LEARNING FROM ERROR STRUCTURE: SELECTIVE SUPERVISION FOR REASONING POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) and supervised fine-tuning provide complementary signals for reasoning post-training. As recent approaches increasingly combine them, a key question is where additional supervised correction is most valuable. Existing approaches often rely on rollout accuracy, using failure frequency as a proxy for supervision need. Our empirical analysis shows that this reliance on rollout accuracy can mask important differences in failure structure. Even at the same rollout accuracy, errors may either repeatedly collapse onto the same semantic mode or span diverse reasoning strategies. Motivated by this observation, we introduce LESS (**L**earning from **E**rror **S**tructure: **S**elective Supervision), which allocates auxiliary supervision according to recurrent failure structure. LESS groups incorrect rollouts by their solution method and final answer, measures error concentration with semantic entropy, and selectively applies verified reference demonstrations to recurrent-error questions while retaining standard GRPO for all rollouts. Experiments with Qwen2.5-Math-7B show that LESS substantially improves performance relative to existing methods across multiple mathematical, scientific, and broader reasoning benchmarks. These results suggest that effective supervision depends not only on how often a model fails, but also on how it fails.

Anonymous repository: `https://anonymous.4open.science/r/LESS-A8DB/`

1 INTRODUCTION

Large language models (LLMs) (Brown et al., 2020; Ouyang et al., 2022; OpenAI et al., 2023; Dubey et al., 2024; DeepSeek-AI, 2025) have demonstrated remarkable reasoning capabilities on various tasks (Bang et al., 2023; Zhao et al., 2023; Huang & Chang, 2023; Qiao et al., 2023). Recent LLMs typically generate chain-of-thought responses (Wei et al., 2022; Zelikman et al., 2022) and exhibit complex emergent capabilities, such as autonomous planning (Ke et al., 2025) and self-correction (Kumar et al., 2025). Reinforcement learning with verifiable rewards (RLVR) is the primary driver of these emergent capabilities. In domains such as mathematics, science, and code, automatically verifiable outcomes allow models to learn from large amounts of self-generated reasoning experience without requiring dense human supervision (Cobbe et al., 2021; Lightman et al., 2023). Algorithms such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) further exploit relative rewards within groups of sampled solutions to optimize the reasoning policy directly.

Outcome-based reinforcement learning and supervised reasoning provide fundamentally different forms of learning signals. RLVR offers scalable outcome feedback over trajectories sampled from the current policy, while a reference demonstration provides a richer directional signal by explicitly showing a successful reasoning path. Recent approaches therefore increasingly combine reinforcement learning with supervised reasoning demonstrations to provide additional guidance (Ma et al., 2026; Yan et al., 2025; Zhang et al., 2025). This combination further emphasizes the importance of selecting supervision signals that benefit the current policy.

Applying supervised demonstrations indiscriminately is inefficient. Their value depends on what the current policy already knows and, crucially, on what information is missing from its own rollouts. Most existing methods use failure frequency as a proxy for the need for corrective supervision or its potential value. On this view, questions on which the policy fails more often are treated as

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.

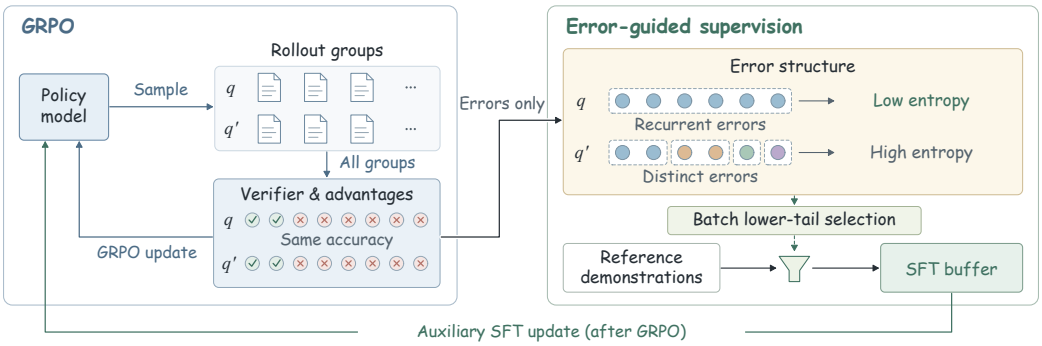


Figure 1: **Overview of LESS.** Standard GRPO is applied to all rollout groups. In parallel, LESS analyzes only incorrect rollouts, identifies questions with recurrent low-entropy error structures, and selectively adds their verified reference demonstrations to an auxiliary SFT update. Error structure determines where additional supervision is allocated without altering the underlying GRPO pathway.

stronger candidates for external guidance. Though intuitive, failure frequency is an incomplete proxy for supervision value. Figure 1 shows an example of two questions for which only two out of eight sampled solutions are correct. For the first question, all six incorrect rollouts repeatedly instantiate the same flawed reasoning strategy and reach the same outcome, while for the second, the six failures span several distinct strategies and answers. Both questions have the same pass rate, but the information contained in their failures is very different. In the first case, additional on-policy failures repeatedly revisit the same unsuccessful direction, while in the second, the policy is already exploring several alternatives.

This distinction is not merely hypothetical. Figure 2a examines questions at fixed rollout accuracy and shows substantial variation in their semantic error-mode configurations. Even among questions with identical outcome statistics, failures can range from highly diverse configurations to strongly concentrated patterns in which many incorrect rollouts belong to the same semantic mode. Summarizing rollouts by failure frequency may induce **structural aliasing**, in which policy states with different internal failure structures are mapped to the same scalar correctness statistic. This structural aliasing matters for supervision allocation because the marginal value of a demonstration depends not only on how often the policy fails, but also on how those failures are organized. When incorrect rollouts repeatedly collapse onto the same semantic mode, additional failed samples contribute progressively less novel information. These samples continue to reinforce the evidence that the same reasoning direction is unsuccessful, without exposing a qualitatively different alternative. The more frequently the policy collapses onto the same failure mode, the less novel information additional failed rollouts provide, and the more valuable a verified alternative trajectory becomes.

A reference demonstration is therefore especially useful in this setting because it supplies a correct reasoning direction that is absent from the policy's recurrent failures. By contrast, when failures span multiple distinct strategies, the policy already exhibits greater semantic exploration, and an additional demonstration may provide less incremental directional information. Therefore, we distinguish two complementary axes of the rollout distribution in this paper:

- **Failure frequency** indicates how often the current policy is unsuccessful.
- **Failure structure** reveals whether it continues to explore different hypotheses or repeatedly becomes stuck in the same wrong direction.

This distinction suggests that supervision should be allocated not only according to how often a policy fails, but also according to whether its failures are recurrent. Motivated by this view, we introduce LESS (**L**earning from **E**rror **S**tructure: **S**elective Supervision), a reasoning post-training framework that allocates corrective supervision according to recurrent failure structure. Specifically, LESS retains standard RLVR for all sampled rollout groups, while using the structure of verifier-rejected rollouts only to decide where additional supervised correction should be introduced. Questions whose failures repeatedly concentrate around the same semantic patterns receive their verified reference demonstrations through an auxiliary SFT update. Conversely, questions with more diverse

failures continue to learn from the standard RL pathway without additional supervision. Thus, the proposed method does not choose between RL and SFT. Instead, it uses the policy's own failure structure to estimate where a verified demonstration is likely to provide the most complementary information.

To operationalize this idea, LESS characterizes the semantic structure of incorrect rollouts. Concretely, for each incorrect rollout, we construct a compact method–answer representation consisting of its core reasoning strategy and final answer. We group semantically equivalent representations into error modes. We then measure how probability mass is distributed across these modes using error-only semantic entropy (Kuhn et al., 2023; Farquhar et al., 2024). Low entropy indicates that failures repeatedly concentrate in a small number of modes, whereas high entropy indicates that the policy produces more diverse errors. Importantly, entropy is not itself an optimization target, but serves only as a diagnostic of error concentration, which LESS uses to allocate auxiliary supervision.

We evaluate LESS at the 7B scale across six mathematical and scientific reasoning benchmarks and three broader reasoning benchmarks. Under the current training protocol, our method achieves macro-averages of 51.0 and 61.4 on the two benchmark groups, outperforming the baselines. Under matched auxiliary-supervision budgets, selecting the concentrated lower-entropy tail also outperforms full-rollout semantic entropy, random selection, and reverse high-entropy selection. These results support failure structure as a useful signal for allocating corrective supervision beyond outcome-level correctness alone.

Our main contributions are fourfold:

- We reformulate adaptive reasoning supervision as a supervision-allocation problem and show that failure frequency is an incomplete proxy for supervision value. Even under fixed rollout accuracy, model generations exhibit substantially different semantic error structures, producing structural aliasing under outcome-only statistics.
- We identify recurrent error structure as a distinct policy pattern in which self-generated failures provide semantically redundant negative evidence, and introduce an error-only semantic-entropy measure to characterize this regime.
- We propose LESS, which retains standard GRPO for all rollout groups while selectively adding verified SFT demonstrations to recurrent-error questions.
- We evaluate LESS across multiple benchmarks, and the results demonstrate that the proposed error-structure-aware supervision substantially improves reasoning post-training across various domains.

2 FAILURE STRUCTURE AS A SUPERVISION-ALLOCATION SIGNAL

The motivation behind LESS is that the value of corrective supervision depends on what information is already contained in the policy's own rollouts. Outcome statistics based on failure frequency describe only how frequently the model succeeds or fails, but do not reveal whether repeated failures provide distinct evidence for learning or repeatedly revisit the same underlying mistake. In this section, we therefore study whether failure structure provides information about the current policy state beyond failure frequency, and whether this distinction is relevant to the allocation of supervised correction.

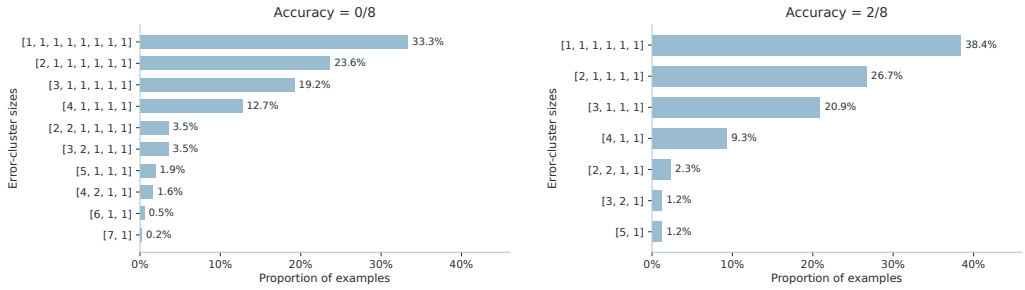
Formally, given a question q, let the current policy generate K reasoning rollouts:

$$Y(q) = \{y_k\}_{k=1}^{K}, \qquad y_k \sim \pi_{\theta_{\text{old}}}(\cdot \mid q), \tag{1}$$

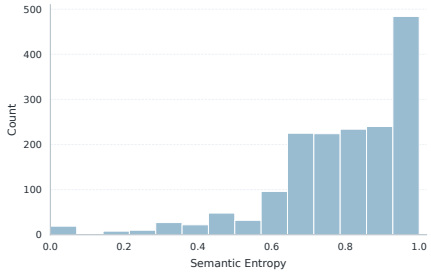
with binary verifier outcomes $r_k \in \{0, 1\}$. The corresponding rollout accuracy is

$$A(q) = \frac{1}{K} \sum_{k=1}^{K} r_k. \tag{2}$$

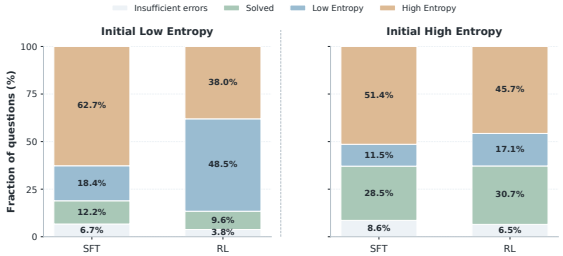
This outcome statistic provides a natural measure of failure frequency by indicating how often the policy succeeds, but it discards relationships among the incorrect trajectories themselves. Under binary verifier feedback, semantically different failures receive the same incorrect outcome label, even when they arise from fundamentally different reasoning strategies. Figure 2a demonstrates



(a) Error-cluster patterns at fixed accuracy.



(b) Error-only semantic entropy.



(c) State transitions after SFT and RL.

Figure 2: **Diagnostic evidence for failure structure beyond failure frequency.** (a) Distribution of semantic error-cluster configurations among questions with fixed rollout accuracy $0/8$ ($n = 432$) and $2/8$ ($n = 86$); (b) Distribution of error-only Method–Answer semantic entropy over analyzable questions; (c) State transitions of initially low- and high-entropy questions after SFT and RL.

this observation directly. Even when rollout accuracy is fixed, incorrect solutions exhibit markedly different error-mode configurations. Some questions produce mostly distinct errors, while others repeatedly collapse onto a dominant mode. Formally, let Z denote the semantic mode of an incorrect rollout. Then,

$$A(q_i) = A(q_j) \quad \not\Rightarrow \quad P(Z|R = 0, q_i) = P(Z|R = 0, q_j). \tag{3}$$

This distinction matters because repeated failures differ in how informative they are. When the policy repeatedly returns to the same semantic failure mode, additional failed rollouts increasingly repeat evidence about the same unsuccessful direction. A verified demonstration can instead introduce a qualitatively different, correct trajectory. Thus, failure structure may provide a useful signal for estimating where directional supervision is particularly valuable.

Figure 2b shows that recurrent failures form a distinct low-entropy tail rather than occurring universally. Figure 2c further shows different training dynamics across error-structure patterns: initially concentrated failures are more often preserved under RL, while SFT more frequently disrupts the dominant recurrent pattern. These observations motivate the use of recurrent failure structure as a signal for supervision allocation.

3 LESS LEARNING FROM ERROR STRUCTURE WITH SELECTIVE SUPERVISION

Motivated by the distinction between failure frequency and failure structure, we propose LESS. As shown in Figure 1, the overall principle of LESS is to retain all rollout groups in the standard GRPO pathway and use verified demonstrations for recurrent-error questions in an auxiliary SFT update. The recurrent errors are identified using semantic entropy (Kuhn et al., 2023; Farquhar et al., 2024). Failure structure therefore determines only where additional supervision is allocated and does not modify the underlying RL objective.

3.1 MEASURING RECURRENT ERROR STRUCTURE

For each question q in the training set $\mathcal{D}$, the current policy samples K rollouts:

$$Y(q) = \{y_k\}_{k=1}^{K}, \qquad y_k \sim \pi_{\theta_{\text{old}}}(\cdot \mid q), \tag{4}$$

and the verifier returns a binary reward r_k. LESS extracts the incorrect rollout subset as

$$Y^-(q) = \{y_k \mid r_k = 0\}, \qquad m(q) = |Y^-(q)|. \tag{5}$$

When $m(q) \geq m_{\min}$, LESS analyzes the incorrect rollouts semantically. For each incorrect rollout y_i, we use a fixed auxiliary model to summarize the core solution strategy, excluding the final answer. We also extract the normalized answer a_i to obtain a compact method–answer representation:

$$z_i = (s_i, a_i), \qquad y_i \in Y^-(q). \tag{6}$$

This representation distinguishes different reasoning failures that may share the same answer while avoiding surface-level variation in full reasoning traces.

We then group valid representations by normalized final answer. Within each answer group, representations are clustered into semantic error modes using the fixed guide, which requires bidirectional semantic equivalence. Each representation joins exactly one matching cluster. Whenever a summary does not match any existing cluster, we create a new one. Representations with different answers remain in separate groups. We denote the resulting semantic error modes by

$$\mathcal{C}^-(q) = \{C_1^-, C_2^-, \ldots, C_L^-\}, \tag{7}$$

where L is the number of error modes, and each cluster represents a recurring combination of extracted answer text and solution method. Appendix A.3 specifies the greedy representative procedure, parsing fallback, and order dependence.

We estimate the empirical distribution over the resulting error modes as

$$p_j^- = \frac{|C_j^-|}{m(q)}, \qquad \sum_{j=1}^{L} p_j^- = 1. \tag{8}$$

For $m(q) \geq m_{\min} = 2$, we measure the concentration of this distribution using normalized error-only semantic entropy (Kuhn et al., 2023; Farquhar et al., 2024):

$$H_{\text{MA}}^-(q) = \frac{-\sum_{j=1}^{L} p_j^- \log p_j^-}{\log m(q)}, \qquad H_{\text{MA}}^-(q) \in [0, 1]. \tag{9}$$

The normalization by $\log m(q)$ allows comparison across questions with different numbers of incorrect rollouts. In the current formulation, $H_{\text{MA}}^-(q) = 0$ when all incorrect rollouts belong to one semantic mode, while $H_{\text{MA}}^-(q) = 1$ when every incorrect rollout forms a distinct mode. Accordingly, lower $H_{\text{MA}}^-(q)$ implies greater error concentration and stronger recurrence.

Answer extraction may fail due to malformed or indeterminate generations. Let $\mathcal{U}(q) \subseteq Y^-(q)$ denote the indeterminate rollouts, and define the risk factor $\eta(q) = |\mathcal{U}(q)|/m(q)$. If $\eta(q) > \rho$, we assign $H_{\text{MA}}^-(q) = 1$ and skip semantic clustering. Otherwise, we cluster the valid representations as above and place all indeterminate rollouts in one additional cluster. The overridden value remains in the batch-relative quantile computation and acts as a maximal-entropy penalty rather than a hard exclusion. We treat ρ as an operational threshold and report its setting in Appendix A.

Importantly, LESS does not optimize $H_{\text{MA}}^-(q)$. The score is used only to estimate how repetitive the observed failures are, and therefore where a reference trajectory may add information not already present in the policy's sampled failures. Algorithm 1 specifies the update sequence and abbreviates $H_{\text{MA}}^-(q_i)$ as H_i.

3.2 SELECTIVE AUXILIARY SUPERVISION

The intuition behind LESS is that recurrent failures make additional failed rollouts increasingly redundant. Therefore, LESS allocates auxiliary supervision to the most recurrent errors in each training batch.

Formally, for a training batch $\mathcal{B}$, we consider questions with enough incorrect rollouts:

$$\mathcal{A}_{\mathcal{B}} = \{q \in \mathcal{B} \mid m(q) \geq m_{\min}\}. \tag{10}$$

We set $m_{\min}$ to 2 by default to exclude the single-error degeneracy. The empirical cluster entropy is necessarily zero and provides no evidence of a recurring error pattern when only one incorrect rollout exists.

Given a supervision-allocation ratio α, let

$$\tau_{\mathcal{B}} = Q_{\alpha}\left(\left\{H^{-}_{\mathrm{MA}}(q) : q \in \mathcal{A}_{\mathcal{B}}\right\}\right), \tag{11}$$

where Q_{α} denotes the α-quantile. LESS selects the lower-entropy tail:

$$\mathcal{S}_{\mathcal{B}} = \left\{q \in \mathcal{A}_{\mathcal{B}} : H^{-}_{\mathrm{MA}}(q) \leq \tau_{\mathcal{B}}\right\}. \tag{12}$$

The inclusive boundary retains every question tied at $\tau_{\mathcal{B}}$. Consequently, the realized auxiliary-admission rate $|\mathcal{S}_{\mathcal{B}}|/|\mathcal{A}_{\mathcal{B}}|$ can exceed the nominal quantile level α and is not obtained by rounding a target count. An overridden question with $H^{-}_{\mathrm{MA}}(q) = 1$ is not selected whenever $\tau_{\mathcal{B}} < 1$; if $\tau_{\mathcal{B}} = 1$, every analyzable question is selected because all entropy values lie in $[0, 1]$.

This batch-relative quantile keeps the selection rule focused on the most concentrated part of the current policy's error distribution as that distribution changes across training stages. The verified reference demonstrations for questions in $\mathcal{S}_{\mathcal{B}}$ are added to a FIFO auxiliary SFT buffer. Importantly, selected questions are not removed from reinforcement learning. All rollout groups continue to participate in GRPO. Let θ denote the trainable parameters. Training then proceeds as follows:

$$\theta \xrightarrow{\text{GRPO on all rollout groups}} \theta' \xrightarrow{\text{SFT on selected low-entropy tail}} \theta''. \tag{13}$$

The SFT update is performed after GRPO whenever the auxiliary buffer contains a complete mini-batch, matching the two-path procedure illustrated in Figure 1. Appendix A specifies the buffer size, update schedule, and implementation details for quantile ties, buffer consumption, and parsing fallbacks.

In summary, LESS uses the policy's own error structure to determine where a verified demonstration can provide information that repeated failed rollouts do not. Outcome feedback remains the primary learning signal for all examples, while recurrent-error questions receive an additional directional correction.

4 EXPERIMENTS

We conduct the main evaluation of LESS at the 7B scale under a shared training and evaluation protocol, and examine additional model scales and families in Section 4.2.

4.1 EXPERIMENTAL SETUP

Data Construction. Following the data construction procedure of ReLIFT (Ma et al., 2026), we use the default 94k-prompt subset of OpenR1-Math-220k (Hugging Face, 2025), whose problems originate from NuminaMath 1.5 (Li et al., 2024) and whose demonstrations are generated by DeepSeek-R1 (DeepSeek-AI, 2025). We discard demonstrations longer than 8192 tokens or rejected by MATH-VERIFY (Hugging Face, 2024), obtaining 46k prompt–demonstration pairs, denoted *OpenR1-Math-46k-8192*. This pool supplies the reference demonstrations for the reproduced methods.

Evaluation. We evaluate mathematical and scientific reasoning on AIME-24, AIME-25, AMC, MATH-500 (Hendrycks et al., 2021), Olympiad (He et al., 2024), and Minerva (Lewkowycz et al., 2022). To assess generalization beyond the math-focused training distribution, we also evaluate on MMLU-Pro (Wang et al., 2024), ARC-c (Clark et al., 2018), and GPQA (Rein et al., 2023). We report avg@32 on AIME-24/25 and AMC, which have small test sets, and pass@1 on the remaining benchmarks. Decoding and answer-scoring details are provided in Appendix A.2. We additionally assess sensitivity to stochastic decoding through repeated sampling on the six mathematical and scientific reasoning benchmarks, Detailed sampling variability statistics are reported in Appendix C.2.

Baselines. We use Qwen2.5-Math-7B (Yang et al., 2024) as the base model and compare LESS with previous RLVR methods—SimpleRL-Zero (Zeng et al., 2025), OpenReasoner-Zero (Hu et al., 2025), PRIME-Zero (Cui et al., 2025), and Oat-Zero (Liu et al., 2025)—and standalone SFT and GRPO (Shao et al., 2024). For methods combining SFT and RL, we include the joint and sequential baselines considered by ReLIFT (Ma et al., 2026): *RL w/ SFT Loss* adds a supervised loss to the RL objective using 46K demonstrations, while *SFT then RL* applies RL after supervised initialization with 46K demonstrations. We also compare with ReLIFT and LUFFY (Yan et al., 2025). In the main 7B comparison, previous RLVR results are taken from LUFFY; SFT, GRPO, ReLIFT, and LUFFY are independently reproduced. The main comparison of GRPO, ReLIFT, LUFFY, and LESS shares the training pool, rollout configuration, and GRPO step budget. Baseline definitions and training details are given in Appendix A; demonstration budgets and compute are summarized in Table 2.

4.2 MAIN RESULTS

SOTA performance on RLVR with Qwen2.5-Math-7B. Table 1 compares LESS with prior RLVR methods and our reproduced RL baseline using Qwen2.5-Math-7B. Across six mathematical and scientific reasoning benchmarks, LESS achieves a macro-average of 51.0, the highest among the compared methods. This exceeds Oat-Zero, the strongest previously reported RLVR baseline in this group, by 7.3 points and reproduced RL by 5.7 points. On Minerva, LESS scores 38.2, while it leads all baselines on MATH-500 and Olympiad.

On the three out-of-distribution benchmarks, LESS achieves a macro-average of 61.4, improving over OpenReasoner-Zero by 9.8 points and reproduced RL by 5.5 points. Thus, its aggregate gains extend beyond the mathematical training distribution.

Table 1: Main results on Qwen2.5-Math-7B. AIME-24/25 scores are separated by a slash. Avg. denotes the unweighted macro-average over the corresponding benchmark group. Previous RLVR results follow the values compiled by LUFFY (Yan et al., 2025). Best results are bolded and second-best results are underlined, including ties.

Model	In-Distribution Performance						Out-of-Distribution Performance			
	AIME 24/25	AMC	MATH-500	Minerva	Olympiad	Avg.	ARC-c	GPQA	MMLU-Pro	Avg.
Qwen2.5-Math-7B	18.9/6.4	47.1	66.6	23.9	31.7	32.4	10.2	16.8	33.0	20.0
Qwen-Instruct (Yang et al., 2024)	12.5/10.2	48.5	80.4	32.7	41.0	37.6	70.3	24.7	34.1	43.0
Previous RLVR Methods										
SimpleRL-Zero	27.0/6.8	54.9	76.0	25.0	34.7	37.4	30.2	23.2	34.5	29.3
OpenReasoner-Zero	16.5/15.0	52.1	82.4	33.1	47.1	41.0	66.2	29.8	**58.7**	51.6
PRIME-Zero	17.0/12.8	54.0	81.4	39.0	40.3	40.7	73.3	18.2	32.7	41.4
Oat-Zero	**33.4**/11.9	61.2	78.0	34.6	43.4	43.7	70.1	23.7	41.7	45.2
Replication of SFT and RL										
SFT	22.2/22.3	52.8	82.6	**40.8**	43.7	44.1	75.2	24.7	42.7	47.5
RL	25.8/14.4	60.1	83.2	39.4	48.7	45.3	83.8	35.4	48.6	55.9
Methods Combining SFT and RL										
RL w/ SFT Loss	27.0/22.0	61.0	85.0	34.0	54.0	47.2	66.0	22.0	44.0	44.0
SFT then RL	27.3/22.4	63.9	87.6	39.0	56.1	49.4	69.6	24.2	33.6	42.5
LUFFY	28.1/21.5	**66.2**	88.4	38.1	54.8	49.5	81.7	38.8	52.6	57.7
ReLIFT	27.5/21.2	64.9	86.4	40.1	57.0	49.5	84.4	38.1	53.7	58.7
LESS	32.7/**23.0**	65.5	**88.6**	38.2	**58.1**	**51.0**	**85.8**	**43.4**	55.1	**61.4**

Comparing against methods combining SFT and RL. We next compare alternative ways of incorporating demonstrations into RL. LESS exceeds joint training (RL w/ SFT Loss) by 3.8 and 17.4 points and sequential training (SFT then RL) by 1.6 and 18.9 points on the in-distribution and out-of-distribution macro-averages, respectively. Relative to ReLIFT and LUFFY, the corresponding gains are 1.5/2.7 and 1.5/3.7 points. On AIME-24 and GPQA, LESS exceeds LUFFY by 4.6 points on each benchmark, although it trails LUFFY by 0.7 points on AMC. Additionally, Table 2 shows that LESS uses fewer demonstrations than the SFT-based baselines, with compute comparable to SFT then RL.

Extending LESS to More Models. Table 3 extends the comparison to a smaller model, Qwen2.5-Math-1.5B, and two 8B models, Qwen3-8B and LLaMA-3.1-8B-Instruct. On the six mathematical

and scientific benchmarks, LESS achieves a macro-average of 36.6 at the 1.5B scale, exceeding the reported SFT and RL results by 4.7 and 6.6 points. The initialization and SFT results for the 1.5B and LLaMA models are reported as in LUFFY (Yan et al., 2025); we independently train the on-policy RL baselines. On Qwen3-8B, LESS achieves a macro-average of 53.1, exceeding SFT and RL by 4.6 and 4.1 points, respectively. On LLaMA-3.1-8B-Instruct, LESS achieves a macro-average of 13.3.

4.3 ABLATION STUDY AND ANALYSIS

We conduct an ablation study to examine how selecting questions for auxiliary supervision affects LESS. We compare four variants: *Full-rollout SE (Full SE)*, which computes semantic entropy over both correct and incorrect rollouts; *Random*, which first identifies questions with zero rollout accuracy in the current batch, then uniformly samples 20% of these questions for auxiliary supervision; *Reverse*, which selects the complement of the low-entropy set; and *Lowest-accuracy selection (Low Acc.)*, which replaces error entropy with rollout accuracy under the same inclusive quantile rule as LESS. All variants use the same training and evaluation configuration and a matched auxiliary-SFT budget, with every rollout group retained for GRPO. Figure 3 shows that LESS outperforms all four variants, with the largest drop under reverse selection.

Table 2: Demonstration and compute budgets.

Method	Demonstrations	GPU-hours
SFT	46K	8×10
RL	0	8×42
RL w/ SFT Loss	46K	8×120
SFT then RL	46K	8×57
LESS	10K	8×62

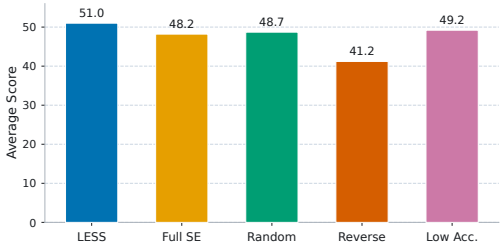


Figure 3: Component ablations: macro-average over six reasoning benchmarks.

Selection by rollout accuracy is the strongest alternative but remains below LESS on both benchmark groups, supporting error structure as a selection signal beyond failure frequency. Detailed results are in Appendix C.

Table 3: Results on six mathematical and scientific reasoning benchmarks across additional models. Avg. is the unweighted macro-average. Best results within each model group are bolded and second-best results are underlined, including ties.

Model	AIME 24	AIME 25	AMC	MATH-500	Minerva	Olympiad	Avg.
LLaMA-3.1-8B-Instruct							
Instruct	**5.1**	0.4	**18.6**	**44.6**	**19.5**	**14.1**	**17.1**
SFT	0.5	0.1	5.4	20.2	4.0	5.3	5.9
RL	0.3	**0.5**	9.4	23.4	17.6	6.1	9.6
LESS	1.7	0.2	12.9	39.2	14.8	11.0	13.3
Qwen3-8B							
Base	10.1	8.9	37.9	65.6	33.8	33.6	31.7
Instruct	22.0	19.3	66.5	82.8	37.9	52.3	46.8
SFT	25.4	**25.7**	61.3	86.4	46.0	46.0	48.5
RL	22.5	21.4	**71.8**	84.6	43.8	50.1	49.0
LESS	**32.7**	22.2	65.9	**91.2**	**47.1**	**59.4**	**53.1**
Qwen2.5-Math-1.5B							
Base	7.2	3.6	26.4	28.0	9.6	21.2	16.0
Instruct	12.1	8.9	**48.1**	77.4	28.7	39.1	35.7
SFT	11.7	**13.2**	37.8	70.6	26.8	31.3	31.9
RL	11.8	7.7	40.2	61.8	26.8	32.0	30.0
LESS	**14.8**	10.5	45.5	**77.8**	**29.8**	**41.3**	**36.6**

We also vary the lower-tail selection quantile α, retaining boundary ties. Table 4 shows that more selective supervision yields the highest macro-average in both benchmark groups among the tested settings, while admitting higher-entropy questions reduces performance.

Table 4: Effect of the selection quantile α on all nine benchmarks. AIME-24/25 scores are separated by a slash. Avg. denotes the unweighted macro-average over the corresponding benchmark group. Higher is better.

α	In-Distribution Performance						Out-of-Distribution Performance			
	AIME 24/25	AMC	MATH-500	Minerva	Olympiad	Avg.	ARC-c	GPQA	MMLU-Pro	Avg.
0.20	**32.7/23.0**	**65.5**	**88.6**	38.2	**58.1**	**51.0**	**85.8**	**43.4**	**55.1**	**61.4**
0.50	19.5/17.3	52.3	82.0	32.0	48.4	41.9	77.0	31.6	49.9	52.8
0.80	19.9/18.4	52.6	80.8	35.7	46.7	42.4	77.7	37.8	46.0	53.8

5 Related Work

Supervised guidance for reasoning. Hybrid training incorporates supervised guidance to support reasoning beyond what a policy readily discovers through sampling. LUFFY combines expert demonstrations with on-policy rollouts (Yan et al., 2025); ReLIFT interleaves RL with supervised updates on difficult questions (Ma et al., 2026); and BREAD branches rollouts from partial expert prefixes (Zhang et al., 2025). Recent approaches exploit failed attempts directly: CIPO augments RL with correction rollouts conditioned on the model's own errors (Ren et al., 2026), while MOPD conditions a distillation teacher on successful and failed peer rollouts from the same question (Yu et al., 2026). At the token level, Liu et al. (2026) show that supervising one or two tokens per trajectory can match dense on-policy distillation on reasoning benchmarks. Together, these studies motivate examining both the source of guidance and where supervision is applied.

Semantic uncertainty and recurrent errors. Semantic entropy measures uncertainty over meaning-equivalent generations (Kuhn et al., 2023; Farquhar et al., 2024). SEED-GRPO uses this signal to attenuate updates on uncertain questions (Chen et al., 2025), and GCPO combines geometric semantic disagreement with reward-based calibration to regulate policy updates (Zhang et al., 2026). Agreement alone, however, does not imply correctness: Li et al. (2026) show that repeated incorrect answers can evade semantic-entropy-based hallucination detection and use diversity-oriented fine-tuning to improve their detectability. LESS uses this distinction to address question-level supervision allocation. It computes entropy over method–answer clusters of verifier-rejected rollouts and selects reference demonstrations for questions with concentrated error patterns. This selection distinguishes questions with identical rollout accuracy but different failure structures, directing auxiliary SFT toward recurrent errors while retaining every complete rollout group in the GRPO update.

6 Conclusion

Outcome-level correctness describes how often a reasoning policy fails but omits how its incorrect rollouts are organized. LESS captures this organization with semantic entropy over verified incorrect Method–Answer representations and uses the signal to route auxiliary supervised correction. The diagnostic and experimental results support error structure as a training signal complementary to outcome-level correctness.

AI use statement

Generative AI tools assisted with drafting and polishing the Experiments section, as well as language editing, paper organization, literature discovery, citation and reference formatting, and LaTeX formatting. AI tools were not used for research ideation, problem formulation, or method design. The authors manually verified all references, technical claims, numerical results, interpretations, and scientific conclusions and are responsible for the reported content.

As part of executing the proposed method, we call a fixed external LLM guide, identified as gpt-5.4-mini, to compress incorrect rollouts into method summaries and judge semantic equivalence between compressed candidates. We chose this model because its relatively low cost and fast response time support the repeated calls required by the pipeline, while its text-processing capability is sufficient for summarization and semantic-equivalence judgments. These outputs construct the

routing signal; verifier outputs supply rewards, and reference demonstrations come from the training data. Appendix B reports the exact guide prompts.

REFERENCES

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. SEED-GRPO: Semantic Entropy Enhanced GRPO for Uncertainty-Aware Policy Optimization. arXiv preprint arXiv:2505.12346, 2025. URL https://arxiv.org/abs/2505.12346.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. arXiv preprint arXiv:1803.05457, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. arXiv preprint arXiv:2502.01456, 2025.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. doi: 10.1038/s41586-024-07421-0.

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. arXiv preprint arXiv:2402.14008, 2024.

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. arXiv preprint arXiv:2103.03874, 2021.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-Reasoner-Zero: An open source approach to scaling up reinforcement learning on the base model. arXiv preprint arXiv:2503.24290, 2025.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1049–1065, 2023.

Hugging Face. Math-Verify. GitHub repository, 2024. URL https://github.com/huggingface/Math-Verify.

Hugging Face. Open-R1: A fully open reproduction of DeepSeek-R1. GitHub repository, 2025. URL https://github.com/huggingface/open-r1.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, et al. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*, 2025.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. arXiv preprint arXiv:2302.09664, 2023.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 54523–54549, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. arXiv preprint arXiv:2309.06180, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. arXiv preprint arXiv:2206.14858, 2022.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Q. Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath 1.5: A dataset of competition mathematics problems and solutions. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-1.5.

Qiuyuan Li, Hongliang Dai, and Piji Li. Diversity-Oriented Fine-Tuning for Uncertainty-Based Hallucination Detection. arXiv preprint arXiv:2607.16643, 2026. URL https://arxiv.org/abs/2607.16643.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. arXiv preprint arXiv:2305.20050, 2023.

Zhishuai Liu, Xingzi Xu, Mehmet Saygin Seyfioglu, Pan Xu, and Karim Bouyarmane. Extremely Sparse Supervision Incentivizes Reasoning Ability. arXiv preprint arXiv:2609.04565, 2026. URL https://arxiv.org/abs/2609.04565.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. arXiv preprint arXiv:2503.20783, 2025.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, Bin Cui, and Wentao Zhang. Learning What Reinforcement Learning Can't: Interleaved Online Fine-Tuning for Hardest Questions. In *International Conference on Learning Representations*, 2026. doi: 10.48550/ARXIV.2506.07527. URL https://arxiv.org/abs/2506.07527.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, and et al. Gpt-4 technical report, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155, 2022.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5368–5393, 2023.

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. arXiv preprint arXiv:2311.12022, 2023.

Mengjie Ren, Jie Lou, Boxi Cao, Xueru Wen, Hongyu Lin, Xianpei Han, Le Sun, Xing Yu, and Yaojie Lu. Learning from Failures: Correction-Oriented Policy Optimization with Verifiable Rewards. arXiv preprint arXiv:2605.14539, 2026. URL https://arxiv.org/abs/2605.14539.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. arXiv preprint arXiv:2406.01574, 2024.

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. arXiv preprint arXiv:2201.11903, 2022.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=vO8LLoNWWk.

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. arXiv preprint arXiv:2409.12122, 2024.

Weichen Yu, Xiaomin Li, Yizhou Zhao, Xiaoze Liu, Ruowang Zhang, Haixin Wang, Yinyi Luo, Chen Henry Wu, Gaurav Mittal, Matt Fredrikson, and Yu Hu. Multi-Rollout On-Policy Distillation via Peer Successes and Failures. arXiv preprint arXiv:2605.12652, 2026. URL https://arxiv.org/abs/2605.12652.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. arXiv preprint arXiv:2203.14465, 2022.

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. arXiv preprint arXiv:2503.18892, 2025.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. BREAD: Branched rollouts from expert anchors bridge SFT and RL for reasoning. arXiv preprint arXiv:2506.17211, 2025.

Zheyuan Zhang, Kaiwen Shi, Han Bao, Zehong Wang, Tianyi Ma, and Yanfang Ye. Why Semantic Entropy Fails: Geometry-Aware and Calibrated Uncertainty for Policy Optimization. arXiv preprint arXiv:2605.21801, 2026. URL https://arxiv.org/abs/2605.21801.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

A Main Experiment Details

This section specifies the training, routing, evaluation, and resource-accounting settings used for the main results.

A.1 Experiment Details

RL. We train the standalone RL baseline with GRPO on the same training prompts as LESS, without auxiliary supervised updates. The verifier assigns a binary reward of 1 to correct answers and 0 otherwise. We omit the KL penalty, length normalization, and standard-error normalization from the GRPO objective. The rollout batch size is 128, with eight rollouts per prompt sampled at temperature 1.0 for exploration. The update batch size is 64 and the learning rate is 1×10^{-6}. In the main 7B comparison, RL, ReLIFT, LUFFY, and LESS share the rollout configuration and each use 600 RL steps on 8 A800 GPUs.

SFT. We train the standalone SFT baseline for three epochs with batch size 64 and learning rate 5×10^{-5}. The model is trained only with the supervised objective on the fixed demonstration pairs, without RL updates or online question selection.

RL w/ SFT Loss. We compute the on-policy loss on 7 on-policy samples and the SFT loss on 1 off-policy sample per prompt. The remaining setup is the same as in the other RL experiments.

SFT then RL. This baseline separates supervised learning and reinforcement learning into two successive stages. We first use the full 46K demonstration pool for supervised initialization, then continue RL training from the resulting checkpoint. The demonstrations are used only in the initialization stage, and no further supervised updates are performed during RL. This baseline tests whether providing demonstration-based knowledge before RL is sufficient to obtain the benefits of combining the two training objectives.

LESS. We use the RL configuration described above and interleave auxiliary SFT updates. The learning rate is 1×10^{-6} for both GRPO and auxiliary SFT.

LLaMA-3.1-8B-Instruct. We train LLaMA-3.1-8B-Instruct on an 11K subset of OpenR1-Math-220k, filtered to include samples with correct DeepSeek-R1 solutions shorter than 2,048 tokens. We adopt a simplified prompt without explicit <think> and </think> tokens. The SFT learning rate is set to 2×10^{-5}, and the maximum response length is 2,048 tokens for both RL training and evaluation.

A.2 Evaluation and Reproducibility

Evaluation. We conduct our evaluations with vLLM (Kwon et al., 2023), using temperature 0.6 and a maximum generation length of 8192 tokens, except for LLaMA-3.1-8B-Instruct, which uses the 2,048-token limit described above. Because AIME-24, AIME-25, and AMC have relatively small test sets, we sample 32 responses per problem and report avg@32. On MATH-500, Olympiad, Minerva, MMLU-Pro, ARC-c, and GPQA, we sample one response per problem and report pass@1 accuracy. We extract and verify mathematical answers with MATH-VERIFY (Hugging Face, 2024). For the three multiple-choice benchmarks, we randomly permute the answer choices before inference and remap the gold option label accordingly. The evaluator extracts the final option label and scores it by exact match against the remapped label; responses without a valid option are incorrect. For RL w/ SFT Loss and SFT then RL, the GPQA entries in Table 1 correspond to GPQA-Diamond. For the baselines, we independently reproduce and verify the results of all methods combining SFT and RL. The results for previous RLVR methods are reported as in the LUFFY paper (Yan et al., 2025). The initialization and SFT results for Qwen2.5-Math-1.5B and LLaMA-3.1-8B-Instruct in Table 3 are also reported as in LUFFY. We independently train the on-policy RL baselines, labeled RL in our tables.

Implementation and reproducibility. Training and inference use the rollout-generation template in Appendix B. The auxiliary SFT buffer resides only in memory and is not stored in training check-

points. Resuming from a checkpoint therefore reinitializes the buffer and discards any pending group of fewer than 32 demonstrations.

Algorithm 1 LESS: GRPO with Error-Only Method–Answer Semantic-Entropy-Guided Auxiliary SFT

Require: Dataset $\mathcal{D}$, policy π_θ, rollout count K, verifier $\mathcal{R}$, fixed guide $\mathcal{G} = (\mathcal{G}_{\text{sum}}, \mathcal{G}_{\text{eq}})$, lower-tail selection quantile level α, minimum error count $m_{\min}$, indeterminate threshold ρ, steps T, SFT buffer batch size B_{SFT}
Ensure: Updated policy π_θ

Initialize FIFO auxiliary SFT queue $\mathcal{M} \leftarrow \varnothing$
for $t = 1, \ldots, T$ **do**
  Sample a question–reference-demonstration batch $\mathcal{D}_b$ from $\mathcal{D}$
  $\mathcal{D}_{\text{RL}} \leftarrow \varnothing$; $\mathcal{A}_b \leftarrow \varnothing$
  for all $(q_i, d_i) \in \mathcal{D}_b$ **do**
    $\mathcal{Y}_i \leftarrow \{y_{i,k}\}_{k=1}^{K} \sim \pi_\theta(\cdot \mid q_i)$
    $\{r_{i,k}\}_{k=1}^{K} \leftarrow \mathcal{R}(q_i, \mathcal{Y}_i, d_i)$
    Add $(q_i, \{(y_{i,k}, r_{i,k})\}_{k=1}^{K})$ to $\mathcal{D}_{\text{RL}}$
    $\mathcal{Y}_i^- \leftarrow \{y_{i,k} \mid r_{i,k} = 0\}$
    if $|\mathcal{Y}_i^-| \geq m_{\min}$ **then**
      $H_i \leftarrow \textsc{ErrorEntropy}(q_i, \mathcal{Y}_i^-, \mathcal{G}, \rho)$
      Add q_i to $\mathcal{A}_b$
    end if
  end for
  if $\mathcal{A}_b \neq \varnothing$ **then**
    $\tau \leftarrow \textsc{Quantile}(\{H_i \mid q_i \in \mathcal{A}_b\}, \alpha)$
    $\mathcal{S} \leftarrow \{q_i \in \mathcal{A}_b \mid H_i \leq \tau\}$
    Enqueue one (q_i, d_i) for each $q_i \in \mathcal{S}$, in batch order
  end if
  Update π_θ with GRPO on $\mathcal{D}_{\text{RL}}$
  if $|\mathcal{M}| \geq B_{\text{SFT}}$ **then**
    $\mathcal{E}_{\text{SFT}} \leftarrow \textsc{PopFront}(\mathcal{M}, B_{\text{SFT}})$
    Update π_θ with SFT on $\mathcal{E}_{\text{SFT}}$
  end if
end for
return π_θ

procedure $\textsc{ErrorEntropy}(q, \mathcal{Y}^-, \mathcal{G}, \rho)$
  $(\mathcal{V}, \mathcal{U}) \leftarrow \textsc{ExtractAndSummarize}(q, \mathcal{Y}^-, \mathcal{G}_{\text{sum}})$
  if $|\mathcal{U}|/|\mathcal{Y}^-| > \rho$ **then**
    return 1
  end if
  $\mathcal{C} \leftarrow \textsc{GreedyRepresentativeCluster}(q, \mathcal{V}, \mathcal{G}_{\text{eq}})$
  if $\mathcal{U} \neq \varnothing$ **then**
    Add $\mathcal{U}$ to $\mathcal{C}$ as one cluster
  end if
  $p_j \leftarrow |C_j|/|\mathcal{Y}^-|$ for each $C_j \in \mathcal{C}$
  return $-\sum_j p_j \log p_j / \log |\mathcal{Y}^-|$
end procedure

A.3 CLUSTERING AND PARSING IMPLEMENTATION DETAILS

For valid extracted answers, the implementation first partitions Method–Answer representations by exact equality of the normalized answer text. Within each answer group, it processes representations in rollout order using greedy representative clustering. Each new method summary is compared with the first member of every existing cluster in cluster-creation order and joins the first matching cluster; if none matches, it starts a new cluster. A guide judgment is positive only when entail_ab, entail_ba, and equivalent are all true. If the guide call or JSON parsing fails, the enabled fallback_to_heuristic

treats two representations as equivalent only when their normalized method-summary strings are identical. The procedure does not construct an all-pairs graph or apply transitive closure, so the final partition can vary with rollout order. For indeterminate answers, the implementation follows the $\rho = 0.5$ override and additional-cluster rules described above.

B PROMPT TEMPLATES

This section reports the exact prompt templates used to generate policy rollouts and construct the semantic-entropy routing signal.

B.1 TRAINING SYSTEM PROMPT

The following template formats policy rollouts during both training and inference.

Training rollout prompt

System:
Your task is to follow a systematic, thorough reasoning process before providing the final solution. This involves analyzing, summarizing, exploring, reassessing, and refining your thought process through multiple iterations.
Structure your response into two sections: **Thought** and **Solution**. In the **Thought** section, present your reasoning using the format: "<think> {thoughts} </think>". Each thought should include detailed analysis, brainstorming, verification, and refinement of ideas. After "</think>," in the **Solution** section, provide the final, logical, and accurate answer, clearly derived from the exploration in the **Thought** section.
If applicable, include the answer in \boxed{} for closed–form results like multiple choices or mathematical solutions.

User:
This is the problem:
{problem}

Assistant:
<think>

B.2 METHOD-SUMMARY COMPRESSOR PROMPT

The external LLM guide uses the following prompt to compress incorrect rollouts into short method summaries for semantic-entropy clustering.

Method-summary compressor prompt

Developer:
You compress mathematical chain–of–thought solutions into short, stable method summaries for semantic clustering.
Summaries must preserve only the main solving strategy, must not judge correctness, and must not repeat the final answer.

User:
Question:
{question}

For each rollout below, produce one short method summary of at most 18 words. Do not include the final answer in the method summary.

Method-summary compressor prompt (continued)

Rollouts:
{rendered_rollouts_json}

B.3 Semantic-Equivalence Prompt

The external LLM guide uses the following prompt to determine whether two compressed reasoning patterns belong to the same semantic cluster.

Semantic-equivalence prompt

Developer:
You judge whether two compressed mathematical reasoning patterns are semantically equivalent for clustering. Judge under the question context. Be conservative. Two candidates are equivalent only if they express the same core solving method and the same final answer.

User:
Question:
{question}

Candidate A:
Answer: {rollout_a.answer_text}
Method: {rollout_a.method_summary}

Candidate B:
Answer: {rollout_b.answer_text}
Method: {rollout_b.method_summary}

Return whether A entails B, whether B entails A, and whether they are equivalent.

```
{
 "entail_ab": true,
 "entail_ba": true,
 "equivalent": true
}
```

The clustering procedure assigns two candidates to the same semantic cluster only when entail_ab, entail_ba, and equivalent are all true.

C Detailed Experiment Results

This section collects detailed result tables, ablation designs, and an analysis of evaluation sampling variability.

C.1 Component Ablations

Table 5 reports the benchmark-level results under a matched auxiliary-SFT budget. All variants retain every rollout group for GRPO and use the same reference-demonstration pool and SFT update configuration.

Full-rollout SE (Full SE). This variant computes semantic entropy over all eight rollouts, including correct responses, rather than restricting the estimate to incorrect rollouts. It retains the Method–Answer representation and selects questions with low entropy for auxiliary supervision.

Random selection among zero-accuracy questions. This variant first filters the current batch to questions with zero rollout accuracy: all eight sampled solutions receive a verifier reward of zero. Denoting this candidate set by $\mathcal{E}_{\mathcal{B}}$, it then uniformly samples $\lceil 0.20|\mathcal{E}_{\mathcal{B}}|\rceil$ distinct questions without replacement. Each selected question contributes its reference demonstration to the auxiliary SFT buffer. Thus, correctness determines eligibility, while selection within the eligible pool is random and uses no semantic entropy. The pool can overlap with the questions selected by LESS. This differs from Low Acc., which applies an inclusive accuracy-quantile cutoff rather than randomly subsampling the zero-accuracy pool.

Reverse selection. This variant uses the same error-only entropy and candidate eligibility as LESS, but selects the complement of its low-entropy set: $H^{-}_{\mathrm{MA}}(q) > \tau_{\mathcal{B}}$, where $\tau_{\mathcal{B}}$ is the 20th-percentile threshold. Questions tied at the threshold are excluded.

Lowest-accuracy selection (Low Acc.). This variant replaces error entropy with rollout accuracy, measured as the fraction of the eight sampled responses judged correct by the verifier. Among questions with at least two incorrect rollouts, it selects those at or below the current batch's 20th-percentile accuracy, retaining all boundary ties. Accuracies and the threshold are recomputed each batch; no method summaries, semantic clustering, or entropy enter selection.

Table 5: Component-ablation results. Higher is better. The two Avg. columns are the unweighted math/science and broader-reasoning macro-averages, respectively.

Variant	A24	A25	AMC	M500	Oly.	Min.	Avg.	MMLU-Pro	ARC-c	GPQA	Avg.
LESS	**32.7**	**23.0**	**65.5**	**88.6**	**58.1**	38.2	**51.0**	**55.1**	**85.8**	**43.4**	**61.4**
Full SE	29.8	19.3	63.6	86.8	51.7	37.9	48.2	50.9	82.3	38.3	57.2
Random	27.3	19.1	66.7	86.4	53.0	39.7	48.7	52.3	84.1	39.1	58.5
Reverse	19.6	17.4	50.2	79.8	48.1	32.0	41.2	45.8	74.6	28.4	49.6
Low Acc.	28.0	21.5	65.0	86.8	56.5	37.5	49.2	53.8	84.5	38.7	59.0

C.2 EVALUATION SAMPLING VARIABILITY

Stochastic decoding can change benchmark scores even when model weights are fixed. We examine this evaluation variability for a fixed LESS checkpoint based on Qwen2.5-Math-7B using repeated responses on all six mathematical and scientific reasoning benchmarks. Within each benchmark, we hold the questions, decoding configuration, and scoring function fixed. Minerva uses the updated scoring function throughout this analysis.

We assemble each evaluation round by taking the correspondingly indexed response from every question and computing the benchmark score. Round variance and SD are the sample variance and standard deviation across these rounds. We estimate the standard error of the mean over all responses separately from unbiased within-question sample variances, assuming independent responses within and across questions. The round statistics describe observed score dispersion, while standard error measures the precision of the repeated-sampling mean. The rounds are assembled from recorded responses within an inference run; they are not independently launched evaluations or training seeds.

Table 6: Sampling variability of LESS on six reasoning benchmarks. Round variance and SD summarize assembled evaluation rounds; standard error is estimated separately from within-question response variation under independent sampling. Variance is in squared percentage points (pp^2); SD and standard error are in percentage points (pp).

Benchmark	Round variance (pp^2)	Round SD (pp)	Std. error (pp)
MATH-500	0.571	0.756	0.303
OlympiadBench	0.765	0.874	0.341
Minerva	3.089	1.758	0.533
AIME-24	16.443	4.055	0.910
AMC	9.968	3.157	0.550
AIME-25	14.158	3.763	0.751

Table 6 shows that AIME-24, AIME-25, and AMC exhibit greater round-to-round variation than MATH-500 and OlympiadBench, with Minerva between these groups. Both AIME benchmarks retain greater uncertainty in their mean scores despite using more responses per question. A larger sampling budget therefore does not by itself guarantee comparable evaluation precision across benchmarks: precision also depends on the number of questions and their response variability. This pattern supports allocating more samples to benchmarks with greater score variability and interpreting close score differences in light of sampling uncertainty. These statistics concern generation randomness for a fixed checkpoint and evaluation protocol; they do not establish stability across training seeds or account for different test questions or systematic scoring errors.

C.3 ACCURACY BY INITIAL ROLLOUT SOLVABILITY

We examine how the gains from LESS vary with initial rollout solvability. We group 1,175 questions by the number of correct responses among eight rollouts from the initial policy: $0/8$, $1/8$–$2/8$, and $3/8$–$8/8$, containing 301, 174, and 700 questions, respectively. Figure 4 compares post-training accuracy on these fixed groups.

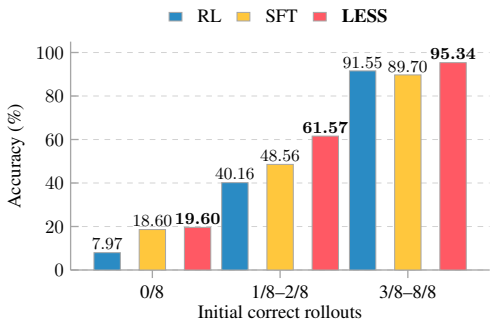


Figure 4: Post-training accuracy by initial rollout solvability.

LESS achieves the highest accuracy in all three groups, with its largest gains in the $1/8$–$2/8$ group: 61.57%, exceeding RL and SFT by 21.41 and 13.01 percentage points, respectively. On questions with no correct initial rollout, LESS reaches 19.60%, compared with 7.97% for RL and 18.60% for SFT. In the $3/8$–$8/8$ group, it reaches 95.34%, improving over RL and SFT by 3.79 and 5.64 points. Thus, the largest accuracy gains occur on questions for which the initial policy can occasionally generate a correct solution, while improvements also extend to both lower- and higher-solvability groups.

D CASE STUDY

This example, drawn from OlympiadBench (He et al., 2024), contrasts three reasoning traces for the same problem. LESS returns $d = 3$ in 1,147 tokens; RL returns $d = 4$ after miscounting the remaining terms; and SFT self-corrects to $d = 3$ in 4,853 tokens.

Question: OlympiadBench Problem 125

In an increasing sequence of numbers with an odd number of terms, the difference between any two consecutive terms is a constant d, and the middle term is 302. When the last 4 terms are removed from the sequence, the middle term of the resulting sequence is 296. What is the value of d?

LESS (Correctness: True; Length: 1,147 tokens)

$\cdots$ Let the original sequence contain $2n+1$ terms. Its middle term is the $(n+1)$-th term, so $a+nd = 302$. After removing the last four terms, the sequence contains $2n-3$ terms, whose middle term is the $(n-1)$-th original term. Therefore, $a+(n-2)d = 296$. Subtracting gives $2d = 302 - 296 = 6$, and hence $\boxed{d = 3}$.

RL (Correctness: False; Length: 1,219 tokens)

$\cdots$ When the last four terms are removed, the new sequence has $n-4$ terms, which is even. Taking the average of its two middle terms gives $a_1 + \left(\frac{n-4}{2}\right) d = 296$. Together with $a_1 + \left(\frac{n-1}{2}\right) d = 302$, we obtain $\frac{3}{2}d = 6$, so $\boxed{d = 4}$.

SFT (Correctness: True; Length: 4,853 tokens)

$\cdots$ The original middle term and the new middle term satisfy $a_1 + kd = 302$ and $a_1 + (k-2)d = 296$. Subtracting gives $2d = 6 \implies d = 3$.
Wait, is that possible? Let me verify with an example. Suppose $k = 5$. Then the sequence is $287, 290, 293, 296, 299, 302, 305, 308, 311, 314, 317$.
If we remove the last four terms $(314, 317)$, the new middle term is 299, not 296. Hmm, that is a discrepancy. $\cdots$
Let me derive the equations again. The original middle term is the $(k+1)$-th term, while the new middle term is the $(k-1)$-th term. Thus, $2d = 302 - 296 = 6$. Therefore, $\boxed{d = 3}$.