-FKL	314/320	98.13	568.1	-1.25
$b = 500$	RL	695/704	98.72	1013.5	+0.00
	LSD (SG-FKL)	704/704	100.00	794.0	+1.28
	LSD (SG-RKL)	702/704	99.72	763.0	+0.99
	LSD (PG-RKL)	701/704	99.57	815.1	+0.85
	Fixed SG-FKL	624/704	88.64	710.8	-10.09
$b = 1000$	RL	726/736	98.64	1016.2	+0.00
	LSD (SG-FKL)	734/736	99.73	822.0	+1.09
	LSD (SG-RKL)	730/736	99.18	760.5	+0.54
	LSD (PG-RKL)	727/736	98.78	866.1	+0.14
	Fixed SG-FKL	631/736	85.73	761.4	-12.91

All three LSD variants use fewer tokens and have higher reported accuracy than RL on each cohort; all retain 100% accuracy at $b = 100$.

D.1 THE MAIN-TABLE COHORT AND INDIVIDUAL EXAMPLES

Table 9 jointly reports accuracy and length on the easy-query set used in Table 2, using its explicitly relaxed RL step-100 cohort ($\tau = 0.9$). SG-RKL and PG-RKL reach 99.43% and 99.15% accuracy on these same queries, compared with 98.15% for RL, while reducing mean length from 997 to 757 and 802 tokens. Fixed produces still shorter responses but lower accuracy, illustrating why length and correctness must be reported together. Their hard-query accuracies are 32.50% and 33.45%, respectively, compared with 32.85% for RL. SG-FKL reaches 99.10% and 34.22% accuracy on the easy and hard cohorts, with mean lengths of 859 and 2511 tokens, respectively.

Random Routing matches SG-FKL's EMA teacher ($H = 4$), objective, coefficient, and per-step distillation group count. Groups are selected uniformly at random, with GRPO retained on all groups.

Table 9: **Paired accuracy and response length.** Easy and hard columns use the RL step-100 cohort ($\tau = 0.9$) and its complement; LSD and Random Routing Hard Acc. is derived from overall and easy-set accuracy. LST_{1000} uses the strict step-1000 cohort and shared reference in Table 8.

Method	Avg. Pass@1 (%)	Easy queries		Hard queries		LST_{1000} (%)
		Acc. (%)	Mean tokens	Acc. (%)	Mean tokens	
RL	42.90	98.15	996.9	32.85	2329.3	19.03
LSD (SG-FKL)	44.20	99.10	859	34.22	2511	-3.72
LSD (SG-RKL)	42.80	99.43	757.2	32.50	2395.0	-10.92
LSD (PG-RKL)	43.56	99.15	801.6	33.45	2392.5	1.45
Random Routing	22.08	89.74	895.0	9.78	2006.6	-11.00
Fixed SG-FKL	30.44	94.89	690.6	18.72	1322.8	-10.81

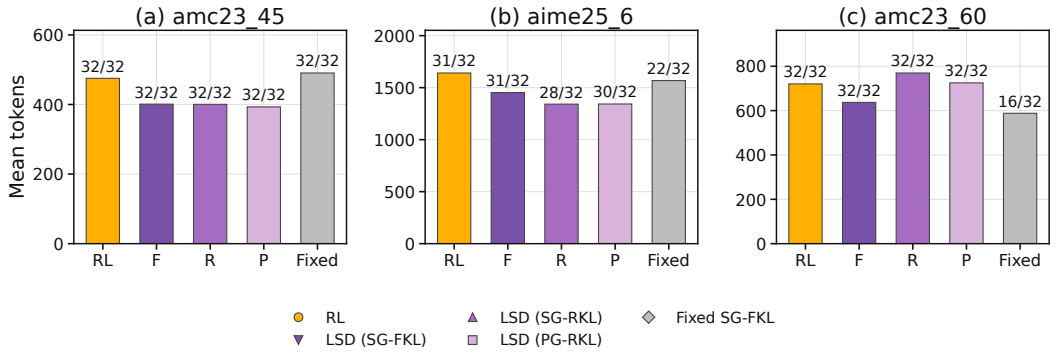


Figure 14: **Query-level examples from the strict step-1000 cohort.** Bar height is mean response length; labels give correct rollouts out of 32. All three queries were answered correctly in all 32 RL anchor rollouts. The examples include unchanged correctness, an LSD regression, and a Fixed regression. F, R, and P denote SG-FKL, SG-RKL, and PG-RKL, respectively.

For `amc23_45`, RL and all three LSD variants answer 32/32 correctly, while mean length falls from 475.2 to 401, 400.4, and 393.2 tokens under SG-FKL, SG-RKL, and PG-RKL, respectively. This is a direct example of shortening at identical observed accuracy. It is selected near the median joint length reduction among queries with 32/32 correctness for RL, SG-RKL, and PG-RKL and shorter responses under both RKL variants.

The other examples expose the limits of the aggregate result. For `aime25_6`, correctness falls from RL's 31/32 to 28/32 under SG-RKL and 30/32 under PG-RKL despite shorter responses; SG-FKL retains 31/32 while reducing mean length from 1640.9 to 1453 tokens. For `amc23_60`, RL and all three LSD variants retain 32/32 correctness; SG-FKL reduces mean length from 720.4 to 637 tokens, while Fixed drops to 16/32 while shortening its response. These examples are selected as the largest respective correctness regressions among shortened queries in the cohort, so that the case analysis includes failure cases as well as successful preservation.

E Related Work

Approaches to efficient reasoning broadly include test-time compute allocation, RL-based optimization, and distillation.

Test-time compute allocation. At inference time, reasoning computation can be adjusted through the length of reasoning traces, the number of sampled solutions, and verification or search. Chain-of-thought prompting and self-consistency illustrate the benefits of explicit reasoning and exploring multiple solution paths (Wei et al., 2022; Wang et al., 2022), while process supervision supports verifier-guided selection (Lightman et al., 2024). However, longer reasoning can waste computation on simple problems (Chen et al., 2024). Snell et al. (2024) show that effective compute allocation depends on prompt difficulty, motivating adaptive inference strategies. Wan et al. (2026c) further formulate cross-query budget allocation using a global shadow price to balance the marginal utility of reasoning computation. Budget forcing provides another way to control the amount of reasoning at test time (Muennighoff et al., 2025). These methods adjust how much computation a model spends when answering a query.

RL-based reasoning efficiency. RL can train policies to use computation more efficiently. L1 optimizes accuracy together with adherence to requested reasoning-length constraints (Aggarwal & Welleck, 2025), and lazy length penalties incorporate response-length reduction into reasoning RL (Yuan et al., 2026). DAST uses difficulty-dependent budgets, reward shaping, and preference optimization to discourage excessive reasoning on easier problems while retaining sufficient computation for harder ones (Shen et al., 2025). AdapThink adapts length penalties to query difficulty (Xu et al., 2026). Training efficiency can also be improved through sample selection: DAPO filters rollout groups with uninformative rewards (Yu et al., 2025), while online difficulty filtering focuses learning on tasks of intermediate difficulty (Bae et al., 2026). Under group-relative objectives, however, all-correct groups have no reward-based relative advantage (Shao et al., 2024). This leaves little direct signal for preserving their existing concise behavior as the policy continues learning from other prompts.

Distillation and LSD. Distillation provides token-level supervision for learning concise reasoning. Generalized Knowledge Distillation trains students on their own generated sequences using teacher feedback, supports different divergence objectives, and can be combined with RL fine-tuning (Agarwal et al., 2024). CRISP uses a periodically refreshed student copy conditioned on a conciseness instruction as its teacher and optimizes reverse KL on all student rollouts (Sang et al., 2026). Contrastive On-Policy Distillation compares teacher probabilities under light- and heavy-reasoning instructions to construct token-level advantages (Ruan et al., 2026).

Unlike these approaches, LSD combines on-policy distillation with online difficulty routing. Its focus is preserving concise behavior on already-solved queries during ongoing RL training, which we evaluate through LST on fixed easy-query sets.

F EXPERIMENTAL CONFIGURATIONS

Table 10 consolidates shared settings and task-specific differences. Table 11 specifies the RL and LSD variants.

Table 10: **Training and inference configurations for RL and LSD.** Values spanning both columns are shared. Agent mini-batches count transformed training rows. A dash indicates an unspecified setting.

Parameter	Single-turn reasoning	Multi-turn agentic tasks
Optimizer	AdamW; $\beta = (0.9, 0.999)$	
Learning rate / weight decay	10^{-6} / 0.01	
Training batch	32 prompts $\times$ 8 rollouts = 256 trajectories	
PPO epochs / advantage	1 / GRPO	
PPO clip $(\epsilon_{\text{low}}, \epsilon_{\text{high}})$	$(0.20, 0.28)$	
Entropy / RL KL coef.	0 / 0	
Sampling / top-k / TP	Enabled / -1 / 1	
Backbone	Qwen3-4B-Base	Qwen3-8B-Base
Training data	DAPO-Math-17K	CutTheBill `train`
Evaluation data	AMC23 / AIME25 / AIME26 (83 / 30 / 30 queries)	BrowseComp-Plus `test` (830 queries)
Agent / tools	Single-turn math agent / none	DeepResearch / retrieval + Refine
Max. prompt / response tokens	4096 / 4096	700 / 20,000
Max. interaction turns	1	48
Stopping	EOS or response-token limit	EOS, response-token limit, or turn limit
PPO mini-batch	64 trajectories	2048 transformed rows
Dynamic micro-batch limit	16,384 tokens/GPU	24,576 tokens/GPU
Dual-clip c	3	10
LR schedule / warmup	Constant / 0	Constant / 10 steps
Loss aggregation	`seq-mean-token-mean`	`token-mean`
Train $(T, \text{top-}p, n)$	$(0.6, 1, 8)$	$(1.0, 1, 8)$
Validation $(T, \text{top-}p, n)$	$(0.6, 1, 32)$	$(1.0, 0.7, 8)$
Inference engine	vLLM	SGLang
Max. model length	8192	20,700
GPU memory utilization	0.8	0.65
Runtime options / max. sequences	– / –	Eager; overlap scheduling off / 128
Validation / saving interval	–	Every 25 steps
Parallelism / model precision	FSDP / BF16	FSDP2 / FP16
EMA accumulation precision	FP32 (EMA teachers only)	–
Hardware	1 node $\times$ 8 GPUs	1 node $\times$ 8 H20 GPUs

Table 11: **RL and LSD configurations.** LSD retains GRPO on the hard route; the gradient row describes only the distillation route. The three online LSD variants share these settings across tasks. Fixed SG-FKL results are reported for single-turn reasoning.

Parameter	RL	Fixed SG-FKL	SG-FKL	SG-RKL	PG-RKL
Teacher	–	Frozen π_0	EMA		
Routing	–	Fixed map	Online rollout-group accuracy		
Routing threshold τ	–	1.0			
EMA half-life H	–	–	4		
Distillation objective	–	Top-32 FKL		Top-32 RKL	Sample-token RKL
Distillation gradient	–	Supervised			Policy gradient
LSD coefficient	–	1.0			
Easy-route RL coefficient	–	0			
PG advantage clip	–	–	–	–	10
Single-turn W&B ID	–	`s71jbnha`	`tuabqb6l`	`z7hhhn5x`	`kvjksgpr`	`178n91ko`