RETHINKING RL FOR LLM REASONING: IT'S SPARSE POLICY SELECTION, NOT CAPABILITY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning has become the standard for improving reasoning in large language models, yet evidence increasingly suggests that RL does not teach new strategies; it redistributes probability mass over solutions the base model already contains. In this work we ask: if RL merely steers the model toward paths it already knows, is the RL optimization loop itself necessary? Through token-level analysis across multiple model families and RL algorithms, we find that RL's beneficial footprint is a sparse, predictable correction concentrated at high-entropy *decision points* where the model is uncertain which branch to take. Only 1–4% of token positions are affected, the promoted token almost always lies within the base model's top-5 alternatives, and targeted corrections at those few positions causally recover a large fraction of RL's accuracy gain, while random corrections fail. The base model's own entropy identifies these positions without any RL-trained model, and the entire correction is low-dimensional, representable in a tiny fraction of model parameters. These findings reframe reasoning improvement as sparse *policy selection*, not capability acquisition. We translate this insight into REASONMAXXER, a minimal RL-free method that applies contrastive loss only at entropy-gated decision points, using a few hundred base-model rollouts and no online generation. Across three model families, six scales, and six math reasoning benchmarks, REASONMAXXER matches or exceeds full RL performance while requiring only tens of problems and minutes of single-GPU training, a reduction in training cost of roughly three orders of magnitude.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become the dominant paradigm for improving reasoning in large language models (Guo et al., 2025; Shao et al., 2024; Zeng et al., 2025). Systems such as DeepSeek-R1 (Guo et al., 2025), OpenAI o1 (OpenAI et al., 2026), and Qwen3 (Yang et al., 2025a) demonstrate substantial gains from this pipeline, and the field has broadly adopted RL, typically GRPO (Shao et al., 2024) or PPO (Schulman et al., 2017), as the standard post-training method for mathematical and code reasoning. The implicit assumption underlying this paradigm is that RL, similar to how it discovers novel strategies in games (Silver et al., 2016), enables LLMs to acquire genuinely new reasoning patterns through reward-driven exploration. A growing body of evidence challenges this assumption. Yue et al. (2025) show that while RL improves pass@1, base models achieve higher pass@k at large k: the base model's sampling distribution already contains correct solutions that RL merely promotes. Davis & Recht (2025) prove that popular RL algorithms with binary rewards all reduce to stochastic gradient ascent on monotone transforms of the probability of a correct answer, and that such optimization is only profitable when the base model already succeeds non-trivially. Zhang et al. (2025) confirm this through controlled experiments: RL produces genuine gains only at the model's edge of competence, on problems that are difficult but not yet out of reach. At the token level, Wang et al. (2025c) identify that RL's improvements concentrate at high-entropy "forking tokens" where the model is uncertain which reasoning path to follow, and show that restricting gradient updates to these tokens matches training on all tokens. From a structural angle, Park et al. (2025) find that RL operates through a small number of emergent attention heads. Collectively, these findings converge on an emerging picture: *RL primarily steers the model toward*

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.

committing to solution paths that the base model already contains, rather than inventing genuinely new reasoning strategies.

Despite this growing understanding, a critical gap remains. The works that identify this structure still operate inside the RL framework: Wang et al. (2025c) make RL more efficient rather than eliminating it, Yue et al. (2025) call for improved RL paradigms, and Karan & Du (2025) offer only inference-time alternatives. The natural next question is *whether we can precisely characterize RL's token-level effect and, if that characterization is simple enough, whether the RL optimization loop itself is necessary*.

In this paper, we answer that question through a systematic token-level analysis across multiple model families and RL algorithms. We find that RL's behavioral footprint is strikingly simple: it modifies only 1–4% of token positions, does not introduce tokens outside the base model's top-5 candidates, and concentrates edits at high-entropy *decision points* where the model is uncertain which reasoning branch to take. Using oracle intervention with random controls, we establish that the specific token chosen at these positions matters causally, recovering a large share of RL's gain, while random corrections fail. Crucially, these decision points can be located without any RL-trained model: the base model's own token entropy, which peaks at the positions RL edits, provides a strong proxy for where intervention is useful. We further show that the full correction is low-dimensional, representable in a tiny fraction of model parameters. Together, these findings reframe reasoning improvement as a **sparse *policy selection* problem: committing to the right branch at a handful of uncertainty points**, rather than acquiring new capabilities through expensive exploration.

To test this reframing directly, we construct REASONMAXXER, a minimal RL-free method that exploits the identified structure. REASONMAXXER generates a small set of rollouts from the base model, uses entropy gating to locate decision points, and applies an advantage-weighted contrastive loss exclusively at those positions, while anchoring all other tokens to the base distribution. The method requires no RL, no online generation, and no large-scale compute: it maximizes reasoning performance with a shoestring budget. Across three model families and multiple scales, REASONMAXXER matches or exceeds the performance of models trained with full RL, yet uses only tens of problems, hundreds of rollouts, and minutes of single-GPU training, **reducing training cost by roughly three orders of magnitude**. That so simple a method suffices challenges the prevailing assumption that heavy RL infrastructure is necessary for reasoning improvement.

Our contributions are as follows:

- **Mechanistic characterization of RL for reasoning.** Through token-level analysis across multiple model families and RL algorithms, we show that RL's beneficial effect is a sparse, entropy-localized reranking of tokens the base model already favors, and we establish causality through oracle intervention with random controls.
- **An RL-free method that matches full RL.** We introduce REASONMAXXER, which applies contrastive fine-tuning only at entropy-gated decision points using the base model's own rollouts. It matches or exceeds RL-trained models on math reasoning benchmarks while using orders-of-magnitude less compute and data.
- **Evidence that heavy RL is not a prerequisite.** By showing that a lightweight method can replicate RL's reasoning improvement, we demonstrate that the problem RL solves in this domain is sparse policy selection, not capability acquisition. This suggests that the *community's default investment in full RL pipelines for outcome-based reasoning may be excessive relative to the problem's complexity*.

2 BACKGROUND AND EXPERIMENTAL SETUP

2.1 REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS

We briefly review the RL algorithms used by the baseline models in our study. Given a prompt q with ground-truth answer a, RLVR generates G rollouts $\{o^i\}_{i=1}^G$ from the current policy π_θ and assigns each a binary reward $R^i = \mathbf{1}[\texttt{match}(o^i, a)]$. The dominant algorithm among the baselines we evaluate is Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which computes

per-rollout advantages via group normalization:

$$\hat{A}^i = \frac{R^i - \text{mean}(\{R^j\}_{j=1}^G)}{\text{std}(\{R^j\}_{j=1}^G)}, \tag{1}$$

and updates the policy by maximizing a clipped surrogate objective applied uniformly across all token positions. This uniform application is a key point of contrast with our approach: GRPO distributes gradient across every token in every rollout, despite the evidence (presented in §3) that only a small fraction of positions carry the useful signal. Several baselines use alternative algorithms that share the same core structure. Open-Reasoner-Zero (Hu et al., 2025) employs Proximal Policy Optimization (PPO) (Schulman et al., 2017) with GAE, while other recent work explores REINFORCE-style variants such as RLOO (Ahmadian et al., 2024). All of these methods optimize the same underlying objective: increasing the probability of tokens that lead to correct answers, with the primary differences lying in advantage estimation and regularization strategies. Our mechanistic analysis in §3 studies models trained with GRPO, PPO, and RLOO, and finds the same sparse-correction pattern across all three.

2.2 Token-Level Entropy and Decision Points

For an autoregressive language model π_θ, the token-level generation entropy at position t is defined as

$$H_t = -\sum_{v \in \mathcal{V}} \pi_\theta(v \mid q, o_{<t}) \log \pi_\theta(v \mid q, o_{<t}), \tag{2}$$

where $\mathcal{V}$ is the vocabulary and $o_{<t}$ denotes the tokens generated so far. Positions with high H_t correspond to points where the model distributes probability mass across multiple plausible continuations rather than committing to a single token. Recent work has identified these high-entropy positions as functionally significant: Wang et al. (2025c) show that they act as "forks" steering the model toward different reasoning pathways, and Agarwal et al. (2025) demonstrate that minimizing entropy without labeled data can improve reasoning performance. We refer to positions where H_t exceeds a threshold τ as *decision points*, the subset of the generation where the model's commitment to a reasoning path is genuinely uncertain.

2.3 Models and Baselines

The RL algorithms used by the baselines were introduced in Section 2.1 (GRPO, PPO, and their variants). Table 1 summarises the model families and the specific publicly available RL-trained checkpoints we used in our experiments across the paper. All baselines are trained with verifiable outcome rewards on mathematical reasoning problems.

Table 1: Model families and RL baselines.

Family	Base models	RL baseline	Algorithm & note
Qwen2.5	1.5B, 7B, Math-7B, 32B	SimpleRL-Zoo (Zeng et al., 2025)	GRPO
	1.5B, 7B, 32B	Open-Reasoner-Zero (Hu et al., 2025)	PPO
	Math-7B	Eurus-2-7B-PRIME (Cui et al., 2025)	RLOO
Qwen3	0.6B	GRPO (raw base) (Yang et al., 2025a)	GRPO
	4B	General-Reasoner (Ma et al., 2025)	GRPO + verifier, multi-domain
DeepSeek	R1-Distill-Qwen-1.5B	DeepScaleR	GRPO, context scaling
		STILL-3 (Chen et al., 2025)	PPO
		Open-RS3 (Dang & Ngo, 2025)	GRPO, tight compute
Mistral	7B v0.1	SimpleRL-Zoo (Zeng et al., 2025)	GRPO

3 What RL Actually Changes: Sparse Corrections at Decision Points

Recent work suggests that RL for reasoning primarily steers the model toward solutions it already knows rather than inventing new strategies (Yue et al., 2025). To understand *what* this steering looks like at the token level, we compare the outputs of a base model and its RL-trained counterpart on the same set of prompts.

Our investigation addresses three questions:

1. **How often, and at what kind of positions, does the RL model disagree with the base model?** (§3.1)
2. **Do these token-level disagreements *cause* the observed accuracy gain?** (§3.2)
3. **Can we locate the critical positions without access to the RL model, using only signals from the base model?** (§3.3)

We focus on the four base/RL-tuned pairs introduced in §2.3 and evaluate on MATH-500 with deterministic decoding (T=0).

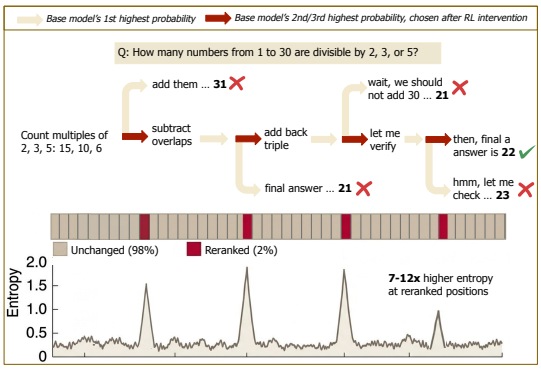

Figure 1: **RL edits are rare, conservative, and concentrated at decision points.**

3.1 Disagreement Is Rare, Conservative, and Concentrated at Decision Points

For each prompt, we generate a response from the base model and, at every token position, we record which token the RL-tuned *teacher* model would have preferred given the identical prefix. Positions are then classified as follows:

$$\begin{aligned} \textsc{Unshifted}: &\ \arg\max \pi_{\text{base}} = \arg\max \pi_{\text{teacher}}, \\ \textsc{Reranked}: &\ \arg\max \pi_{\text{teacher}} \neq \arg\max \pi_{\text{base}},\ \arg\max \pi_{\text{teacher}} \in \text{Top-5}(\pi_{\text{base}}), \\ \textsc{Shifted}: &\ \arg\max \pi_{\text{teacher}} \notin \text{Top-5}(\pi_{\text{base}}). \end{aligned} \tag{3}$$

In words, RERANKED means the teacher promotes a token that was already among the base model's top-5 candidates, whereas SHIFTED would indicate a genuinely new preference.

Table 2: **Token-level divergence is minimal and localized.**

Pair	Reranked	Shifted	Entropy ratio	Mean rank
Qwen2.5-1.5B → GRPO	2.09%	0.02%	7.58×	2.30
Qwen2.5-7B → GRPO	1.03%	0.01%	8.27×	2.14
Qwen2.5-7B → PPO	3.96%	0.12%	9.41×	2.39
Qwen3-4B → GRPO	2.29%	0.02%	12.63×	2.25

The results, summarized in Fig. 1 and Table 2, paint a clear picture. Only 1.0–4.0% of all token positions are reranked, while shifted positions are extremely rare (0.01–0.12%). The teacher's preferred token is, on average, the second most likely token under the base model (mean rank 2.14–2.39). Moreover, the reranked positions have 7–12× higher base-model entropy than unchanged positions. Thus, RL's edits are not only extremely sparse; they are also highly predictable: they occur exactly at high-entropy *decision points* where the model is uncertain which reasoning branch to follow (cf. §2.2). *RL does not introduce novel tokens; it consistently elevates one of the base model's top alternatives at moments of uncertainty.* This explains why prior work observed low perplexity between RL-trained and base models Yue et al. (2025): the promoted token was already a plausible candidate.

3.2 Correcting Only the Disagreements Recovers RL Performance

Having established *where* the two models differ, we now ask whether these differences are causally responsible for the RL model's higher accuracy. We design an oracle intervention: during deterministic generation from the base model, at every position where the teacher disagrees (i.e., the reranked positions from Table 2), we replace the base token with the teacher's preferred token and continue generating from the corrected prefix. As a control, we instead insert a randomly chosen alternative from the base model's top-20 (*random substitution*).

Figure 2 shows the outcome. The oracle intervention reproduces the teacher's pass@1 *exactly* on every pair, while the random substitution baseline performs no better than the base model (often

worse). The fraction of tokens touched by the oracle equals the rerank percentages from Table 2 (1.0–4.0%). Hence, the RL model's entire accuracy advantage can be attributed to a tiny set of precise token choices at decision points. In short, *a handful of token corrections can redirect the full reasoning trajectory*; RL's benefit is not a diffuse effect but is concentrated at a few branch points where the choice of continuation determines the solution path.

3.3 Entropy Alone Identifies the Critical Positions

The oracle experiment relies on the teacher to both locate and correct the important tokens. For a practical RL-free method, we need to locate these positions without the teacher. The strong correlation observed in §3.1 suggests that base-model entropy might serve this role. We therefore test an *entropy-gated* intervention: we replace the base token with the teacher's preferred token at every position where the base-model entropy exceeds a threshold τ, *without* using any information about the teacher's preferences. This probe tells us how well entropy alone can substitute for the teacher's knowledge of *where* to intervene.

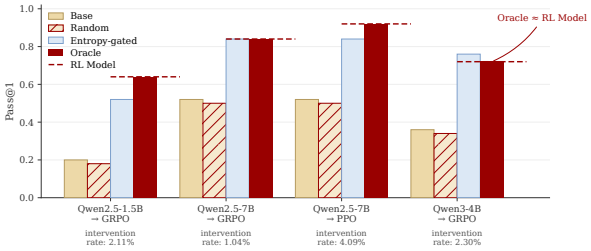

Figure 2: **Oracle correction recovers RL performance exactly, and entropy-gating largely matches it.**

The blue bars in Fig. 2 show the performance of this entropy-gated correction. With only an entropy threshold ($\tau = 1.2$), the intervention matches the teacher exactly on the 7B GRPO pair, closely approaches it on the PPO pair, and substantially improves over the base model on the other pairs, while touching only 1.2–8.3% of tokens. Entropy therefore acts as an effective, fully teacher-free proxy for the decision points that RL would correct. Thus, *the* where *of RL's correction is predictable from the base model's entropy alone*; the remaining challenge is to learn *which* token to substitute at those positions, a problem we solve with ReasonMaxxer (§5).

4 The Correction Is Low-Dimensional

Section 3 showed that RL's beneficial effect is sparse in token space and predictable from the base model's entropy. A natural next question is whether the correction is also simple in parameter space. If replicating the RL model's behavior at decision points required high-dimensional parameter changes, the observed token-level sparsity might be an emergent property of a complex distributed computation, and the full RL optimization loop might still be necessary. Full-parameter RL updates have been found to be sparse but nearly full-rank (Mukherjee et al., 2025), so a low-rank representation is not guaranteed. To test whether RL's *correction* is inherently low-dimensional, we measure how much adapter capacity is needed to capture it.

4.1 Distilling RL into a Low-Rank Adapter

Our diagnostic is a **KL-LoRA** distillation: we attach a LoRA adapter (Hu et al., 2021) to the base model and train only the adapter parameters to minimise the token-level Kullback–Leibler divergence between the adapter-augmented model and the RL-trained teacher:

$$\mathcal{L}_{\text{distill}} = \sum_t \mathrm{KL}(\pi_{\text{teacher}}(\cdot \mid x_{<t}) \,\|\, \pi_{\text{base}+\Delta\theta}(\cdot \mid x_{<t})) \,. \tag{4}$$

We cache the teacher's top-k logits on a set of rollouts generated by the teacher itself. The adapters are trained on only **100 randomly chosen problems**. If a tiny adapter can absorb RL's full distributional change from such a small number of problems, then that change must be fundamentally low-dimensional.

4.2 A Small Adapter Captures RL's Full Correction

Figure 3 presents the results for the four base/RL pairs studied in §3. On both MATH-500 and GSM8K, a LoRA adapter with rank 32 applied to all attention projections (QKVO) matches the RL teacher's accuracy, while modifying only 0.27–0.49% of the base model's parameters.

The adapter sizes above each group (0.3% to 0.5%) make the low-dimensional nature of RL's correction immediately visible. The design is frugal by intent: using only 100 randomly chosen problems, the adapter sees just enough examples of the model's behaviour at critical decision points to capture RL's policy steering.

This reinforces the insight from §3 that RL's signal is concentrated in a few high-entropy locations; a small, targeted dataset suffices because the base model already possesses the necessary vocabulary and reasoning patterns.[1] *Thus, RL's correction is not only sparse in token space but also low-dimensional in parameter space: a tiny adapter, on the order of a fraction of a percent of the model's parameters, captures the entire distributional change.*

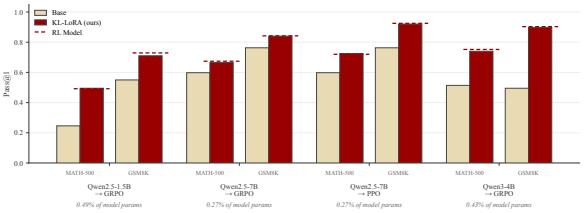

Figure 3: **RL's correction is low-dimensional.**

From Representability to Learnability The KL-LoRA experiment shows that RL's corrective signal is representable in a tiny parameter budget. Recent work has further demonstrated that learning such a signal from scratch with LoRA-constrained RL can match full-parameter RL, indicating that the solution is not only low-dimensional but also accessible within a small parameter space (Wang et al., 2025b). This simplicity suggests that the signal might be learnable without RL's stochastic search, a hypothesis we test directly with REASONMAXXER in the next section.

5 REASONMAXXER – ENTROPY-GATED CONTRASTIVE FINE-TUNING

REASONMAXXER translates the findings of Sections 3 and 4 into a direct, RL-free training procedure. The method generates a small set of base-model rollouts, selects token positions where the base model's entropy is high, and applies a contrastive loss that encourages tokens leading to correct answers while penalizing those that lead to incorrect ones. The following subsections describe the problem selection, entropy-based identification of decision points, and contrastive fine-tuning.

5.1 PROBLEM SELECTION: EXPLOITING THE EDGE OF COMPETENCE

For a collection of math problems with verifiable answers, we sample K completions per problem from the frozen base model at nonzero temperature and compare each completion against the ground-truth answer. From this pool we keep exclusively problems where the base model's pass rate lies strictly between 0 and 1: some rollouts are correct, others are incorrect.

This filter is the direct operationalisation of a property that both prior theoretical work (Davis & Recht, 2025; Zhang et al., 2025) and our own oracle experiments (Section 3.2) have shown to be necessary for learning from outcome feedback. When the base model always succeeds on a problem, there is no incorrect behaviour to penalise; when it always fails, there is no correct behaviour to reinforce. Only the mixed-success regime supplies the two-sided contrastive signal that can distinguish good decisions from bad ones at the same decision points. The filter guarantees that every retained problem contributes this signal. We verify empirically that the exact width of the pass-rate window is not critical; the existence of both correct and incorrect rollouts within a problem is what matters.

5.2 DECISION-POINT IDENTIFICATION VIA ENTROPY

For each retained rollout we compute the per-token entropy of the frozen base model (Eq. 2). A token position t is designated as a *decision point* if $H_t > \tau$, where τ is a model-family-specific threshold chosen so that the marked positions correspond to roughly the top few percent of the model's entropy distribution. We write $\mathcal{D} = \{t : H_t > \tau\}$.

This step rests directly on two findings from Section 3. First, the positions where an RL-trained teacher disagrees with the base model are precisely the high-entropy positions (Table 2, Fig. 1). Second, an entropy-based gate can replace the teacher's disagreement signal without loss of corrective power (Section 3.3). Consequently, $\mathcal{D}$ is a fully teacher-free, principled selection of the locations where the model's behaviour most needs refinement. Because entropy is computed from the base model alone, this stage requires no external supervision beyond the rollouts already generated.

5.3 ADVANTAGE-WEIGHTED CONTRASTIVE LOSS WITH BASE ANCHORING

Given a set of rollouts for a single problem, we compute a per-rollout normalised advantage

$$A_i = \frac{r_i - \bar{r}}{\sigma_r + \epsilon}, \tag{5}$$

[1] Further compression is possible: a rank-8 output-projection adapter matches the full QKVO adapter within a few points on MATH-500 (Appendix B), indicating that RL's correction can be expressed almost entirely through the output layer. We conservatively use the full rank-32 QKVO configuration for REASONMAXXER.

where $r_i \in \{0, 1\}$ indicates whether rollout i arrived at the correct answer, and $\bar{r}, \sigma_r$ are the mean and standard deviation of the correctness indicators for that problem. This normalisation centres the advantages so that correct and incorrect rollouts receive symmetric positive and negative weights, preventing class imbalance from distorting the gradient.

The training loss is the sum of two terms. At decision points we apply an advantage-weighted cross-entropy,

$$\mathcal{L}_{\text{dec}} = -\sum_{t \in \mathcal{D}} A_i \cdot \log p_\theta(x_t \mid x_{<t}), \tag{6}$$

which increases the likelihood of the observed token when the rollout was correct ($A_i > 0$) and decreases it when the rollout was incorrect ($A_i < 0$). The model is therefore shaped to reproduce the token-level choices that preceded a correct final answer and to avoid those that preceded an incorrect one.

At all positions *outside* the decision set $\mathcal{D}$, we minimise the Kullback–Leibler divergence to the frozen base model,

$$\mathcal{L}_{\text{anchor}} = \sum_{t \notin \mathcal{D}} \text{KL}\big(p_{\text{base}}(\cdot \mid x_{<t}) \parallel p_\theta(\cdot \mid x_{<t})\big). \tag{7}$$

This anchor term preserves the base model's behaviour everywhere that the mechanistic analysis found RL to have no effect, and it prevents the small adapter from overfitting to spurious correlations in the limited training set. The total loss is $\mathcal{L} = \mathcal{L}_{\text{dec}} + \lambda \mathcal{L}_{\text{anchor}}$, the λ balancing the two objectives.

Architecturally, REASONMAXXER implements this loss through a **LoRA adapter** (Hu et al., 2021) attached to the base model. The base model remains frozen; only the low-rank adapter matrices are updated. This choice is the natural consequence of the low-dimensionality established in Section 4: if a rank-32 adapter containing well under one percent of the model's parameters can absorb RL's entire distributional change, then the same parameter budget is more than sufficient to learn the contrastive signal directly from the base model's own rollouts. Implementation details are in Appendix C.

6 EXPERIMENTS

We benchmark REASONMAXXER on six mathematical reasoning benchmarks against publicly available RL-trained models spanning three model families and multiple RL algorithms, and we analyze its performance, efficiency, and critical design choices

6.1 EXPERIMENTAL SETUP

Benchmarks and evaluation protocol. We evaluate on six standard mathematical reasoning benchmarks: MATH-500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), AMC 2023, AIME 2024, Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). For AMC 2023 and AIME 2024, which contain few problems, we report *avg@8* (average pass@1 over eight independent generations) to reduce variance; for all other benchmarks we report standard pass@1 from a single generation. Experiment details are given in Appendix D.

Training configuration for REASONMAXXER. We implement REASONMAXXER as a rank-32 LoRA adapter on all attention projections, leaving the base model frozen. From a pool of 150 math problems balanced across difficulty levels we sample 20 rollouts per problem and retain 50 problems on which the base model exhibits mixed success, yielding 1000 training sequences. Entropy-gated decision points are selected by sweeping the threshold τ on a small held-out set, and the adapter is trained with the advantage-weighted contrastive loss described in Section 5. We train for a single epoch and select the final checkpoint on a fixed 50-problem validation split. Full hyper-parameters and optimizer settings are provided in Appendix C.

Cost estimation. To quantify efficiency, we report the estimated monetary cost of training for every method in Table 3. For REASONMAXXER, costs are directly measured from our runs on NVIDIA RTX Pro 6000 (96 GB) GPUs using RunPod on-demand pricing and include rollout generation, entropy scoring, the τ sweep, and checkpoint selection. Baseline costs are either taken from published reports or inferred from the official training scripts, hardware configuration, and on-demand pricing of the corresponding GPU type (detailed in Appendix E). All costs are rounded to the nearest US dollar; italicised figures indicate estimates where exact numbers were not publicly documented.

6.2 RESULTS AND ANALYSIS

Table 3 presents the full set of results. We structure the discussion around three key findings.

Table 3: **ReasonMaxxer matches or exceeds RL baselines at dramatically lower training cost.** Best score per row in **bold**; shaded rows report our method. Italicised costs are estimates.

Model	GSM8K	MATH 500	Minerva	Olymp. Bench	AIME24 avg@8	AMC23 avg@8	Avg.	Cost (USD)
Qwen2.5 Family Models								
Qwen2.5-1.5B	0.561	0.298	0.044	0.093	0.004	0.094	0.182	
↪ + SimpleRL-Zoo	**0.733**	0.496	0.107	**0.189**	**0.038**	0.156	**0.287**	*$200*
↪ + Open-Reasoner-Zero	0.629	0.436	0.088	0.160	0.025	0.138	0.246	*$1,200*
↪ + ReasonMaxxer	0.710	**0.502**	**0.114**	0.167	0.029	**0.169**	0.282	**$4**
Qwen2.5-7B	0.751	0.586	0.129	0.294	0.046	0.300	0.351	
↪ + SimpleRL-Zoo	0.851	0.656	0.132	0.358	0.088	0.375	0.410	$600
↪ + Open-Reasoner-Zero	**0.924**	**0.732**	**0.213**	**0.454**	**0.121**	0.463	**0.485**	*$6,300*
↪ + ReasonMaxxer	0.918	0.706	0.206	0.411	0.113	**0.475**	0.472	**$5**
Qwen2.5-Math-7B	0.450	0.374	0.088	0.091	0.092	0.278	0.229	
↪ + SimpleRL-Zoo	**0.827**	**0.706**	0.151	0.344	0.188	**0.550**	0.461	$600
↪ + PRIME-Zero	0.543	0.644	0.132	0.341	0.171	0.459	0.382	*$190*
↪ + ReasonMaxxer	0.816	0.674	**0.246**	**0.365**	**0.192**	0.484	**0.463**	**$5**
Qwen2.5-32B	0.836	0.548	0.143	0.275	0.017	0.300	0.353	
↪ + SimpleRL-Zoo	0.876	0.642	0.180	0.327	0.067	0.363	0.409	$5,737
↪ + Open-Reasoner-Zero	**0.945**	**0.646**	**0.272**	0.337	0.067	0.350	0.436	*$103,000*
↪ + ReasonMaxxer	0.865	0.636	0.228	**0.356**	**0.117**	**0.438**	**0.440**	**$25**
Mistral, DeepSeek, and Qwen3 Models								
Mistral-7B-v0.1	0.070	0.199	0.056	0.000	0.007	0.005	0.056	
↪ + SimpleRL-Zoo	**0.076**	0.212	0.025	0.000	0.011	0.005	0.055	$600
↪ + ReasonMaxxer	0.072	**0.317**	**0.069**	0.000	**0.066**	**0.010**	**0.089**	**$8**
DeepSeek-R1-Distill-1.5B	0.775	0.436	0.136	0.208	0.092	**0.266**	0.319	
↪ + DeepScaleR	0.793	0.502	0.162	0.251	0.075	0.259	0.340	$4,500
↪ + STILL-3	0.800	0.536	0.140	0.294	**0.129**	0.238	0.356	*$2,268*
↪ + Open-RS3	0.770	0.454	0.114	0.201	0.079	0.263	0.314	$42
↪ + ReasonMaxxer	**0.825**	**0.662**	**0.213**	**0.356**	0.117	0.231	**0.401**	**$4**
Qwen3-0.6B	0.489	0.338	0.059	0.126	0.000	0.125	0.189	
↪ + GRPO	0.503	0.372	0.059	0.146	0.008	0.109	0.200	*$100*
↪ + ReasonMaxxer	**0.656**	**0.470**	**0.096**	**0.179**	**0.017**	**0.253**	**0.278**	**$4**
Qwen3-4B	0.497	0.514	0.085	0.300	0.083	0.216	0.282	
↪ + General-Reasoner	0.916	**0.746**	0.213	**0.491**	0.021	0.050	0.406	*$4,600*
↪ + ReasonMaxxer	**0.919**	0.660	**0.305**	0.403	**0.096**	**0.472**	**0.476**	**$4**

REASONMAXXER matches full RL on clean comparisons. The most direct test compares REASONMAXXER against publicly available RL trained models that were trained from the same raw base models without additional distillation or SFT stages. On Qwen2.5-1.5B, REASONMAXXER achieves 50.2% on MATH-500 versus SimpleRL-Zoo's 49.6% and Open-Reasoner-Zero's 43.6%, while costing $4 compared to $200 and $1200 respectively. On Qwen2.5-7B, it reaches 70.6% vs. 65.6% (SimpleRL-Zoo) and 73.2% (Open-Reasoner-Zero), again at a tiny fraction of the cost. The pattern holds across the Qwen2.5-Math-7B and Qwen2.5-32B variants, as well as for the Mistral-7B and Qwen3 models. In every case, we perform on par with or better than the RL baseline while reducing training cost by two to three orders of magnitude. This confirms that the sparse policy-selection signal identified in Sections 3–4 is not merely a diagnostic artifact; it is *the* signal that RL itself ultimately captures, and REASONMAXXER recovers it without the RL optimization.

Performance generalizes beyond pure RL settings. Several baselines in Table 3 incorporate additional training strategies beyond outcome-only RL: DeepSeek-R1-Distill-1.5B baselines start from a distilled checkpoint, Qwen3-4B's General-Reasoner uses a model-based verifier and multi-domain data, and STILL-3 employs iterative RL on a curated dataset. Despite having access to none of these enhancements, REASONMAXXER still matches or exceeds their accuracy on the majority of benchmarks. These results indicate that a substantial fraction of the gains attributed to sophisticated post-training pipelines actually originates from the same sparse policy-selection mechanism that REASONMAXXER isolates and directly optimizes.

Efficiency and scalability. The computational and data efficiency of REASONMAXXER are equally notable. In terms of compute, REASONMAXXER completes in single-digit GPU-hours across all models, while the RL baselines require hundreds to tens of thousands of GPU-hours (Table 3); the average training cost is less than $10, compared with $100 to $100,000 for the RL baselines. In terms of data, REASONMAXXER trains on 50 problems. SimpleRL-Zoo uses approximately 8,000 MATH problems for its GRPO runs, and Open-Reasoner-Zero trains on 57,000 math and reasoning problems. This gap of over two orders of magnitude in training data is not an incidental optimisation; it follows directly from the mechanistic picture established in Sections 3 and 4. Because RL's useful signal is

concentrated at a sparse set of high-entropy decision points, a handful of mixed-success problems supplies sufficient contrastive supervision to capture the full policy-steering correction. For the same reason, REASONMAXXER generates rollouts once and offline, and trains only a lightweight adapter; its cost scales with the number of training sequences rather than with the product of model size and on-policy iterations, keeping it practical even for larger models. In summary, REASONMAXXER demonstrates that the essential reasoning improvement from outcome-based RL can be obtained by a simple, data-efficient contrastive procedure, challenging the necessity of heavy RL infrastructure.

Beyond mathematical reasoning. Although our primary controlled evaluation focuses on mathematics, we additionally evaluate REASONMAXXER on symbolic reasoning and commonsense QA. Direct non-math training improves over the base model on ProofWriter (Tafjord et al., 2021), CommonsenseQA, and OpenBookQA (Mihaylov et al., 2018), while a token-level analysis on ProofWriter reproduces the same sparse, uncertainty-localized reranking pattern observed in mathematics. Full results, including zero-shot math-to-non-math transfer, are reported in Appendix F.

6.3 ABLATION STUDIES

Sensitivity to τ. Figure 4 shows pass@1 on MATH-500 and GSM8K as τ varies from 1.0 to 2.2, alongside the corresponding mean fraction of tokens that are gated as decision points (annotated below each τ). Performance is robust over a broad range: the optimal MATH-500 score (0.50) is achieved at $\tau = 1.4$ (5.2% of tokens), and the optimal GSM8K score (0.72) at the same threshold, matching the RL model. A second peak appears at $\tau = 1.8$ (2.6% of tokens) where the decision fraction closely matches RL's observed intervention rate of 2.11% (Table 2). This indicates that REASONMAXXER does not require a precise replication of RL's sparsity; as long as the threshold selects a plausible set of high-entropy positions, the contrastive signal is effective. The broad plateau confirms that entropy gating is a reliable, teacher-free proxy for locating RL's intervention sites, as argued in Section 3.3.

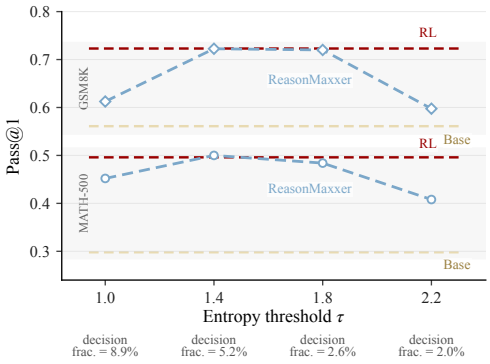

Figure 4: **Pass@1 on MATH-500 and GSM8K for Qwen2.5-1.5B as τ varies.**

Necessity of the contrastive term. To isolate the contribution of the negative gradient, we compare the full REASONMAXXER loss against a variant where only correct rollouts ($A_i > 0$) are used (positive-only training, equivalent to supervised fine-tuning on correct trajectories). On Qwen2.5-1.5B, positive-only training raises MATH-500 pass@1 from 0.298 (base) to 0.398, a non-trivial improvement that confirms the value of targeting decision points. However, it remains far below the RL model (0.496) and full REASONMAXXER (0.502). The contrastive term that suppresses incorrect decisions thus contributes roughly half of the total gain over the base model, and it is the combination of positive reinforcement and negative suppression that together capture RL's full policy-steering effect. This directly supports the design choice: a two-sided contrastive loss exploits the edge-of-competence signal discussed in Section 5.1, teaching the adapter not only which tokens to prefer but also which tokens to avoid.

7 CONCLUSION

We set out to answer whether the RL optimization loop is necessary for improving reasoning in LLMs. Through systematic token-level analysis across multiple model families and RL algorithms, we showed that RL's useful effect on math reasoning is a sparse, predictable, and low-dimensional correction. We then demonstrated that this correction can be obtained without RL at all with REASONMAXXER. These results reframe reasoning improvement as a *sparse policy-selection problem*: the model already knows the necessary reasoning paths; it only needs to commit to the right branch at a handful of critical moments. The RL optimization loop, while capable of discovering this correction, is not a prerequisite for it. Our findings suggest that the community's default investment in heavy RL infrastructure for post-training may be disproportionate to the complexity of the problem that is actually being solved. Recognizing this simplicity opens the door to a generation of far more efficient post-training methods.

AI USE STATEMENT

We used generative AI tools to assist with editing and improving the clarity, organization, and presentation of the manuscript, including rephrasing text, and suggesting revisions based on author-provided material.

All scientific ideas, experimental design decisions, analyses, results, and conclusions were developed and validated by the authors. The authors manually reviewed and verified all AI-assisted text and references against the corresponding sources and take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We provide the details needed to reproduce our experiments throughout the main paper and appendix. Section 5 specifies the REASONMAXXER training procedure and objective. Appendix C reports the optimization, rollout-generation, LoRA, and model-family-specific hyperparameters, while Appendix D provides the exact prompting and evaluation procedures. Appendix E documents the hardware and methodology used to measure or estimate training costs. Additional ablations and robustness analyses are reported in the corresponding appendix sections.

REFERENCES

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning, 2025. URL https://arxiv.org/abs/2505.15134.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Ömer Faruk Akgül, Yusuf Hakan Kalaycı, Rajgopal Kannan, Willie Neiswanger, and Viktor Prasanna. Lynx: Learning dynamic exits for confidence-controlled reasoning. *arXiv preprint arXiv:2512.05325*, 2025.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. An empirical study on eliciting and improving r1-like reasoning models, 2025. URL https://arxiv.org/abs/2503.04548.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 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, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Quy-Anh Dang and Chris Ngo. Reinforcement learning for reasoning in small llms: What works and what doesn't. *arXiv preprint arXiv:2503.16219*, 2025.

Damek Davis and Benjamin Recht. What is the objective of reasoning with reinforcement learning? *arXiv preprint arXiv:2510.13651*, 2025.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun

Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

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

Dan Hendrycks, Collin Burns, 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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 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, 2025. URL https://arxiv.org/abs/2503.24290.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think, 2025. URL https://arxiv.org/abs/2510.14901.

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, 2022. URL https://arxiv.org/abs/2206.14858.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-reasoner: Advancing llm reasoning across all domains, 2025. URL https://arxiv.org/abs/2505.14652.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models, 2025. URL https://arxiv.org/abs/2505.11711.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Bohan Zhang, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward,

Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wenting Zhan, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2026. URL https://arxiv.org/abs/2412.16720.

Yein Park, Minbyul Jeong, and Jaewoo Kang. Thinking sparks!: Emergent attention heads in reasoning models during post training. *arXiv preprint arXiv:2509.25758*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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, 2024. URL https://arxiv.org/abs/2402.03300.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models, 2025. URL https://arxiv.org/abs/2503.16419.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, 2021.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Burak Bilgin, Ollie Liu, Deqing Fu, and Willie Neiswanger. Resa: Transparent reasoning models via saes, 2025a. URL https://arxiv.org/abs/2506.09967.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Burak Bilgin, Ollie Liu, and Willie Neiswanger. Tina: Tiny reasoning models via lora, 2025b. URL https://arxiv.org/abs/2504.15777.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen

Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning, 2025c. URL https://arxiv.org/abs/2506.01939.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example, 2025d. URL https://arxiv.org/abs/2504.20571.

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*, 2025a.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models, 2025b. URL https://arxiv.org/abs/2504.15895.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023. URL https://arxiv.org/abs/2308.01825.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 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.

Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models. *arXiv preprint arXiv:2512.07783*, 2025.

A Related Work

Many of the works most directly relevant to this paper were already discussed in the introduction (Section 1). Here we provide a more complete discussion, situating our study within the broader literature.

What RL does for reasoning. A growing body of work has questioned whether RLVR expands or merely refines the base model's reasoning capabilities. Yue et al. (2025) apply pass@k analysis to show that RL-trained models' reasoning paths lie within the base model's sampling distribution. Davis & Recht (2025) prove that popular RL algorithms with binary rewards reduce to stochastic gradient ascent on monotone transforms of the probability of a correct answer, implying that optimisation is profitable only when the base model already succeeds non-trivially. Zhang et al. (2025) confirm this through controlled experiments, finding that RL produces genuine gains only at the model's edge of competence. Wang et al. (2025d) demonstrate that a single training example can yield large improvements, suggesting that the corrective signal RL imparts is highly compressible. Our work provides a token-level mechanistic characterisation that unifies these observations.

Entropy and decision points in LLM reasoning. Wang et al. (2025c) identify high-entropy "forking tokens" as the locus of RL's gradient signal and show that restricting GRPO updates to these positions matches training on all tokens. Agarwal et al. (2025) demonstrate that entropy minimisation without labeled data improves reasoning performance. Park et al. (2025) find that RL operates through a small number of emergent attention heads. Together, these studies indicate that RL's effect on large language models is concentrated in a small number of structural and representational units. The present work builds on these insights by establishing causality through oracle intervention and by showing that the sparse signal can be captured without RL.

RL post-training baselines. Our main experiments compare against a diverse set of publicly available RL-trained models that span multiple algorithms and training strategies. SimpleRL-Zoo (Zeng et al., 2025) provides GRPO-trained checkpoints across ten base models, enabling systematic comparison. Open-Reasoner-Zero (Hu et al., 2025) scales PPO on base models without distillation, demonstrating that vanilla PPO with GAE suffices for reasoning improvement. PRIME (Cui et al., 2025) introduces implicit process rewards for online RL, combining outcome supervision with dense token-level feedback. General-Reasoner (Ma et al., 2025) extends GRPO with a model-based verifier across diverse domains beyond mathematics. On the distilled-model track, DeepScaleR applies iterative GRPO with context-length scaling, STILL-3 (Chen et al., 2025) employs a three-stage pipeline combining imitation, exploration, and self-improvement, and Open-RS3 (Dang & Ngo, 2025) investigates GRPO under tight compute constraints.

RL-free alternatives for reasoning. Several methods improve reasoning without RL. STaR (Zelikman et al., 2022) and rejection sampling fine-tuning (Yuan et al., 2023) train on the model's own correct solutions using uniform token-level losses. Best-of-N and MCMC sampling (Karan & Du, 2025) improve reasoning at inference time without modifying the policy. DPO (Rafailov et al., 2023) offers an offline preference-based alternative that operates at the sequence level. The method proposed in this paper, ReasonMaxxer, differs from these approaches by explicitly targeting the sparse, entropy-localised decision points identified in the mechanistic analysis.

Efficiency in reasoning. Making LLM reasoning more computationally efficient has been approached from several complementary angles. On the training side, parameter-efficient fine-tuning methods such as LoRA (Hu et al., 2021) and QLoRA (Dettmers et al., 2023) have dramatically reduced the cost of adapting large models, and a growing body of work extends these ideas to reasoning specifically: Resa (Wang et al., 2025a) uses sparse autoencoder tuning to extract reasoning abilities from a source model and guide lightweight supervised fine-tuning, while TINA (Wang et al., 2025b) shows that LoRA-constrained RL can match full-parameter RL on reasoning benchmarks at a fraction of the cost. On the inference side, a separate line of work targets the overthinking phenomenon in large reasoning models by terminating generation once sufficient confidence is reached; DEER (Yang et al., 2025b) proposes a training-free early-exit mechanism that monitors reasoning transition points and self-truncates chain-of-thought, and LYNX (Akgül et al., 2025) extends this idea with lightweight hidden-state probes and conformal prediction for distribution-free confidence

control. Broader surveys such as (Sui et al., 2025) provide structured taxonomies of these and related approaches. Together, these works point toward a trend where strong reasoning performance is sought with substantially lower computational overhead than that of current RL-heavy pipelines.

B Detailed KL-LoRA Compression Ablations

Section 4 demonstrated that a rank-32 QKVO adapter trained via KL distillation captures RL's full correction on reasoning tasks. Table 4 reports a more aggressive compression study on Qwen2.5-1.5B, varying both the rank and the targeted attention modules.

Table 4: **Adapter rank and module ablation.** Even a rank-8 adapter applied only to the output projection ($\mathbf{W}_O$) approaches the full rank-32 QKVO adapter, indicating that RL's correction is concentrated in the output layer.

Configuration	Parameters	Pass@1
Base model	–	0.233
Qwen2.5-1.5B → GRPO	1.54B (100%)	0.492
KL-LoRA, $\mathbf{W}_{QKVO}$, rank 32	8.7M (0.49%)	0.495
KL-LoRA, $\mathbf{W}_{QKVO}$, rank 16	4.4M (0.25%)	0.492
KL-LoRA, $\mathbf{W}_{QKVO}$, rank 8	2.2M (0.12%)	0.487
KL-LoRA, $\mathbf{W}_O$ only, rank 8	688K (0.04%)	0.482

The rank-8 $\mathbf{W}_{QKVO}$ adapter already matches the RL teacher, and even an aggressively small output-projection adapter ($\mathbf{W}_O$ only, rank 8, 688 K parameters) lags by only 1 point on MATH-500. This suggests that RL's correction can be expressed almost entirely through the output layer: the base model already attends to mostly the right evidence, and RL mainly changes how the attended information is written into the hidden state to produce better next-token choices. We leave the exploration of such extremely compressed adapters for future work, noting that while they are sufficient to *represent* RL's signal, learning the signal from scratch in such a constrained space may require different optimization strategies. Throughout the main paper we conservatively use the full rank-32 $\mathbf{W}_{QKVO}$ configuration for ReasonMaxxer.

C Implementation Details

We provide the full training and architectural details for the KL-LoRA distillation experiments reported in Section 4 and for ReasonMaxxer (Section 5).

KL-LoRA Distillation

The KL-LoRA adapters in Section 4 are trained with a manually implemented KL divergence over the teacher's top-64 logits. The teacher model (SimpleRL-Zoo GRPO checkpoint) generates rollouts with temperature 0.6 and top-p 0.95 (seed 44), and the distribution is cached. The student (base model + LoRA) is trained for three epochs with batch size 2 and gradient accumulation 8, using the AdamW optimizer, learning rate 10^{-4}, weight decay 10^{-2}, and a 10 % warmup ratio. The objective is averaged over generated-token positions only. LoRA settings for the full-rank and compressed variants are given in Table 5.

ReasonMaxxer

ReasonMaxxer adapters are trained with AdamW using default betas $(0.9, 0.999)$ and $\epsilon = 10^{-8}$, with a linear warmup followed by linear decay to zero. All models are trained for one epoch with batch size 1 and gradient accumulation over 8 steps; the exact number of optimizer steps therefore depends on the number of training sequences, but the hyperparameters are identical across model scales. The decision loss $\mathcal{L}_{\text{dec}}$ and the KL anchor $\mathcal{L}_{\text{anchor}}$ are each averaged over their respective token masks (decision points for $\mathcal{L}_{\text{dec}}$, all other valid prediction tokens for $\mathcal{L}_{\text{anchor}}$). Decision points are defined exclusively on generated completion tokens; prompt tokens and padding positions are

Table 5: KL-LoRA distillation hyper-parameters.

Hyper-parameter	Value
Teacher checkpoint	Qwen2.5-1.5B-SimpleRL-Zoo
Teacher top-k (cached)	64
Rollout temperature / top-p	0.6 / 0.95
Rollout seed	44
Epochs	3
Batch size / gradient accumulation	2 / 8
Learning rate	10^{-4}
Weight decay	0.01
Warmup ratio	10 %
LoRA rank (full $\mathbf{W}_{QKVO}$)	32, 16, 8
LoRA alpha (full $\mathbf{W}_{QKVO}$)	64, 32, 16
LoRA rank ($\mathbf{W}_O$-only)	8
LoRA alpha ($\mathbf{W}_O$-only)	16
LoRA dropout	0.05

excluded. Advantages are clipped per rollout to $[-2.5, 2.5]$ before token weighting. The KL anchor is computed over all valid non-decision prediction tokens in the truncated sequence (including both prompt and completion tokens), and the total loss is $\mathcal{L} = \mathcal{L}_{\text{dec}} + 0.2\,\mathcal{L}_{\text{anchor}}$. Table 6 summarises the common hyper-parameters.

Table 6: Common REASONMAXXER hyper-parameters.

Hyper-parameter	Value
Optimizer	AdamW
Learning rate schedule	Linear warmup + linear decay
Warmup steps (Qwen/DeepSeek)	50
Warmup steps (Mistral)	30
Epochs	1
Batch size / gradient accumulation	1 / 8
Gradient clipping	1.0
Weight decay (Qwen, DeepSeek)	0.0
Weight decay (Mistral)	0.01
LoRA rank / alpha / dropout	32 / 64 / 0.0
LoRA target modules	Q, K, V, O
Max sequence length	8192 (right truncation)
Rollout generation temperature / top-p	0.6 / 0.95
Advantage clipping range	$[-2.5, 2.5]$
KL anchor weight λ	0.2
Validation split	50 problems
Random seed	42

Family-specific overrides prompt styles are listed in Table 7. See Appendix D for exact templates).

Table 7: REASONMAXXER family-specific prompt styles.

Model	Prompt style
Qwen2.5-1.5B, 7B, 32B	qwen_boxed
Qwen3-0.6B, 4B	qwen_boxed
DeepSeek-R1-Distill-1.5B	chat_template
Mistral-7B-v0.1	llama_abel

D PROMPTING AND ANSWER EXTRACTION

The exact prompt templates and answer extraction rules used for each model family are reported below. Within a family, the same template and extraction are applied to the base model, the RL baselines, and the REASONMAXXER adapter, ensuring a fair comparison.

PROMPT TEMPLATES

Qwen2.5 and Qwen3. Both families use a raw completion prompt without a chat template.

```
Solve the following math problem step by step.  Put your final
answer in \boxed{}.
Problem:  {problem}
Solution:
```

Mistral-7B. The prompt follows a simple instruction-answer format, without a chat template.

```
Question:
{problem}
Answer:
Let's think step by step.
```

DeepSeek-R1-Distill. The native chat template is applied with a single user message.

```
Solve the following math problem step by step.  Give a concise
solution and put your final answer in \boxed{}.
Problem:  {problem}
```

E COST ESTIMATION DETAILS

Table 3 reports monetary training costs for all methods. Here we describe how those figures were obtained. Whenever a baseline paper explicitly reports wall-clock time and hardware, we use those numbers directly. When such information is not published, we infer GPU-hours from the official training scripts and documented hyperparameters, then convert to cost using RunPod on-demand pricing as of Apr 28, 2026 ($2.49 per H100-hour for H100-80G instances; $2.69 per H100-hour for H100-SXM instances where specified; $1.89 per GPU-hour for RTX Pro 6000 instances). In all cases we note whether the cost figure is a direct measurement or an estimate.

ReasonMaxxer. All ReasonMaxxer costs are direct measurements on the authors' hardware (4× NVIDIA RTX Pro 6000 Blackwell, 96 GB GDDR7). The reported GPU-hours include rollout generation, entropy scoring, a sweep over $\tau \in \{1.2, 1.4, 1.6, 1.8\}$, training, and final checkpoint selection on a hold-out set. Cost is computed at the RunPod on-demand rate for that GPU type.

SimpleRL-Zoo. For Qwen2.5-7B and larger models, SimpleRL-Zoo (Zeng et al., 2025) directly reports the number of GPUs and training hours. For the 1.5B variant, which does not have a separately published wall-clock, we estimate GPU-hours by scaling the per-step generation time of the 7B run by model size, holding the training configuration constant (1024 prompts × 8 rollouts × ∼100 GRPO steps)[2]. The Mistral-7B run uses the same hardware and step count as the Qwen2.5-7B group.

Open-Reasoner-Zero. Open-Reasoner-Zero (Hu et al., 2025) does not report wall-clock for any model size. We estimate GPU-hours from the public PPO recipes in the project repository[3]. The

[2]The public training command is available at `https://github.com/hkust-nlp/simpleRL-reason`.

[3]See `playground/orz_1p5b_ppo.py` and `playground/orz_7b_ppo.py` in the official Open-Reasoner-Zero codebase: `https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero`.

estimation procedure uses the documented hardware configuration (number of nodes, GPUs per node), the number of prompts and rollouts per step, and the step counts inferred from the training curves in the paper. Per-step time is calibrated against SimpleRL-Zoo's published figures for a comparable model size, with a 1.5–2× overhead factor to account for the PPO critic and GAE computation absent in GRPO. The final point estimates are reported as midpoints of plausible ranges; italicised costs in Table 3 reflect this estimation.

PRIME-Zero. PRIME-Zero (Cui et al., 2025) publishes per-step wall-clock time and the total number of RL steps. We multiply the two to obtain GPU-hours and convert using the RunPod rate for the reported GPU type. The training data size uses the number of prompts consumed during RL, approximately 13K out of the full Eurus-2-RL dataset.

General-Reasoner (Qwen3-4B). Ma et al. (Ma et al., 2025) state that the 4B model is trained on 4 nodes × 8 H100 GPUs for around 2 days, giving ∼1,536 H100-hours. We adopt this figure directly.

DeepSeek-based baselines. DeepScaleR self-reports 3,800 A100-hours and a total cost of ∼$4,500, which we use as given. STILL-3 (Chen et al., 2025) does not provide hardware details; the cost figure in Table 3 is taken from the comparison table in Open-RS3 (Dang & Ngo, 2025), which estimates 1,200 A100-hours on 8× A100-80GB. Open-RS3 self-reports 96 A40-hours, which we convert to cost at the RunPod A40 rate.

Mistral-7B. The SimpleRL-Zoo run for Mistral-7B uses the same hardware and step count as the Qwen2.5-7B group, with simplified prompts. We therefore assign it the same GPU-hour estimate (240 H100-hours). The training data is 8K Easy-split problems.

In all cases, the cost figures in Table 3 are rounded to the nearest US dollar. The italicisation indicates estimates where the exact wall-clock was not directly published by the original authors.

F Generalization Beyond Mathematical Reasoning

The main experiments focus on mathematical reasoning, where exact-answer verification provides a clean setting for controlled comparison. Here we provide complementary evidence on symbolic reasoning and commonsense question answering to test whether the observed behavior is specific to mathematics.

Direct non-math training. We apply the same ReasonMaxxer training procedure to ProofWriter (Tafjord et al., 2021) for symbolic reasoning, CommonsenseQA, and OpenBookQA (Mihaylov et al., 2018). For each dataset, training and evaluation examples are disjoint. Table 8 reports the results.

Table 8: **Direct training on non-math reasoning tasks.** ReasonMaxxer improves over the base model across all three settings.

Dataset	Base	SimpleRL-Zoo	ReasonMaxxer
ProofWriter	0.3600	0.4100	**0.4700**
CommonsenseQA	0.5933	0.6067	**0.7300**
OpenBookQA	0.6057	0.6029	**0.6314**

ReasonMaxxer improves over the base model in all three settings and exceeds the corresponding SimpleRL-Zoo checkpoint in these evaluations. The largest gain is on CommonsenseQA, where accuracy increases from 0.5933 to 0.7300.

Zero-shot transfer from mathematics. We additionally evaluate the math-trained ReasonMaxxer checkpoint on ProofWriter and CommonsenseQA without further adaptation. Results are shown in Table 9.

Table 9: **Zero-shot transfer from mathematics.** Evaluation of the math-trained REASONMAXXER checkpoint on non-math tasks without further adaptation.

Dataset	Base	SimpleRL-Zoo	REASONMAXXER
ProofWriter	0.3600	**0.4100**	**0.4120**
CommonsenseQA	0.5933	0.6067	**0.6200**

The math-trained checkpoint improves over the base model on both datasets. We interpret this only as evidence that the learned correction is not entirely specific to the mathematical training distribution; these experiments are not intended to establish broad cross-domain transfer.

Token-level analysis on ProofWriter. We repeat the base/RL analysis from Section 3 on 100 held-out ProofWriter examples. The RL-trained model reranks the base model's preferred token at 2.74% of positions, while the reranked positions have $3.42\times$ higher base-model entropy than unchanged positions. We observe no shifted positions in this experiment.

Thus, the same qualitative pattern observed in mathematics also appears in this symbolic-reasoning setting: RL edits remain sparse, stay within the base model's existing top alternatives, and concentrate at high-uncertainty positions. We view these results as supporting evidence beyond mathematics rather than a general claim about arbitrary reasoning domains.

G ALTERNATIVE DECISION-POINT SIGNALS

Section 3 uses token entropy to identify the high-uncertainty positions at which RL modifies the base policy. To test whether this localization is specific to entropy, we compare four statistics computed from the base model's next-token distribution: entropy, the top-2 probability ratio p_2/p_1, top-1 uncertainty $1 - p_1$, and tail mass $1 - p_1 - p_2$. We evaluate each signal by its ability to identify RL-reranked positions and positions at which the oracle intervention substitutes the RL-preferred token.

Table 10: **Alternative uncertainty signals.** AUROC for identifying RL-reranked and oracle-intervention positions from the base model's next-token distribution.

Signal	RL reranking	Oracle intervention
Entropy	0.956	0.959
Top-2 ratio (p_2/p_1)	**0.976**	**0.977**
Top-1 uncertainty ($1 - p_1$)	0.972	0.974
Tail mass ($1 - p_1 - p_2$)	0.909	0.898

All four signals provide substantial localization power, with the top-2 ratio and top-1 uncertainty performing slightly better than entropy. This indicates that the effect identified in Section 3 is not entropy-specific: RL corrections concentrate at positions where the base model assigns substantial probability to competing continuations. We retain entropy in REASONMAXXER because it is a simple and effective proxy for this local uncertainty.