TASK REFORMULATION ENABLES REINFORCEMENT LEARNING FROM HARD REASONING PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has improved the reasoning abilities of large language models (LLMs), but it cannot learn from problems that are too difficult to solve under the current policy, since these yield no reward signal and therefore no gradient. We propose a simple solution based on task reformulation. We transform challenging open-ended problems into cognitively simpler variants (such as multiple-choice and cloze formats) that preserve the original answer while reducing the effective search space and providing denser learning signals. These variants span a spectrum from discriminative to generative tasks, so a model can first learn from the structured, easier formats and transfer that knowledge back to the original problems. Building on this insight, we introduce Cog-DRIFT, a framework that automatically constructs reformulated variants and organizes them into an adaptive curriculum, progressing from easier to harder formats as the model improves. This lets the model learn from problems that previously yielded zero signal under standard RL post-training, without distilling reasoning from any stronger external model. Cog-DRIFT improves on the originally unsolvable hard problems (+10.11% for Qwen and +8.64% for Llama in absolute gains) and also generalizes to held-out datasets. Across two models and six challenging reasoning benchmarks, it outperforms standard GRPO and strong guided-exploration baselines that supply stepwise cues or partial solutions on average, improving the average accuracy by +4.72% (Qwen) and +3.23% (Llama) over the strongest baseline in our main comparison while needing only the gold answer. We further show that Cog-DRIFT improves pass@k at test time and that the curriculum improves sample efficiency. Cog-DRIFT outperforms GRPO when GRPO is given 8 times more rollouts or 4 times longer generations. Overall, our results highlight task reformulation and curriculum learning as an effective paradigm for overcoming the exploration barrier in LLM post-training.[1]

1 INTRODUCTION

Learning from problems that exceed a learner's current capacity is difficult, and this challenge is not unique to models. In human learning, the *Zone of Proximal Development* (VYGOTSKY, 1978) describes how learning is maximized when tasks are neither too easy nor too hard, but lie just beyond the learner's current capabilities and can be solved with scaffolding (Wood et al., 1976; Berk & Winsler, 1995). Educational psychology further shows that problem format shapes cognitive load. Structured formats such as multiple-choice questions (MCQs) (Kelly, 1916; Cantor et al., 2014) and fill-in-the-blank (cloze) tasks (Taylor, 1953; Polat, 2020) constrain the effective search space and turn generation into recognition, which reduces cognitive load, scaffolds learning, and facilitates transfer to more complex, open-ended problems (Sweller, 1988; Ko, 2010; Howard et al., 2017; van Gog et al., 2004; Sweller et al., 1998). These insights suggest an analogy for large language models (LLMs). When a problem is too difficult to learn from directly, reformulating it into structured variants might bring it into the model's Zone of Proximal Development, where learning signals become accessible.

This perspective is particularly relevant for reinforcement learning with verifiable rewards (RLVR). While RLVR has become a dominant paradigm for improving LLM reasoning (Guo et al., 2025), a key limitation remains in commonly used algorithms such as PPO (Schulman et al., 2017) and GRPO

[1] Code is available in the supplementary materials.

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.

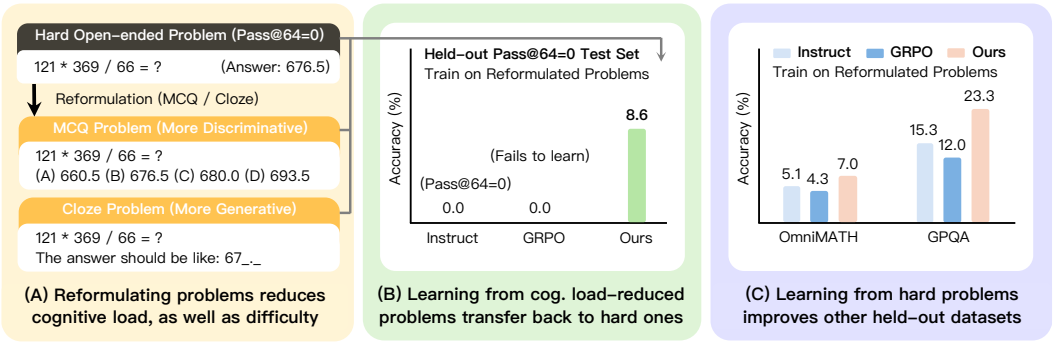

Figure 1: (A) If a problem is too hard (e.g., pass@64=0), the model cannot learn from it. Reformulation into an MCQ or a cloze task can effectively reduce both the cognitive load and the difficulty for the model. (B) We find that learning from cognitive-load-reduced tasks transfers back to the original hard questions (i.e., learning from MCQ/cloze also improves open-ended questions). (C) These improvements on the original hard questions also generalize to held-out datasets. Results in (B) and (C) are based on Llama3.2-3B-Instruct.

(Shao et al., 2024), which we visualize in Figure 1(A). When models are trained directly on difficult problems, on-policy RL often fails to obtain any positive reward signal, leading to zero advantages and no meaningful updates. Here, we define "difficult" as pass@64=0, following past work (Jiang et al., 2026). Prior work addresses this either by leveraging stronger models for expert trajectories (Qu et al., 2026), or by using the same policy to generate easier problems or hints (Sundaram et al., 2026; Chen et al., 2026a; Li et al., 2026; Wang et al., 2025). However, stronger models are not always available, and hard problems often sit at the ceiling of capacity, so an expert solution may not exist. Self-generated hints avoid this, but are bounded by the model's own capabilities.

We instead propose task reformulation as a practical solution. By rewriting open-ended questions (OEQ) as multiple-choice (MCQ) or fill-in-the-blank (cloze) tasks, the output space is dramatically constrained and learning signals become denser. Importantly, these reformulations span a spectrum between discriminative and generative learning tasks, from MCQs (ranking or selection) through cloze (producing missing components of the answer) to OEQs at the most generative extreme. Our approach leverages this spectrum by using discriminative tasks to bootstrap learning, since they are easier to learn (Geirhos et al., 2020), while the generative ones teach models to avoid shortcuts (Balepur et al., 2025; Chandak et al., 2025; Li et al., 2025). In Figure 1(B) and Section 3, we find that once the model learns from the reformulated tasks, it can transfer back to the original open-ended setting. However, a single reformulation leads to limited improvements on held-out datasets, whereas a mixture of multiple reformulations yields clear gains, suggesting that diversity in difficulty is key. This motivates Cog-DRIFT, a framework that reformulates hard problems into a spectrum of variable difficulty and organizes them into an adaptive curriculum analogous to cognitive scaffolding in human education.[2] Concretely, we generate a family of variants per question (e.g., multiple-choice with varying numbers of options, cloze-style questions), all of which preserve the original answer and thus remain verifiable. Cog-DRIFT then advances each instance from easier to harder variants based on empirical accuracy, moving to the next level once a target accuracy threshold is reached, thereby automating curriculum construction.

We evaluate Cog-DRIFT on six benchmarks, namely BigMath-Hard (Albalak et al., 2025), OmniMATH-Hard (Gao et al., 2025), AIME24, AIME25 (Art of Problem Solving, 2025), GPQA-Diamond (Rein et al., 2024), and Date Understanding (Srivastava et al., 2023; Suzgun et al., 2023), with two models: Qwen3-4B-Instruct-2507 (Yang et al., 2025) and Llama3.2-3B-Instruct (Grattafiori et al., 2024). Directly training on the hard problems with GRPO is unstable under sparse rewards and can harm generalization on held-out tasks. In contrast, Cog-DRIFT achieves the best average accuracy, with an absolute +4.72% average improvement for Qwen and +3.23% for Llama over the strongest baseline in Table 1. Cog-DRIFT achieves this while needing only the gold answer

[2]We use the term "cognitive" to draw an analogy to human learning, referring specifically to structured progression in task difficulty rather than implying human-like cognition in models.

to build its reformulations. Our method also improves pass@k over both the base model and the GRPO-trained variant, suggesting that it acquires new reasoning capabilities from examples initially beyond the model's capacity. Our analyses highlight two factors driving this: (1) a diverse mixture of reformulations is essential for transferring reasoning skills back to open-ended tasks, and (2) an instance-level curriculum boosts sample efficiency and prevents plateaus. We then isolate the source of the gain with four controls. (a) Extra compute does not substitute for reformulation, as Cog-DRIFT stays ahead even when GRPO is given 8 times more rollouts or 4 times longer generations. (b) Exposure to the reformulations is not sufficient either, since consuming them in a separate fine-tuning stage before RL leaves the average at the level of GRPO alone. (c) The gains do not come from answer leakage, since appending the gold answer reaches improved in-domain accuracy but collapses on the held-out test set due to overfitting. (d) The curriculum's order is crucial, as a reversed progression drops the average by 8.15%.

2 Related Work

Reinforcement Learning With Verifiable Rewards (RLVR). RLVR improves LLM reasoning through rule-based feedback on answer correctness, with GRPO providing a widely used training objective (Shao et al., 2024; Guo et al., 2025). Whether these gains reflect improved sampling of existing solutions or acquisition of new reasoning capabilities remains debated (Yue et al., 2025; Yuan et al., 2026). Closely related to our use of task formats, Lu et al. (2026) transform internet text into multiple-choice and masked prediction tasks to enable RLVR in domains without verifiable rewards. Our setting instead starts with verifiable but difficult reasoning problems and uses reformulation to make successful exploration more accessible, measuring transfer back to open-ended tasks.

Learning From Hard Problems. Guided-exploration methods supply reasoning signals when standard rollouts fail. NuRL uses self-generated abstract hints (Chen et al., 2026a), HINT supplies stepwise guidance (Wang et al., 2026), and QuestA and POPE condition exploration on partial solutions or offline prefixes (Li et al., 2026; Qu et al., 2026). Complementary approaches interleave supervised fine-tuning with RL (Ma et al., 2026) or combine warm-ups, replay, and curricula to learn from previously unsolved problems (Sun et al., 2026). Cog-DRIFT instead constructs answer-preserving reformulations of the same problem and advances from multiple-choice through cloze to open-ended formats as the model improves. We compare against guided exploration and SFT followed by RL to distinguish these mechanisms. We discuss related work further in Appendix A.

3 Problem Statement

Let $\mathcal{M}_\theta$ denote a language model with parameters θ. We consider a dataset of open-ended problems $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where x_i is a problem prompt and y_i is the corresponding exact answer.[3] The model can generate k independent samples $\{\hat{y}_i^{(j)}\}_{j=1}^k$ for a specific x_i, and the commonly used pass@k metric evaluates whether at least one of the k generations is correct. We define the set of *hard problems* as the subset of examples on which the model fails to produce a correct answer even after 64 samples: $\mathcal{D}_{\text{hard}} = \{(x_i, y_i) \in \mathcal{D} \mid \text{pass@64}(x_i) = 0\}$. This definition has an important implication for RL training with rollout-based objectives (e.g., GRPO): if pass@64$(x_i) = 0$, then the probability of sampling a correct trajectory is extremely low (requiring $\gg 64$ samples) given the same policy. Consequently, the reward signal is generally non-existent: all sampled trajectories receive zero reward, leading to zero variance in the advantage estimates. This in turn prevents meaningful gradient updates, as no trajectory is preferred over others (Qu et al., 2026; Yu et al., 2025). Empirically, we show in Section 6 that directly applying GRPO on $\mathcal{D}_{\text{hard}}$ yields little to no improvement because the model rarely explores the correct trajectory in an unconstrained, open-ended output space. Moreover, training on $\mathcal{D}_{\text{hard}}$ wastes computational effort performing rollouts on problems with no hope of being solved. To address this, we introduce a task reformulation function $\mathcal{T} : (x, y) \to (x', y)$ that converts an open-ended problem into a cognitively simplified format while preserving the underlying solution. A successful reformulation should satisfy three criteria. First, it must preserve the correct answer of the original problem. Second, it should make the problem empirically easier for the model. Third, learning from the reformulated problem should improve performance on the original problem.

[3] We use "open-ended" to denote problems with unconstrained output length, in contrast to multiple-choice or short-form tasks, while still assuming a well-defined correct answer.

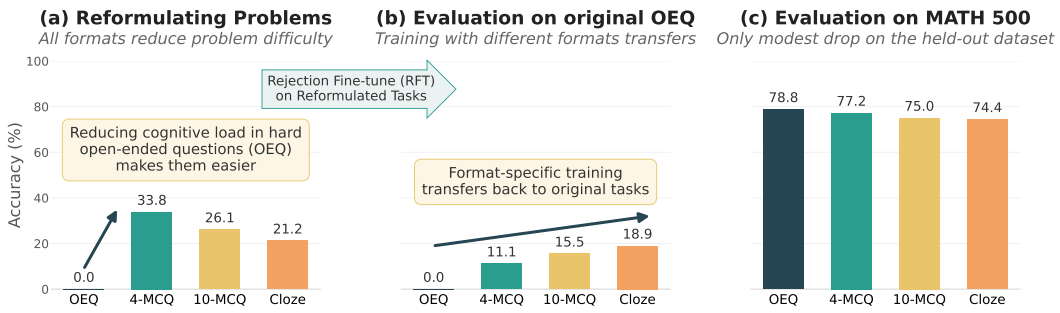

Figure 2: (a) Reformulating open-ended math problems into alternative formats consistently increases accuracy by easing structural constraints. (b) After rejection fine-tuning (RFT), these performance gains transfer back to the original open-ended problems used for training. (c) On unseen MATH500, the model trained with an easier format incurs only a modest drop.

We explore several instantiations of $\mathcal{T}$, utilizing an LLM (Qwen3-4B-Instruct-2507) to generate the reformulated dataset $\mathcal{D}'_{\text{hard}} = \{(x'_i, y_i) \mid (x_i, y_i) \in \mathcal{D}_{\text{hard}}\}$. We provide the prompt for reformulation in Appendix D.2. The formats we consider include: (1) **4-Choice:** The original question is combined with four answer options, where the gold answer appears in exactly one option. (2) **10-Choice:** Same as above, but we expand the number of options to ten. (3) **Cloze:** The original question is combined with a partially masked answer. For example, if the answer is 101, we append "The answer should look like: 1__". Example 1 is an example of an open-ended problem reformulated into a 4-choice format, and other examples for 10-choice and cloze can be found in Appendix D.1. Crucially, even after reformulation, the model must output the full answer value (e.g., 21) rather than just the option letter (e.g., B) for better transferability.

Example 1: An Example of $\mathcal{T}$ (Open-ended to Four Choice)

Question (x): What is the largest number of cells in a 9×9 board that can be cut along both diagonals without the board falling into pieces? **Answer (y): 21**

Reformulated Question (x'): What is the largest number of cells in a 9×9 board that can be cut along both diagonals without the board falling into pieces?
A. 18 B. 21 C. 24 D. 15 **Answer (y): 21**

3.1 Evaluating Difficulty Reduction

By definition, all of our chosen reformulations satisfy the first criterion by sharing the original gold answer. To evaluate the second criterion, we measure zero-shot performance on the reformulated dataset $\mathcal{D}'_{\text{hard}}$, as shown in Figure 2(a). Because we exclusively sampled problems where pass@64 = 0, the baseline performance on the original open-ended questions is strictly 0.0. However, upon applying format changes, performance immediately surpasses the random guessing baseline. For instance, the 4-choice MCQ reformulation achieves 33.8% accuracy, well above the 25% random baseline. Similar trends hold for the 10-choice and cloze formats. We also observe that task reformulations induce a consistent ordering of empirical difficulty: $\text{MCQ}_4 \leq \text{MCQ}_{10} \leq \text{Cloze} \leq \text{Open-ended}$.

3.2 Transferability and Generalization

Having established that these formats reduce difficulty, we evaluate the third criterion by testing whether learning from the converted problems transfers back to the original open-ended ones. To test this, we use $\mathcal{D}'_{\text{hard}}$ for rejection fine-tuning (RFT) (Yuan et al., 2023), as depicted in Figure 2(b). Specifically, we use $\mathcal{D}'_{\text{hard}}$ to elicit correct chain-of-thought (Wei et al., 2022) reasoning trajectories, perform supervised fine-tuning (SFT) on the questions x' paired with their correct trajectories, and then evaluate the trained model on the original open-ended dataset $\mathcal{D}_{\text{hard}}$. While acquiring correct reasoning chains directly from $\mathcal{D}_{\text{hard}}$ is not possible, the reduced difficulty of $\mathcal{D}'_{\text{hard}}$ allows us to extract valid trajectories. The results show that models trained on $\mathcal{D}'_{\text{hard}}$ transfer their learning back to

$\mathcal{D}_{\text{hard}}$. Interestingly, in Figure 2(b), we also observe that the more generative the task, the better the transfer back to the open-ended problem. For example, on the original $\mathcal{D}_{\text{hard}}$, the model trained with 4-choice achieves 11.1%, 10-choice achieves 15.5%, and cloze achieves 18.9% accuracy. To isolate genuine reasoning capabilities from memorization, we evaluate the format-specific trained models on the unseen MATH500 benchmark (Lightman et al., 2024; Hendrycks et al., 2021). As shown in Figure 2(c), models trained with a single reformulation incur only a modest drop on this held-out dataset, so the transfer in Figure 2(b) does not come at a severe cost to general reasoning. However, a single reformulation also does not improve held-out performance. Overall, the MCQ and cloze formats meet all three criteria, as they (1) preserve the original answer, (2) reduce empirical difficulty, and (3) transfer back to the original problems. This makes task reformulation a promising way to learn from hard problems, and we select these formats for our main experiments. Since a single reformulation is not sufficient for held-out gains, Cog-DRIFT combines multiple reformulations in an adaptive curriculum, and we show in Section 6 that both the mixture and the curriculum are needed.

4 Our Proposed Method: Cognitive Demand Reduction via Incremental Format Transitions

We build on the observation from Section 3 that different task reformulations form an order of empirical difficulty: $\text{MCQ}_4 \leq \text{MCQ}_{10} \leq \text{Cloze} \leq$ Open-ended. We leverage this structure to design a curriculum that adaptively increases problem difficulty based on the model's competence. Specifically, given a hard problem $(x_i, y_i) \in \mathcal{D}_{\text{hard}}$, we construct a sequence of reformulated variants: $\{x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, x_i^{(4)}\}$, where each level corresponds to a format of increasing difficulty:

$$x_i^{(1)} : \text{4-choice MCQ}, \quad x_i^{(2)} : \text{10-choice MCQ}, \quad x_i^{(3)} : \text{Cloze}, \quad x_i^{(4)} : \text{Open-ended}.$$

All variants share the same ground-truth answer y_i and are generated via $\mathcal{T}$ introduced in Section 3. We check each reformulation with deterministic verification scripts. For MCQ, we verify that: (1) the gold answer appears among the options, and (2) it is uniquely correct (i.e., no other option corresponds to the gold answer). For cloze, we ensure that: (1) the masked span is non-trivial (i.e., not equivalent to the full gold answer), and (2) each revealed digit is consistent with the gold answer. For example, a gold answer of 101 and a mask of 1_0 would constitute a failure case.

Instance-level Curriculum. We associate each training example with a difficulty level $d_i \in \{1, 2, 3, 4\}$, initialized to the easiest level $d_i = 1$. During training, the model $\mathcal{M}_\theta$ generates m samples for the current format: $\{\hat{y}_i^{(j)}\}_{j=1}^m \sim \mathcal{M}_\theta(\cdot \mid x_i^{(d_i)})$. We estimate the empirical accuracy at level d_i as: $\hat{a}_i = \frac{1}{m}\sum_{j=1}^m \mathbb{I}[\hat{y}_i^{(j)} = y_i]$. If $\hat{a}_i \geq \tau$, we promote the example to the next difficulty level. We adopt $\tau = 0.5$ in our experiments following past work (Chen et al., 2026b; Foster et al., 2025). This creates a per-sample curriculum that adapts to the model's evolving capability, ensuring that training focuses on the most informative difficulty for each problem.

GRPO Training Objective. We optimize the model using Group Relative Policy Optimization (GRPO) with rule-based rewards (Guo et al., 2025; Shao et al., 2024). For each input $x_i^{(d_i)}$, we sample a group of m responses and compute rewards for each sample. Let $r_i^{(j)}$ denote the reward for $\hat{y}_i^{(j)}$. GRPO normalizes rewards within the group, and the objective is

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}_{i,j}\left[\tilde{r}_i^{(j)} \log \mathcal{M}_\theta(\hat{y}_i^{(j)} \mid x_i^{(d_i)})\right],$$

which encourages the model to assign higher probability to relatively better samples within each group. Our reward function combines correctness and format compliance: $r_i^{(j)} = r_{\text{correct}}(\hat{y}_i^{(j)}, y_i) + r_{\text{format}}(\hat{y}_i^{(j)})$, where $r_{\text{correct}} \in [0, 1]$ indicates whether the predicted answer matches the ground truth, and $r_{\text{format}} \in [0, 0.2]$ measures whether the output follows the required format, i.e., whether the final answer is found in `\boxed{}` (Guo et al., 2025).

5 Experimental Setup

Models. We evaluate Cog-DRIFT on two models: Llama3.2-3B-Instruct (Grattafiori et al., 2024) and Qwen3-4B-Instruct-2507 (Yang et al., 2025). We consider these two models because they

show different behaviors in post-training (Shao et al., 2026) and they allow us to test cross-model generalization. Hereafter, we refer to them as Llama and Qwen, respectively.

Dataset. For training, we use the BigMath dataset (Albalak et al., 2025) due to its diversity and its difficulty annotations. Specifically, it contains 251k samples, each annotated with its pass rate under Llama3-8B, which lets us start from a harder subset with lower pass rates. We take the hardest 15% of samples (37.5k) and let Qwen sample 64 reasoning chains for each, keeping only the samples for which *none of the 64 chains reaches the correct solution*. Since Qwen generally outperforms Llama, we use Qwen to obtain $\mathcal{D}_{\text{hard}}$ and use the same $\mathcal{D}_{\text{hard}}$ for Llama. This results in 8.9k samples. We notice that filtering by pass@64 alone introduces a selection bias toward problems that are actually unsolvable, such as incomplete questions or questions that refer to a missing figure (examples in Appendix E.7). To alleviate this, we keep problems for which GPT-5.4's majority answer over three runs agrees with the gold answer (OpenAI, 2026). GPT-5.4 is used only to check data quality and contributes no reasoning traces to training. This leads to our final filtered training data, consisting of 958 hard problems (filtering comparisons and the impact of selection bias are in Appendix C.2). For evaluation, we test Cog-DRIFT on six benchmarks, namely **(1) BigMath-Hard (BMH)**, **(2) OmniMATH-Hard**, **(3) AIME24**, **(4) AIME25**, **(5) GPQA-Diamond**, and **(6) Date Understanding**. We provide details in Appendix C.1.

Baselines. We mainly compare with baselines that do not need privileged information or require a stronger teacher model: (1) **Zero-shot:** We prompt the model to think step-by-step and provide the answer within \boxed{} (Kojima et al., 2022). (2) **Few-shot:** Besides prompting the model to think step-by-step, we include 8 in-context examples. (3) **Rejection fine-tuning (RFT):** We prompt the model with cloze-reformulated problems and use the correct trajectories to perform supervised fine-tuning (Yuan et al., 2023). We choose fill-in-the-blank here because it performs best (cf. Figure 2). (4) **GRPO**: We train on $\mathcal{D}_{\text{hard}}$ using outcome correctness as the reward (Guo et al., 2025). (5) **NuRL (Abstract)**: We use abstract cues generated from the answer explanations of $\mathcal{D}_{\text{hard}}$ to hint the model during GRPO (Chen et al., 2026a). (6) **POPE** (Qu et al., 2026): We use partial solution steps as prefixes to guide exploration into reachable reward regions. The original POPE setup draws these prefixes from a stronger frontier model. Here they are generated by the policy itself, which keeps the comparison within the same self-improvement setting. We further compare against two methods that make hard problems learnable by supplying reasoning guidance, namely HINT (Wang et al., 2026) and QuestA (Li et al., 2026). Cog-DRIFT outperforms both while needing only the gold answer, which we report in Appendix E.2.

Implementation Details. We evaluate all methods using pass@1. Results are averaged over 16 runs for AIME and 3 runs for the other datasets. We adopt verl (Sheng et al., 2025) as the backbone and use vLLM (Kwon et al., 2023) to speed up rollout generation and inference. To verify equivalence between predictions and references, we use Math-Verify.[4] Other details can be found in Appendix B.

6 Results and Analysis

Reformulation enables learning from the hard problems. In Table 1, zero-shot baselines achieve 0% performance on BigMath-Hard (BMH), as expected for a pass@64=0 dataset. Consistent with Section 3, RFT on a single cloze reformulation improves BMH performance, but these gains do not carry over to the held-out benchmarks. In principle, GRPO should not yield learning gains when trained on samples with pass@64 = 0. However, with Qwen, we observe a +6.37% improvement on BMH. This may be attributed to the spurious reward effect specific to Qwen (Shao et al., 2026), where learning can occur even under weak or uninformative reward signals. Another hypothesis is that occasional successful rollouts (arising from stochastic exploration) introduce sparse but amplified reward signals, causing the model to overfit to a narrow subset of problems without learning transferable reasoning strategies. In contrast, Llama shows no improvement under GRPO, with BMH performance remaining at 0% after training. Despite gains on Qwen, GRPO harms generalization, causing drops across held-out benchmarks. NuRL with abstract hints and POPE with partial solution prefixes provide a middle ground. Both yield modest BMH gains and occasional improvements on held-out datasets, likely due to leveraging in-domain, self-generated guidance. By contrast, Cog-DRIFT achieves the best average accuracy, improving BMH by an absolute +10.11% (Qwen) and +8.64% (Llama), with average gains across all six benchmarks of +4.72% and +3.23%, respectively.

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

Table 1: Performance comparison on six benchmarks. We **bold** the best and underline the second-best results. Cog-DRIFT achieves the strongest overall results, improving avg. accuracy by +4.72% (Qwen) and +3.23% (Llama) over the strongest baseline. Crucially, it unlocks gains on BMH (0 → 10.11% on Qwen / 8.64% on Llama), where all the problems are collected with pass@64=0.

	BMH	Omni-Hard	AIME24	AIME25	GPQA	Date	Avg.
Qwen3-4B-Instruct-2507							
Zero-shot	0.0	15.67	46.92	**41.71**	23.57	87.71	35.93
Few-shot	0.0	16.31	44.78	37.98	26.12	88.37	35.59
RFT (Yuan et al., 2023)	4.32	15.13	40.06	34.88	23.04	85.11	33.76
GRPO (Shao et al., 2024)	6.37	16.37	31.87	27.29	31.81	**89.53**	33.87
NuRL (Chen et al., 2026a)	7.73	16.03	44.81	36.44	24.72	87.48	36.20
POPE (Qu et al., 2026)	6.51	15.69	45.31	38.81	22.27	84.17	35.46
Cog-DRIFT	**10.11**	**18.58**	**51.74**	40.28	**36.01**	88.75	**40.92**
Llama3.2-3B-Instruct							
Zero-shot	0.0	5.09	3.75	0.42	15.32	54.87	13.24
Few-shot	0.0	5.74	3.33	0.33	17.89	55.63	13.82
RFT (Yuan et al., 2023)	2.47	6.52	1.67	0.00	16.13	50.31	12.85
GRPO (Shao et al., 2024)	0.0	4.32	0.68	0.42	12.03	38.85	9.38
NuRL (Chen et al., 2026a)	1.38	6.67	**4.17**	**1.11**	16.98	56.72	14.51
POPE (Qu et al., 2026)	1.04	**7.19**	3.33	0.42	15.79	54.91	13.78
Cog-DRIFT	**8.64**	7.04	**4.17**	0.42	**23.34**	**62.82**	**17.74**

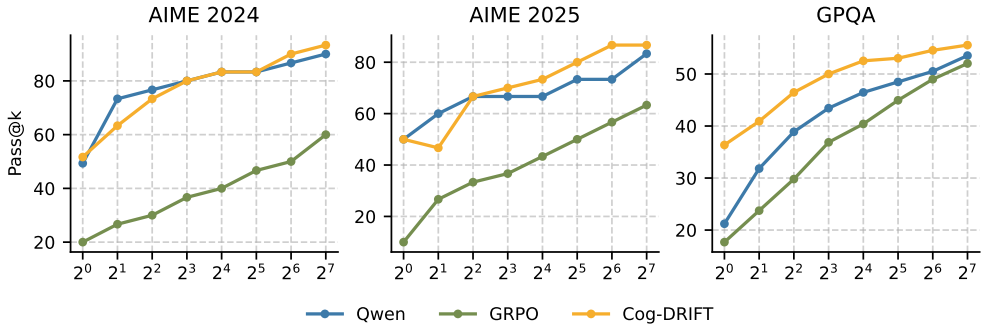

Figure 3: When training on hard open-ended problems and evaluating on AIME24, AIME25, and GPQA-Diamond using Qwen, Cog-DRIFT consistently achieves higher pass@k compared to both the base Qwen model and the GRPO-trained model, particularly as k increases.

Cog-DRIFT improves pass@k. In Table 1, we observe that Cog-DRIFT achieves the best average pass@1. Because our approach specifically targets problems with pass@64 = 0 (i.e., problems that are unsolvable under standard training and therefore contain knowledge inaccessible to the model), we hypothesize that successful learning from these examples should also manifest as improvements in pass@k for larger values of k, reflecting the acquisition of new knowledge. To test this hypothesis, we plot pass@k for $k \in \{1, 2, \ldots, 128\}$ in Figure 3 using Qwen. On AIME24, AIME25, and GPQA-Diamond, Cog-DRIFT consistently achieves higher pass@k as k increases, outperforming both the base instruct model and the GRPO-trained model. At $k = 128$, Cog-DRIFT improves over the base model by +3.33%, +3.33%, and +2.02% on AIME24, AIME25, and GPQA-Diamond, respectively. In contrast, GRPO again exhibits degraded performance, echoing the finding that directly training on hard problems fails to generalize. These results indicate that learning from reformulated problems reshapes the model's output distribution, making correct solutions more accessible under sampling. Overall, the gains in pass@k provide further evidence that, by training on reformulated hard problems, the model can effectively learn reasoning capabilities.

Table 2: Training with more rollouts raises in-distribution BMH accuracy but does not transfer, and Cog-DRIFT with $n = 8$ outperforms GRPO with $n = 64$. Results are with Qwen.

	BMH	Omni-Hard	AIME24	AIME25	GPQA	Date	Avg.
GRPO ($n = 8$)	6.37	16.37	31.87	27.29	31.81	**89.53**	33.87
GRPO ($n = 16$)	7.35	15.04	37.78	40.00	20.37	88.80	34.89
GRPO ($n = 32$)	7.78	17.04	51.11	36.67	20.37	85.28	36.38
GRPO ($n = 64$)	8.43	16.52	**55.56**	40.00	23.57	85.46	38.26
Cog-DRIFT ($n = 8$)	**10.11**	**18.58**	51.74	**40.28**	**36.01**	88.75	**40.92**

Scaling GRPO's rollout budget does not substitute for reformulation. Our notion of hardness is tied to a bounded sampling budget (pass@64 = 0), which raises the question of whether GRPO's failure is intrinsic to the policy or simply budget-driven. Although scaling GRPO's number of rollouts is costly, we vary the number of rollouts $n \in \{8, 16, 32, 64\}$ on the same training data. We find that more rollouts do help in-distribution, with BMH rising from 6.37 at $n = 8$ to 8.43 at $n = 64$, consistent with occasionally sampling a rare correct trajectory. However, these gains transfer poorly. GPQA-Diamond falls from 31.81 at $n = 8$ to 20.37 at $n = 16$ and $n = 32$, and although the average rises steadily with n, it reaches only 38.26 at $n = 64$, still below Cog-DRIFT with $n = 8$ (40.92). Cog-DRIFT therefore outperforms GRPO even when GRPO is given 8 times the rollout budget. Taken together, this suggests that brute-forcing rare successes and reducing cognitive load are not interchangeable mechanisms, and that the reward signal Cog-DRIFT creates differs from the one obtained by simply sampling more rollouts.

Increasing generation budgets does not close the gap. We observe that GRPO drops sharply on AIME24 and AIME25 in Table 1, so we ask whether test-time truncation contributes to this drop. All methods are trained with an 8k response limit for fairness, and we evaluate GRPO and Cog-DRIFT at 16k and 32k in Figure 4. A longer budget does recover much of GRPO's loss, by +20.35% on AIME24 (31.87 → 52.22) and +16.05% on AIME25 (27.29 → 43.34), so truncation is part of the story. However, the longer budget narrows the gap but does not close it. Cog-DRIFT improves from 40.92 to 42.30 and then 43.71, staying ahead by +3.11% at 16k and +3.44% at 32k. Cog-DRIFT evaluated at 8k also averages slightly higher than GRPO evaluated at 32k (40.92 against 40.27), at a quarter of the maximum generation length. We hypothesize that longer generations let GRPO finish truncated solutions to problems it already partially solves, but do not create signal for the pass@64 = 0 problems that motivate our setting.

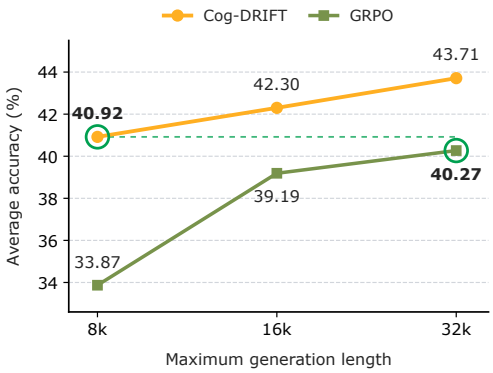

Figure 4: Cog-DRIFT at 8k outperforms GRPO at 32k (40.92% vs. 40.27%), highlighted in green. Results are six-benchmark averages with Qwen; training uses an 8k limit. Full results in Table 6.

Reformulation diversity and easy-to-hard ordering are both necessary. We separately ablate the two ingredients of Cog-DRIFT. First, we remove the curriculum and fine-tune the models with RFT on fixed mixtures of problem types (Appendix E.5). For open-ended questions (OEQ), we obtain trajectories through post-hoc rationalization, prompting the model with the gold answer and keeping the rationales that reach it (Zelikman et al., 2022). Training with a single reformulation does not improve the held-out sets (as we observed in Figure 2(c)), whereas a mixture of reformulations yields clear gains. Specifically, 4-MCQ alone lowers OmniMATH-Hard (15.67 → 14.82) and Cloze alone lowers MATH500 (78.80 → 74.40), whereas *4-MCQ + Cloze* raises both, to 18.36 and 82.60. Training solely on *4-MCQ + Cloze* without any open-ended data is already comparable to the mixture that also includes open-ended questions (18.42 and 82.20). Diversity in problem representation is an important factor that drives transfer. We further ablate the three design choices the curriculum involves, namely the promotion threshold, the format ordering, and the promotion granularity (Appendix E.6).

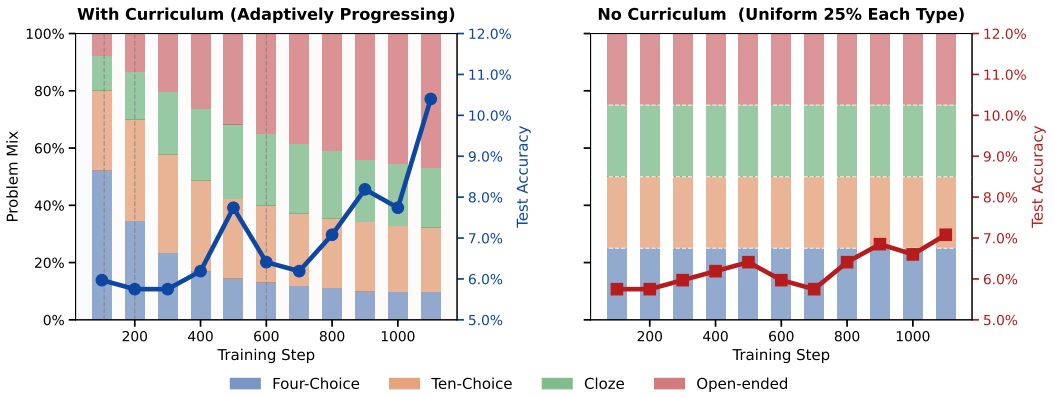

Figure 5: **Left:** Instance-level curriculum adaptively reallocates samples from easier (MCQ) to harder (OEQ) reformulations based on per-instance accuracy, leading to improved sample efficiency and continued performance gains. **Right:** With a static uniform mixture (always 25% for each format), improvement stagnates. Test accuracy is reported on open-ended questions from OmniMATH-Hard.

Performance is insensitive to the threshold, as every $\tau \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ still outperforms the strongest baseline (averages from 39.23 to 42.49). We therefore follow prior work and use $\tau = 0.5$ for simplicity (Chen et al., 2026b; Foster et al., 2025). Per-instance promotion also slightly outperforms a per-batch schedule (40.92 against 40.55). Moreover, we find that ordering matters most. Reversing the easy-to-hard progression lowers the average by 8.15% (to 32.77), below GRPO on the original problems (33.87), and a random permutation lowers it by 5.31% (to 35.61). We hypothesize that a hard-to-easy ordering starts training on open-ended problems with pass@64 = 0, so the model receives almost no reward early on, much as in GRPO.

Instance-level curriculum improves sample efficiency. Training on a mixture of multiple reformulations is already effective for learning from hard problems (Table 9). However, each reformulation increases the dataset size linearly; for example, combining OEQ, 4-choice MCQ, and Cloze triples the number of training samples. Since these reformulations largely share the same underlying information (differing primarily in structure and cognitive load), we hypothesize that a more adaptive curriculum can improve sample efficiency. In Figure 5, we present both the proportion of each problem type during training and the corresponding test performance on OmniMATH-Hard, comparing settings with and without our instance-level curriculum. We refer to this approach as instance-level because progression is determined per instance: once an instance achieves an average accuracy of at least 50%, it is automatically advanced to the next difficulty level. As a result, the proportion of 4-choice MCQ (the easiest format) steadily decreases over training, while intermediate formats (10-choice MCQ and Cloze) first increase and then decline, and open-ended questions grow steadily as the final stage of the curriculum. In contrast, without the curriculum, the proportion of each problem type remains uniform throughout training (25% each). Importantly, the curriculum leads to continued improvements in test performance on OmniMATH-Hard, whereas performance without the curriculum tends to plateau.

7 Conclusion

We study how to enable language models to learn from problems that lie beyond their current capabilities. Drawing on theories of human learning, we use task reformulation to lower the difficulty of hard problems, analogous to reducing cognitive load for a human learner. We propose Cog-DRIFT, a framework that reformulates hard open-ended questions into multiple-choice and cloze variants that preserve the correct answer, and organizes these variants into an instance-level curriculum. Across two models and six benchmarks, Cog-DRIFT achieves the best average performance and improves pass@k at larger k on AIME24, AIME25, and GPQA-Diamond. Our analyses suggest that both the diversity of reformulations and the easy-to-hard ordering contribute to these gains, and that the instance-level curriculum keeps improving over training while a static mixture plateaus. Overall, our results suggest task reformulation as a simple way to obtain learning signal from reasoning problems that a model cannot solve within a fixed sampling budget.

AI USE STATEMENT

We used Claude Code as a writing assistant to polish the manuscript and check its grammar. We also used it as a coding assistant to implement our pipeline from our own design and to check the correctness of the code. We did not use LLMs for research ideation. The authors reviewed all LLM-assisted text and code and take full responsibility for the content of this paper.

ETHICS STATEMENT

This work studies how to train LLMs on reasoning problems that they cannot yet solve. All training and evaluation data come from publicly available benchmarks (BigMath, OmniMATH, AIME, MATH500, GPQA-Diamond, and Date Understanding), which consist of math, science, and date-reasoning questions in English. We use each dataset under its license and in a way consistent with its intended use (see Table 4). The reformulated variants that Cog-DRIFT generates (multiple-choice and cloze questions) preserve the original question content and gold answer, so the risk of introducing new sensitive content is minimal. Cog-DRIFT improves general reasoning ability, and like other post-training methods, it could be applied to models that are later misused. We do not expect Cog-DRIFT to add risks beyond those of standard RLVR, since it trains only on verifiable reasoning problems.

REPRODUCIBILITY STATEMENT

We release our code in the supplementary materials. Both models we train (Llama3.2-3B-Instruct and Qwen3-4B-Instruct-2507) are open-weight, and we build on the open-source frameworks verl, vLLM, and Math-Verify. Section 4 describes Cog-DRIFT, including the deterministic checks we apply to each reformulated variant and the promotion threshold of the curriculum. Appendix B lists the hardware and all training and inference hyperparameters (Table 3). Appendix C describes how we construct the hard training set $\mathcal{D}_{\text{hard}}$ and reports the sample size and license of every dataset. Appendix D.2 gives the full prompts we use to generate each reformulated format. To reduce variance in evaluation, we report pass@1 averaged over 16 runs on AIME and 3 runs on the other benchmarks.

REFERENCES

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models, 2025. URL https://arxiv.org/abs/2502.17387.

Art of Problem Solving. Aime problems and solutions, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, Jeongyeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 700–719, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.30. URL https://aclanthology.org/2026.eacl-long.30/.

Nishant Balepur, Rachel Rudinger, and Jordan Boyd-Graber. Which of these best describes multiple choice evaluation with LLMs? a) forced B) flawed C) fixable D) all of the above. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3394–3418, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.169. URL https://aclanthology.org/2025.acl-long.169/.

Laura E Berk and Adam Winsler. *Scaffolding Children's Learning: Vygotsky and Early Childhood Education. NAEYC Research into Practice Series. Volume 7.* ERIC, 1995. URL https://eric.ed.gov/?id=ED384443.

Allison D. Cantor, Andrea N. Eslick, Elizabeth J. Marsh, Robert A. Bjork, and Elizabeth Ligon Bjork. Multiple-choice tests stabilize access to marginal knowledge. *Memory & Cognition*, 43(2):193–205, 2014. ISSN 1532-5946. doi: 10.3758/s13421-014-0462-6. URL http://dx.doi.org/10.3758/s13421-014-0462-6.

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. Answer matching outperforms multiple choice for language model evaluation, 2025. URL https://arxiv.org/abs/2507.02856.

Justin Chen, XIANGYU PENG, Prafulla Kumar Choubey, Kung-Hsiang Huang, Jiaxin Zhang, Mohit Bansal, and Chien-Sheng Wu. Nudging the boundaries of llm reasoning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 30315–30335, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/336229145e702cfdd4f26cf16438e7b8-Paper-Conference.pdf.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for LLM reasoning. In *Third Conference on Language Modeling*, 2026b. URL https://openreview.net/forum?id=WBSf3T7vTt.

Thomas Foster, Anya Sims, Johannes Forkel, and Jakob Foerster. Lilo: Learning to reason at the frontier of learnability. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 48941–48974. Curran Associates, Inc., 2025. doi: 10.52202/085713-1634. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/45d5823950cab9a88e10c6e8dbff351c-Paper-Conference.pdf.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yaqPf0KAlN.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL http://dx.doi.org/10.1038/s42256-020-00257-z.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri,

Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang,

Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

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.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25548–25560, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1298. URL https://aclanthology.org/2025.emnlp-main.1298/.

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. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Steven J. Howard, Hana Burianová, Alysha Calleia, Samuel Fynes-Clinton, Lisa Kervin, and Sahar Bokosmaty. The method of educational assessment affects children's neural processing and performance: behavioural and fmri evidence. *npj Science of Learning*, 2(1), August 2017. ISSN

2056-7936. doi: 10.1038/s41539-017-0010-9. URL http://dx.doi.org/10.1038/s41539-017-0010-9.

Jiedong Jiang, Wanyi He, Wang Yuefeng, Guoxiong Gao, Yongle Hu, Jingting Wang, Nailin Guan, Peihao Wu, Bryan Dai, Liang Xiao, and Bin Dong. FATE: A formal benchmark series for frontier algebra of multiple difficulty levels. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3bD19r4jqh.

F. J. Kelly. The kansas silent reading tests. *Journal of Educational Psychology*, 7(2):63–80, February 1916. ISSN 0022-0663. doi: 10.1037/h0073542. URL http://dx.doi.org/10.1037/h0073542.

Myong Hee Ko. A comparison of reading comprehension tests: Multiple-choice vs. open-ended. *English Teaching*, 65(1):137–159, March 2010. ISSN 1017-7108. doi: 10.15858/engtea.65.1.201003.137. URL http://dx.doi.org/10.15858/engtea.65.1.201003.137.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213. Curran Associates, Inc., 2022. doi: 10.52202/068431-1613. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626. ACM, October 2023. doi: 10.1145/3600006.3613165. URL http://dx.doi.org/10.1145/3600006.3613165.

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

Alexander Cong Li, Ananya Kumar, and Deepak Pathak. Generative classifiers avoid shortcut solutions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=oCUYc7BzXQ.

Jiazheng Li, Hongzhou Lin, Hong Lu, Kaiyue Wen, Zaiwen Yang, Jiaxuan Gao, Yi Wu, and Jingzhao Zhang. Questa: Expanding reasoning capacity in llms via question augmentation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 144191–144215, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e92d5414d8cc659e0011d0b559467410-Paper-Conference.pdf.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Arpo:end-to-end policy optimization for gui agents with experience replay, 2025. URL https://arxiv.org/abs/2505.16282.

Ximing Lu, David Acuna, Jaehun Jung, Jian Hu, Di Zhang, Shizhe Diao, Yunheng Zou, Shaokun Zhang, Brandon Cui, Mingjie Liu, Hyunwoo Kim, Prithviraj Ammanabrolu, Jan Kautz, Yi Dong, and Yejin Choi. Golden goose: A simple trick to synthesize unlimited RLVR tasks from unverifiable internet text. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=LkiD08kdRy.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, Wentao Zhang, and Bin CUI. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LzCBLrNoyM.

Vaskar Nath, Elaine Lau, Anisha Gunjal, Manasi Sharma, Nikhil Barhate, and Sean M. Hendryx. Adaptive guidance accelerates reinforcement learning of reasoning models. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025. URL https://openreview.net/forum?id=IvFyqtDiti.

OpenAI. Introducing gpt-5.4, 2026. URL https://openai.com/index/introducing-gpt-5-4/.

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum reinforcement learning from easy to hard tasks improves LLM reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KJvHnl3kUv.

Murat Polat. Analysis of multiple-choice versus open-ended questions in language tests according to different cognitive domain levels. *Novitas-ROYAL (Research on Youth and Language)*, 14(2): 76–96, 2020. URL https://eric.ed.gov/?id=EJ1272114.

Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. Pope: Learning to reason on hard problems via privileged on-policy exploration, 2026. URL https://arxiv.org/abs/2601.18779.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in RLVR. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tqTNOpkP5j.

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.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7HNRYT4V44.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL http://dx.doi.org/10.1145/3689031.3696075.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=UEhpyq41b9.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew Kyle Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cesar Ferri, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Christopher Waites, Christian Voigt, Christopher D Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, C. Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germàn Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-Lopez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Francis Anthony Shevlin, Hinrich Schuetze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh Dhole, Kevin Gimpel, Kevin Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros-Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje Ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramirez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael Andrew Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Andrew Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras,

Rosanne Liu, Rowan Jacobs, Rui Zhang, Russ Salakhutdinov, Ryan Andrew Chi, Seungjae Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel Stern Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Shammie Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven Piantadosi, Stuart Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, vinay uday prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Sophie Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=uyTL5Bvosj. Featured Certification.

Yiyou Sun, Yuhan Cao, Pohao Huang, Haoyue Bai, Hanna Hajishirzi, Nouha Dziri, and Dawn Song. Rl grokking recipe: How does rl unlock and transfer new algorithms in llms? In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 93190–93220, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/96842011407c2691ab4eefff48fc864d-Paper-Conference.pdf.

Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, and Julia Kempe. Teaching models to teach themselves: Reasoning at the edge of learnability, 2026. URL https://arxiv.org/abs/2601.18778.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

John Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285, April 1988. ISSN 1551-6709. doi: 10.1207/s15516709cog1202_4. URL http://dx.doi.org/10.1207/s15516709cog1202_4.

John Sweller, Jeroen J. G. van Merrienboer, and Fred G. W. C. Paas. Cognitive architecture and instructional design. *Educational Psychology Review*, 10(3):251–296, 1998. ISSN 1573-336X. doi: 10.1023/a:1022193728205. URL http://dx.doi.org/10.1023/A:1022193728205.

Wilson L. Taylor. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433, 1953. ISSN 0022-5533. doi: 10.1177/107769905303000401. URL http://dx.doi.org/10.1177/107769905303000401.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong,

Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

Tamara van Gog, Fred Paas, and Jeroen J.G. van Merriënboer. Process-oriented worked examples: Improving transfer performance through enhanced understanding. *Instructional Science*, 32(1-2): 83–98, January 2004. ISSN 1573-1952. doi: 10.1023/b:truc.0000021810.70784.b0. URL http://dx.doi.org/10.1023/B:TRUC.0000021810.70784.b0.

L. S. VYGOTSKY. Mind in society: Development of higher psychological processes. *Harvard University Press*, 1978.

Jiayu Wang, Yifei Ming, Zixuan Ke, Caiming Xiong, Shafiq Joty, Aws Albarghouthi, and Frederic Sala. Beyond accuracy: Dissecting mathematical reasoning for llms under reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 66616–66653. Curran Associates, Inc., 2025. doi: 10.52202/085713-2235. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6058d0c628a03fd95dfe5c72cbdf9e64-Paper-Conference.pdf.

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Ma Shuguang, Fei Yu, and Yanghua Xiao. Don't tell the answer, truly guide the reasoning during RL rollouts. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3437–3455, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.169. URL https://aclanthology.org/2026.findings-acl.169/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

David Wood, Jerome S. Bruner, and Gail Ross. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100, April 1976. ISSN 1469-7610. doi: 10.1111/j.1469-7610.1976.tb00381.x. URL http://dx.doi.org/10.1111/j.1469-7610.1976.tb00381.x.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may or may not escape its origin, 2026. URL https://arxiv.org/abs/2507.14843.

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 D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 117157–117186. Curran Associates, Inc., 2025. doi: 10.52202/085713-3904. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a9d5c33e38442450f9781bd7bad3c81e-Paper-Conference.pdf.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui

Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, juncai liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

Zhiqi Yu, Zhangquan Chen, Mengting Liu, Heye Zhang, and Liangqiong Qu. Unveiling implicit advantage symmetry: Why grpo struggles with exploration and difficulty adaptation, 2026. URL https://arxiv.org/abs/2602.05548.

Lifan Yuan, Weize Chen, Yuchen Zhang, Ganqu Cui, Hanbin Wang, Ziming You, Ning Ding, Zhiyuan Liu, Maosong Sun, and Hao Peng. From f(x) and g(x) to f(g(x)): LLMs learn new skills in RL by composing old ones. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jt7oCtYqHE.

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? In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 57654–57689. Curran Associates, Inc., 2025. doi: 10.52202/085713-1933. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/537d5aa768c2d534016a4d06f87bc8fb-Paper-Conference.pdf.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488. Curran Associates, Inc., 2022. doi: 10.52202/068431-1126. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/639a9a172c044fbb64175b5fad42e9a5-Paper-Conference.pdf.

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. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=vSMCBUgrQj.

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

Xiaoying Zhang, Yipeng Zhang, Hao Sun, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. Critique-grpo: Advancing llm reasoning with natural language and numerical feedback, 2026. URL https://arxiv.org/abs/2506.03106.

Rosie Zhao, Alexandru Meterez, Sham M. Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: RL post-training amplifies behaviors learned in pretraining. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=dp4KWuSDzj.

APPENDIX

A EXPANDED RELATED WORK

This section expands the discussion in Section 2, covering broader perspectives on RLVR and approaches to learning from hard problems.

Reinforcement Learning With Verifiable Rewards (RLVR). RLVR has emerged as a dominant paradigm for training large language models on reasoning tasks by utilizing objective, rule-based feedback rather than human preferences (Lambert et al., 2025; Guo et al., 2025; Team et al., 2025; Zeng et al., 2025). By employing deterministic verifiers (e.g., binary rewards based on final answer correctness), RLVR is particularly effective for mathematics and code generation where ground truth is unambiguous. Extending this paradigm to unverifiable domains, Lu et al. (2026) demonstrate that large-scale internet text can be transformed into structured formats such as multiple-choice and masked prediction tasks to enable RLVR. Central to the current discourse is whether RLVR enables the genuine discovery of new reasoning abilities or primarily performs "distribution sharpening" by amplifying high-reward paths already present in the model's latent space (Zhang et al., 2025; Zhao et al., 2025; Shenfeld et al., 2026). While some studies suggest RLVR mainly improves sampling efficiency (pass@1) without increasing the total knowledge (pass@k) (Yue et al., 2025; He et al., 2025), others argue that RL can indeed foster new skills through the strategic combination of existing ones (Yuan et al., 2026) or by balancing the entropy-reward trade-off to prevent premature convergence (Wu et al., 2026). Although with a sufficiently large k (e.g., vocab_size$^{\text{seq_len}}$), even a randomly initialized model could achieve near-perfect pass@k, RLVR operates under a rollout budget, so the real question is how to improve the model within a limited sampling budget, where k must remain small relative to the total number of rollouts (Yu et al., 2026; Nath et al., 2025).

Learning From Hard Problems. A challenge in RLVR arises when problems are so difficult that on-policy sampling fails to discover any correct solutions, i.e., the model cannot get any non-zero reward (Qu et al., 2026). To bridge this gap, research has shifted toward sophisticated exploration strategies and hybrid optimization. One line of work focuses on "nudging" the model through self-generated hints, critiques, or partial solutions to reduce problem difficulty and expand the reasoning boundary (Li et al., 2026; Zhang et al., 2026; Chen et al., 2026a; Wang et al., 2025). Ma et al. (2026) interleave SFT to learn from hard questions that RLVR cannot. Sundaram et al. (2026) train a teacher policy to generate question-answer pairs based on the improvement of the student policy. Other approaches leverage privileged information or offline trajectories to provide a learning signal in sparse-reward environments (Lu et al., 2025; Yan et al., 2025), such as POPE, which uses offline prefixes from a stronger model to guide exploration into reachable reward regions (Qu et al., 2026). In parallel, Sun et al. (2026) propose a "grokking" recipe for RLVR by showing that staged warm-ups, experience replay, and a curriculum enable learnability on previously unsolvable problems. Together, these advancements suggest that scaling RLVR requires moving beyond pure on-policy exploration toward self-improvement mechanisms and careful curriculum design (Shi et al., 2026; Parashar et al., 2026; Bae et al., 2026). Despite their differences, these approaches share a common principle: they aim to make hard problems learnable by injecting additional signal through stronger supervision, guided exploration, or richer training dynamics. Our approach takes a fundamentally different perspective. We introduce cognitive load as a unifying lens and show that reformulating problems into simpler variants enables the model to access reward signals that are otherwise unreachable under standard RLVR. Crucially, this is coupled with an adaptive curriculum that dynamically schedules these reformulations based on the model's learning progress, ensuring that training remains within a learnable regime while gradually increasing difficulty. Empirically, this leads to a substantial increase in the fraction of hard training problems that become solvable, reaching up to 18.9%, compared to the 1-4% range reported by prior methods (Chen et al., 2026a; Sundaram et al., 2026).

B Implementation Details

All methods are run on 4 A6000 GPUs. During training, we use a temperature of 1.0 and set the clip-high parameter ϵ to 0.28 (Yu et al., 2025). The rollout number is 8 for every GRPO-based method. We cap the output length at 8192 tokens for both training and testing. At inference, we fix the temperature to 0.7. Token limits and inference temperature are aligned across all baselines for evaluation. We report the detailed hyperparameters in Table 3.

Table 3: Configurations and hyperparameters we use for training.

	Llama	**Qwen**
num_train_samples	958	958
max_prompt_length	1024	1024
max_response_length	8192	8192
lr	1e-6	1e-6
clip_ratio_low	0.2	0.2
clip_ratio_high	0.28	0.28
rollout_temperature	1	1
rollout_n	8	8
kl_coef	0.01	0.01
train_batch_size	8	8

C Datasets

We document the data used in this work here. Appendix C.1 lists the evaluation benchmarks with their sample sizes and licenses. Appendix E.7 and Appendix C.2 then concern the construction of the hard training set $\mathcal{D}_{\text{hard}}$, showing examples of flawed problems that pass@k filtering retains, and quantifying the resulting selection bias across filtering strategies.

C.1 Dataset Statistics and Licenses

Cog-DRIFT is tested across a diverse set of reasoning-intensive tasks spanning STEM and other domain-specific areas, namely **(1) BigMath-Hard (BMH)**, a pass@64=0 test split with 192 samples from BigMath (Albalak et al., 2025); **(2) OmniMATH-Hard**, the 452 hardest (difficulty ≥ 7.5) Olympiad-level math problems from OmniMATH (Gao et al., 2025); **(3) AIME24** and **(4) AIME25**, the 30 problems from each of the 2024 and 2025 AIME competitions (Art of Problem Solving, 2025); **(5) GPQA-Diamond**, 198 PhD-level questions covering biology, physics, and chemistry (Rein et al., 2024); and **(6) Date Understanding**, 369 date and time arithmetic tasks from BIG-bench (Srivastava et al., 2023; Suzgun et al., 2023). Some of the analyses are also run on **MATH500**, a subset of 500 problems from the MATH benchmark (Hendrycks et al., 2021) curated by Lightman et al. (2024). Note that GPQA-Diamond and Date Understanding consist of multiple-choice questions, but we evaluate the model with answer matching (Chandak et al., 2025), i.e., the generated answer should match the ground-truth answer, instead of just comparing the option letter. This mitigates the tendency to generate a shortcut answer or memorize the option letter. For Date Understanding, whose questions are easier, we remove the options to make it fully open-ended. We provide the sample sizes and licenses of the datasets used in this work in Table 4. All datasets are in English and are used in a fashion consistent with their intended use.

Table 4: The statistics and licenses of the datasets used in this study. All datasets are used for evaluation, except for the BigMath train split.

	Sample Size	License
BigMath (Train) (Albalak et al., 2025)	958	Apache License
BigMath (Test) (Albalak et al., 2025)	192	Apache License
OmniMATH (Gao et al., 2025)	452	Apache License
AIME24 (Art of Problem Solving, 2025)	30	CC0
AIME25 (Art of Problem Solving, 2025)	30	CC0
MATH500 (Lightman et al., 2024; Hendrycks et al., 2021)	500	MIT License
GPQA-Diamond (Rein et al., 2024)	198	MIT License
Date Understanding (Srivastava et al., 2023; Suzgun et al., 2023)	369	Apache License

C.2 Estimating Problem Difficulty With Pass@k Can Introduce Selection Bias

Recall that our training set retains problems with pass@64 = 0 and applies the Silver Answer Filter to remove flawed question-answer pairs. Here, we compare filtering strategies and their effects on downstream performance.

While using pass@k is a common way to estimate problem difficulty (Yuan et al., 2026; Jiang et al., 2026; Yu et al., 2025), it naturally introduces selection bias, since it also retains problems that are "truly unsolvable" rather than merely beyond the model's current capability. In particular, we observe that our initial dataset $\mathcal{D}_{\text{hard}}$ contains a substantial number of flawed samples, including incomplete or unanswerable questions, as well as instances with incorrect ground-truth labels. To quantify the impact of data quality, we evaluate two filtering strategies: (1) *Binary Filter*, where GPT-5.4 (OpenAI, 2026) is prompted to determine whether a question is complete and answerable, and only samples labeled as solvable are retained, without requiring the model to produce a solution; and (2) *Silver Answer Filter*, where GPT-5.4 generates three independent solutions and we retain only those samples whose ground-truth answer agrees with the majority vote. The resulting dataset statistics and corresponding performance are summarized in Table 5. The Silver Answer Filter is highly selective, removing a large fraction of the data, whereas the Binary Filter yields a more moderate reduction. Despite these differences, Qwen is less sensitive to data quality. Its performance on MATH500 remains largely unchanged across filtering strategies, with only minor variation on GPQA-Diamond. We suspect that this is related to Qwen's known ability to improve even with random rewards (Shao et al., 2026). In contrast, Llama is more sensitive to dataset quality, showing a wider performance range across settings and a consistent trend of improved results with cleaner data. Therefore, we apply the Silver Answer Filter by default. Notably, GPT-5.4 is used solely to approximate the filtering of noisy question-answer pairs; we do not use its reasoning traces for any form of distillation.

Table 5: Impact of data quality filtering on model performance. While the Silver Answer Filter is more selective and substantially reduces dataset size, it consistently improves performance for Llama and slightly for Qwen on GPQA-Diamond. Overall, Llama is more sensitive to data quality, whereas Qwen is less affected by variations in data quality.

	# Data	Qwen-MATH	Qwen-GPQA	Llama-MATH	Llama-GPQA
Original D_{hard}	8922	82.2	33.1	24.2	20.7
Binary Filter	5922	81.8	34.7	28.0	22.8
Silver Ans Filter	958	81.6	36.0	32.2	23.3

D Problem Reformulation

D.1 Examples of Different Problem Formats

Example 2: Open-ended Problem

Question (x): What is the largest number of cells in a 9×9 board that can be cut along both diagonals without the board falling into pieces? **Answer (y): 21**

Example 3: Four Choice Problem

Reformulated Question (x'): What is the largest number of cells in a 9×9 board that can be cut along both diagonals without the board falling into pieces? A. 18 B. 21 C. 24 D. 15 **Answer (y): 21**

Example 4: Ten Choice Problem

Reformulated Question (x'): What is the largest number of cells in a 9×9 board that can be cut along both diagonals without the board falling into pieces? A. 4 B. 6 C. 7 D. 9 E. 12 F. 15 G. 18 H. 21 I. 24 J. 27 **Answer (y): 21**

Example 5: Cloze Problem

Reformulated Question (x'): What is the largest number of cells in a 9×9 board that can be cut along both diagonals without the board falling into pieces? The answer should look like: 2_. Fill the blank by giving the full answer. **Answer (y): 21**

D.2 Prompts for Reformulation

We use the following prompts to conduct problem reformulation:

Prompt for Open-ended → Four Choice MCQ

Please create a multiple-choice question with 4 options (A, B, C, D) based on the following math problem and its correct answer.

Original Question: `{question}`
Correct Answer: `{gold_answer}`

Requirements:
1. Keep the original question
2. Add 4 options (A, B, C, D)
3. One option should be the correct answer
4. The other 3 options should be plausible but incorrect distractors
5. Randomly place the correct answer among the options

Output format:
Question: [the question]
A. [option A]
B. [option B]
C. [option C]
D. [option D]

Correct Answer: [letter of correct option]
Only output in this exact format, nothing else.

Prompt for Open-ended → Ten Choice MCQ

Please create a multiple-choice question with 10 options (A, B, C, D, E, F, G, H, I, J) based on the following math problem and its correct answer.

Original Question: `{question}`
Correct Answer: `{gold_answer}`

Requirements:
1. Keep the original question
2. Add 10 options (A, B, C, D, E, F, G, H, I, J)
3. One option should be the correct answer
4. The other 9 options should be plausible but incorrect distractors
5. Randomly place the correct answer among the options

Output format:
Question: [the question]
A. [option A]
B. [option B]
C. [option C]
D. [option D]
E. [option E]
F. [option F]
G. [option G]
H. [option H]
I. [option I]
J. [option J]

Correct Answer: [letter of correct option]
Only output in this exact format, nothing else.

Prompt for Open-ended → Cloze

Your task is to produce a masked version of the correct answer by replacing some digits with underscores (_).

Requirements:
1. Masked answer is the correct answer with some digits replaced by underscores (_)
2. Preserve LaTeX formatting in the masked answer (e.g., if answer is `\frac{1}{2}`, mask it as `\frac{1}{_}` or similar)
3. Mask approximately 50-80% of the digits, keeping at least one digit visible
4. Only mask numbers, not letters or latex symbols

Output format: \boxed{[masked answer with underscores only]}

Examples: If the answer is 1003, output: \boxed{1__3} or \boxed{__03}
If the answer is \frac{5}{8}, output: \boxed{\frac{5}{_}} or \boxed{\frac{_}{8}}

Only output the masked answer in \boxed{}, nothing else.

Original Question: `{question}`
Correct Answer: `{gold_answer}`
Masked Answer:

Table 6: Full per-benchmark results for Figure 4. Re-evaluating GRPO and Cog-DRIFT at 16k and 32k maximum generation length. Training uses an 8k limit for all methods. A longer budget narrows the gap but does not close it, and Cog-DRIFT at 8k already averages slightly higher than GRPO at 32k. Results are with Qwen.

Eval. budget	BMH	Omni-Hard	AIME24	AIME25	GPQA	Date	Avg.
GRPO (8k)	6.37	16.37	31.87	27.29	31.81	**89.53**	33.87
Cog-DRIFT (8k)	10.11	18.58	51.74	40.28	36.01	88.75	40.92
GRPO (16k)	6.87	20.28	48.87	41.11	28.62	89.43	39.19
Cog-DRIFT (16k)	10.38	20.56	54.33	43.34	36.58	88.60	42.30
GRPO (32k)	6.46	21.01	52.22	43.34	29.04	**89.53**	40.27
Cog-DRIFT (32k)	**11.03**	**21.87**	**56.66**	**46.67**	**37.01**	89.04	**43.71**

E Additional Results

This section reports results that supplement Section 6. Appendix E.1 extends the main comparison to larger generation budgets. Appendix E.2, Appendix E.3, and Appendix E.4 report additional baselines and controls that isolate the mechanism behind the gains. Appendix E.5 and Appendix E.6 give the per-benchmark numbers behind the two ablations in Section 6.

E.1 Full Generation-Budget Results

Table 6 reports the per-benchmark numbers behind Figure 4, re-evaluating GRPO and Cog-DRIFT at 16k and 32k maximum generation length.

E.2 Cog-DRIFT Outperforms Guided Exploration Without Privileged Info

A natural question is whether the gains come from injecting privileged information into otherwise reward-less rollouts, which is what guided-exploration methods do. To test this, we compare two such baselines using Qwen in Table 7, each supplying that information in a different form. HINT (Wang et al., 2026) injects stepwise guidance into rollouts that fail, steering the reasoning without disclosing the answer. QuestA (Li et al., 2026) augments each hard question with a partial solution so that rollouts begin inside a reward-bearing region. Both improve over standard GRPO on average (36.08 and 36.25 against 33.87), confirming that added signal does help on $\mathcal{D}_{\text{hard}}$. Cog-DRIFT nevertheless improves over QuestA, the stronger of the two, by +4.67% on average, by +2.76% on BMH (7.35 $\rightarrow$ 10.11), and by +11.27% on GPQA-Diamond (24.74 $\rightarrow$ 36.01). We hypothesize that the difference lies in what the privileged information encodes. These baselines condition learning on *how* to solve the problem, through partial traces or stepwise cues, and therefore need either a stronger model or a reasoning signal that is itself hard to obtain on $\mathcal{D}_{\text{hard}}$. Cog-DRIFT needs only the gold answer y_i to build its reformulations. Overall, reducing the cognitive load of a problem is an effective mechanism for unlocking learning.

Table 7: Guided-exploration comparisons with Qwen3-4B-Instruct-2507. Both baselines supply privileged reasoning guidance, yet Cog-DRIFT, which needs only the gold answer, outperforms the stronger of them by +4.67% on average. Avg. is over all six benchmarks.

Method	BMH	Omni-Hard	AIME24	AIME25	GPQA	Date	Avg.
GRPO	6.37	16.37	31.87	27.29	31.81	**89.53**	33.87
HINT	6.65	16.01	42.25	38.44	26.22	86.93	36.08
QuestA	7.35	16.45	44.38	36.87	24.74	87.69	36.25
Cog-DRIFT	**10.11**	**18.58**	**51.74**	**40.28**	**36.01**	88.75	**40.92**

E.3 Sequential RFT Then GRPO Does Not Match the Joint Curriculum

Section 3 shows that RFT on cloze variants already transfers back to open-ended problems, so a simpler pipeline would be to fine-tune on reformulated trajectories first and then run GRPO on the original open-ended problems. Table 8 compares this sequential composition against its two parts. On its own, RFT lifts BMH from 0 to 4.32 and is level with GRPO on average (33.76 against 33.87), echoing Figure 2(c). Running GRPO on top raises BMH to 7.75 and GPQA-Diamond to 26.85, yet the average remains level with GRPO alone (33.25 against 33.87) and well below Cog-DRIFT (40.92). This suggests that the benefit does not come from exposure to reformulated data as such. Interleaving the reformulations with RL through an adaptive curriculum, rather than consuming them in a separate stage, is what produces the gain. RFT here is the same baseline reported in Table 1, performed on the correct trajectories elicited from the cloze reformulations.

Table 8: Per-benchmark results for sequential RFT followed by GRPO on the original open-ended problems, with Qwen. RFT + GRPO improves over RFT alone on BMH and GPQA-Diamond, but its average remains level with GRPO alone and well below Cog-DRIFT.

	BMH	Omni-Hard	AIME24	AIME25	GPQA	Date	Avg.
RFT	4.32	15.13	40.06	34.88	23.04	85.11	33.76
GRPO	6.37	16.37	31.87	27.29	31.81	**89.53**	33.87
RFT + GRPO	7.75	14.59	33.67	31.17	26.85	85.49	33.25
Cog-DRIFT	**10.11**	**18.58**	**51.74**	**40.28**	**36.01**	88.75	**40.92**

E.4 Reformulation Does Not Reduce to Answer Leakage

MCQ options and cloze masks expose part of the answer, so in principle the model could verify candidates backward instead of deriving answers forward. We provide two pieces of evidence to argue against this. First, the transfer ordering in Section 3 runs opposite to what a backward-verification account would predict. Transfer back to the open-ended problems improves as the format becomes *more* generative (4-choice 11.1%, 10-choice 15.5%, and cloze 18.9%), whereas the most discriminative and most easily verified format should transfer best if the model were selecting rather than deriving. Second, we construct an answer-leakage variant in the spirit of STaR's hindsight rationalization (Zelikman et al., 2022), appending the gold answer to the question. It reaches near-ceiling in-domain accuracy (86.18 against 18.95 for cloze) but collapses on the held-out MATH500 (58.60), far below the 82.20 that our reformulation mixture reaches under the same RFT protocol (Table 9). Once the answer is revealed, the model memorizes training instances without acquiring transferable reasoning. Overall, the reformulations shrink the output space without removing the reasoning the problem requires, which is what separates them from answer-conditioned training.

E.5 Full Reformulation Mixture Results

Recall that in Section 6, we report that combining multiple reformulations is what drives transfer back to the original open-ended problems. Table 9 gives the results behind that claim. Here we remove the curriculum, fine-tune the model with RFT on fixed mixtures of problem types, and then evaluate on OmniMATH-Hard and MATH500. Since the model cannot produce correct trajectories for the open-ended problems in $\mathcal{D}_{\text{hard}}$, we obtain the OEQ trajectories through post-hoc rationalization, prompting the model with the gold answer and keeping the rationales that reach it (Zelikman et al., 2022).

Table 9: Combining reformulations improves transfer to OmniMATH-Hard and MATH500. The first row is the untrained model. A single reformulation lowers at least one of the two held-out scores, whereas 4-MCQ + Cloze raises both and performs on par with the mixture that also includes open-ended questions. Adding open-ended questions to a single reformulation lowers OmniMATH-Hard below the untrained model. These results suggest that a diverse mixture of reformulations drives transfer. Results are with Qwen under RFT, and OEQ trajectories come from post-hoc rationalization with the gold answer.

Data Source	**Omni-Hard**	**MATH500**
-	15.67	78.80
4-MCQ	14.82	77.20
Cloze	16.15	74.40
4-MCQ + Cloze	18.36	**82.60**
OEQ + 4-MCQ	12.39	75.40
OEQ + Cloze	12.61	79.80
OEQ + 4-MCQ + Cloze	**18.42**	82.20

Table 10: Sensitivity of the instance-level curriculum along its three design axes. The curriculum is robust to the promotion threshold, every setting of which still clears the strongest baseline (36.25), but it depends on a difficulty-respecting format ordering. Results are with Qwen.

	BMH	Omni-Hard	AIME24	AIME25	GPQA	Date	Avg.
(a) Promotion threshold τ							
$\tau = 0.1$	11.96	18.07	**55.56**	**46.67**	35.52	87.17	**42.49**
$\tau = 0.3$	**12.18**	16.74	52.22	40.00	36.03	87.99	40.86
$\tau = 0.5$ (default)	10.11	**18.58**	51.74	40.28	36.01	**88.75**	40.92
$\tau = 0.7$	9.92	15.34	47.78	41.11	**36.87**	87.17	39.70
$\tau = 0.9$	9.49	16.74	48.89	37.78	34.51	87.99	39.23
(b) Format ordering							
Reversed (hard to easy)	6.11	15.03	36.76	33.34	20.19	85.17	32.77
Random permutation	6.94	16.71	40.11	36.83	26.72	86.33	35.61
Easy to hard (default)	**10.11**	**18.58**	**51.74**	**40.28**	**36.01**	**88.75**	**40.92**
(c) Promotion granularity							
Per-batch	9.63	17.34	**53.33**	**41.11**	34.37	87.49	40.55
Per-instance (default)	**10.11**	**18.58**	51.74	40.28	**36.01**	**88.75**	**40.92**

E.6 Full Curriculum Ablation Results

Recall that in Section 6, we report that the instance-level curriculum is robust to the promotion threshold but depends on a difficulty-respecting format ordering. Table 10 gives the per-benchmark results behind that claim, along each of the three design axes.

E.7 Examples of Flawed Problems

When we use pass@k to collect $\mathcal{D}_{\text{hard}}$, we find that a large portion of the retained problems are flawed. Some have incorrect gold answers, some are incomplete or ambiguous, and some refer to a figure that is not provided. We list one example of each type below.

Example 6: Incorrect Gold Answer

Question: The cube root of a number is equal to the square root of the number. What is the number? **Gold Answer:** 0 (the correct answer is 0 or 1)

Example 7: Incomplete or Ambiguous Question

Question: Using five sixes, arithmetic operations, and exponentiation, form numbers from 1 to 14. **Gold Answer:** 1-14

Example 8: Missing Figure

Question: As shown in the figure, in a rectangle with an area of 10,000 square centimeters, a large semicircle and two equal small semicircles are cut out. The remaining shaded area is _ square centimeters. (Take π as 3.14) **Gold Answer:** 2150

F Qualitative Examples of Baselines and Cog-DRIFT Outputs

Example 9: Output from Qwen3-4B-Instruct-2507

Question: When 4444^{4444} is written in decimal notation, the sum of its digits is A. Let B be the sum of the digits of A. Find the sum of the digits of B (A and B are written in decimal notation.) **Gold Answer: 7 Reasoning:** We are given the number 4444^{4444}, and we are to compute the sum of the digits of this number, call that A, then the sum of the digits of A is B, and finally, we are to find the sum of the digits of B. Let's go step-by-step. Step 1: Understand what we are dealing with 4444^{4444} is an enormous number. We cannot compute it directly. Instead, we need to find the digit sum of this number, then digit sum of that, then digit sum again. $A \equiv 4444^{4444} \pmod 9$ $B \equiv A \pmod 9$ Then the sum of digits of B is $\equiv B \pmod 9$, and since the digit sum of a number is congruent to the number mod 9, we can use that to find the final answer. Moreover, since the sum of digits of a number is at most a certain value, and we are repeatedly taking digit sums (like in the process of reducing a number to its digital root), we can use the concept of digital root. Step 2: The digital root of a number is the iterative sum of digits until a single digit is obtained. Digital root of 4444^{4444} is $A \to B \to$ (sum of digits of B) So the final answer is the digital root of 4444^{4444}. Therefore, we just need to compute: $4444^{4444} \mod 9$ Then, if the result is 0, the digital root is 9; otherwise, it's the remainder. Step 3: Compute $4444 \mod 9$ We can compute $4444 \mod 9$ by summing digits: Digits of 4444: $4+4+4+4=16$ Then $1+6=7$ So 4444 $equiv7 \pmod 9$ Therefore: $4444^{4444} \equiv 7^{4444} \pmod 9$ Now we need to compute $7^{4444} \mod 9$ Step 4: Let's compute powers of 7 modulo 9: $7^1 = 7 \mod 9 = 7$ $7^2 = 49 \mod 9 = 49 - 54 = -5 \equiv 4 \mod 9$ $7^3 = 7 \cdot 7^2 = 7 \cdot 4 = 28 \mod 9 = 1$ $7^4 = 7 \cdot 1 = 7 \mod 9$ Pattern: $7, 4, 1, 7, 4, 1, \ldots$ Cycle of length 3: $7, 4, 1$ So the cycle is period 3 . Step 5: Find 4444 $mod3$ Sum of digits of 4444: $4+4+4+4=16$, $1+6=7$, $7 \mod 3 = 1$ So $4444 \equiv 1 \pmod 3$ Therefore, in the cycle $(7, 4, 1)$, since exponent $4444 \equiv 1 \pmod 3$, it corresponds to the first position. Step 6: Digital root is 7. So the digital root of 4444^{4444} is 7. This means: $A \equiv 7 \pmod 9$ $B \equiv 7 \pmod 9$. The sum of digits of B is $\equiv 7 \pmod 9$ Step 7: Estimate the number of digits of 4444^{4444} Number of digits of a number N is $\lfloor \log_{10} N \rfloor + 1$. First compute 4444×3.643 Break it down: $4444 \times 3 = 13332$ $4444 \times 0.6 = 2666.4$ $4444 \times 0.04 = 177.76$ $4444 \times 0.003 = 13.332$ Add them: $13332 + 2666.4 = 15998.4$, $+177.76 = 16176.16$, $+13.332 = 16189.492$ So total log 16189.492 Thus, number of digits 16190 So 4444^{4444} has about 16,190 digits Now, the maximum possible digit sum A is $9 \times 16190 = 145,710$ So $A \leq 145,710$ Now, B is the sum of digits of A. Since $A \leq 145,710$, which is a 6-digit number. So $B \leq 9+9+9+9+9+9 = 54$ So $B \leq 54$ Now, $B \equiv 7 \pmod 9$. So possible values of B: 7, 16, 25, 34, 43, 52. Now, for each, compute the sum of digits: 7: sum = 7 16: $1+6=7$, 25: $2+5=7$, 34: $3+4=7$, 43: $4+3=7$, 52: $5+2=7$ All of them have digit sum = 7! So regardless of which one it is, the sum of the digits of B is 7

Final Answer: $\boxed{7}$

Example 10: Output from Qwen trained with GRPO

Question: Let a, b, c, d be real numbers such that $b - d \geq 5$ and all zeros x_1, x_2, x_3, and x_4 of the polynomial $P(x) = x^4 + ax^3 + bx^2 + cx + d$ are real. Find the smallest value the product $(x_1^2 + 1)(x_2^2 + 1)(x_3^2 + 1)(x_4^2 + 1)$ can take. **Gold Answer:** 16 **Reasoning:** We are given a polynomial $P(x) = x^4 + ax^3 + bx^2 + cx + d$ with all real roots x_1, x_2, x_3, x_4, and the condition that $b - d \geq 5$. We are to minimize the product $(x_1^2 + 1)(x_2^2 + 1)(x_3^2 + 1)(x_4^2 + 1)$, given this constraint. Step 1: Understand the expression. We are to minimize $\prod_{i=1}^{4}(x_i^2 + 1)$.. We can write this in terms of symmetric functions. Note that:$(x_i^2 + 1) = (x_i - i)(x_i + i)$, so the full product becomes:$\prod_{i=1}^{4}(x_i^2 + 1) = \prod_{i=1}^{4}(x_i - i)(x_i + i) = \prod_{i=1}^{4}(x_i - i) \cdot \prod_{i=1}^{4}(x_i + i) = P(i) \cdot P(-i)$.. So, $\prod(x_i^2 + 1) = P(i) \cdot P(-i) = |P(i)|^2$.. Key observation: The product $(x_1^2 + 1)(x_2^2 + 1)(x_3^2 + 1)(x_4^2 + 1) = |P(i)|^2$. So, we want to minimize $|P(i)|^2$ over real roots x_1, x_2, x_3, x_4 such that $b - d \geq 5$. Step 2: Express $b - d$ in terms of roots. Given polynomial with real roots x_1, x_2, x_3, x_4, we write:$P(x) = (x - x_1)(x - x_2)(x - x_3)(x - x_4)$.. We are given $b - d \geq 5$. So we must minimize $|P(i)|^2$ subject to $\sum_{i<j} x_i x_j - x_1 x_2 x_3 x_4 \geq 5$.. So symmetric even roots don't satisfy the constraint. We need asymmetric roots. Try three equal roots and one different? Or two equal, and two symmetric others? Alternatively, suppose two roots are equal, and the other two are symmetric. But instead, let's suppose three roots are equal, and the fourth is different. But let's try to assume symmetry between two pairs with different signs. Suppose the roots are $p, q, -p, -q$. That's symmetric, but we just saw that gives negative $b - d$. Wait, we must have $b - d \geq 5$, and $b - d = \sum_{i<j} x_i x_j - \prod x_i$. So we want the sum of products of pairs minus the product of all four to be at least 5. Let's denote: $s_1 = \sum x_i$- $s_2 = \sum_{i<j} x_i x_j$- $s_3 = \sum_{i<j<k} x_i x_j x_k$- $s_4 = x_1 x_2 x_3 x_4$. So now, we have: Minimize $(1 - b + d)^2 + (c - a)^2$ Subject to $b - d \geq 5$. Let $x = b - d \geq 5$. Then $d = b - x$. So $1 - b + d = 1 - b + (b - x) = 1 - x$. So $(1 - b + d)^2 = (1 - x)^2$. Now, the second term is minimized when $x = 5$. Thus, the smallest possible value of the expression is $(5 - 1)^2 = 16$. So the product $(x_1^2 + 1)(x_2^2 + 1)(x_3^2 + 1)(x_4^2 + 1) = |P(i)|^2$ can be as small as 16, when $b - d = 5$, and $c = a$. Step 4: Construct a polynomial with real roots, satisfying $b - d = 5$, $c = a$. We now try to construct such a polynomial. We want: $b - d = 5$- $a = c$. We are free to choose the roots as long as they are real. Let us suppose two roots are equal, and the other two symmetric. Try roots: p, p, q, q. So symmetric. Then the polynomial is: $(x - p)^2(x - q)^2 = (x^2 - 2px + p^2)(x^2 - 2qx + q^2)$. Assume $p + q \geq 0$, divide both sides:$1 = pq$. So $pq = 1$. Now, $b - d = 5$. We compute $b - d$: $b = p^2 + 4pq + q^2 = p^2 + q^2 + 4$- $d = p^2 q^2 = (pq)^2 = 1$. So $b - d = p^2 + q^2 + 4 - 1 = p^2 + q^2 + 3$. Set equal to 5: $p^2 + q^2 + 3 = 5 \Rightarrow p^2 + q^2 = 2$. But $pq = 1$, so we have: $p^2 + q^2 = 2$- $pq = 1$. Try $p = q = 1$. Then roots: $1, 1, 1, 1$, all equal. Polynomial: $(x - 1)^4$. Check a, b, c, d: $a = -4$- $b = 6$- $c = -4$- $d = 1$. So $b - d = 6 - 1 = 5$. And $c = -4$, $a = -4$, $a = c$. Perfect! Now compute the product: $(x_1^2 + 1)(x_2^2 + 1)(x_3^2 + 1)(x_4^2 + 1) = (1^2 + 1)^4 = (2)^4 = 16$. Exactly what we wanted! And all roots are real (in fact, all equal). And $b - d = 5$, which satisfies the condition. Therefore, the minimal value is achievable. Final Answer: $\boxed{16}$

Example 11: Output from Qwen trained with Cog-DRIFT

Question: Let n be a positive integer. Find all $n \times n$ real matrices A with only real eigenvalues satisfying $A + A^k = A^T$ for some integer $k \geq n$. **Gold Answer:** 0
Reasoning: We are given a condition for an $n \times n$ real matrix A: $A + A^k = A^T$ for some integer $k \geq n$, and we are to find all such matrices with real eigenvalues. Step 1: Understand the given equation
Given:. $A + A^k = A^T$ (1). Rewriting it: $A^T = A + A^k$. So the transpose of A is equal to A plus a higher power of A
Analyze the equation: Since we are told that $k \geq n$, we are dealing with a high-order power (at least the size of the matrix) The transpose is expressed in terms of the matrix and its power.
Step 2: Use transpose properties. From equation (1): $A^T = A + A^k$, take the transpose of both sides: $A = A^T + (A^k)^T = (A + A^k)^T = A^T + (A^k)^T = A^T + A^{k^T}$. But $(A^k)^T = (A^T)^k$, so: $A = A^T + (A^T)^k$. But from original, $A^T = A + A^k$, so substitute that in: $A = (A + A^k) + (A^T)^k = A + A^k + (A^T)^k$.. Now subtract A from both sides: $0 = A^k + (A^T)^k$. So we are led to the key identity: $A^k + (A^T)^k = 0$ (2). Now use this key equation. Given that $A^k + (A^T)^k = 0$, we write: $A^k = -(A^T)^k$.. Now observe that $(A^T)^k = (A^k)^T$, so: $A^k = -(A^k)^T$. So the matrix A^k is skew-symmetric!
Important property: A^k is skew-symmetric. But we are told that A is a real matrix with real eigenvalues. So the eigenvalues of A are real. But the matrix A^k is skew-symmetric. Skew-symmetric matrices have purely imaginary eigenvalues (in general), unless zero.
So: A^k is skew-symmetric. A has real eigenvalues. Now, for a real matrix with real eigenvalues, A^k will also have real eigenvalues (since real powers of real matrices are real). But skew-symmetric real matrices have purely imaginary eigenvalues. So we have: A^k is skew-symmetric eigenvalues are $\pm i \times$ real. But A^k has real eigenvalues (since A has real eigenvalues). So the only way both happen is: $A^k = 0$ Because zero matrix is both symmetric and skew-symmetric. So we conclude: $A^k = 0$ Then from original equation: $A + 0 = A^T \Rightarrow A^T = A$So A is symmetric!
But we must verify if this is the only solution. So: $A^k = 0$ for some $k \geq n$ - and $A^T = A$ Therefore, we are to find symmetric real matrices with zero power for some $k \geq n$. But symmetric matrices are diagonalizable over real numbers. So if A is symmetric and nilpotent, then: $A^k = 0$ for some $k \geq 1$. But for diagonalizable matrix, $A^k = 0$ implies all eigenvalues are 0. So $A = 0$. So is the zero matrix a solution? Check: $A + A^k = 0 + 0 = 0$ - $A^T = 0$. So yes, $0 = 0$: true.
But are there other solutions? We derived that A^k is skew-symmetric and has real eigenvalues must be zero. So $A^k = 0$, and then $A^T = A$ So the matrix is symmetric and nilpotent. Therefore, the only such matrix is the zero matrix. So, conclusion: The only real matrix with real eigenvalues satisfying $A + A^k = A^T$ for some $k \geq n$ is the zero matrix. But let's verify this.
Verification: Let $A = 0$- $A + A^k = 0 + 0 = 0$ - $A^T = 0$ So it works. Are there other matrices? Suppose A is symmetric and $A^k = 0$Then A is diagonalizable with eigenvalues 0 so $A = 0$ So only zero matrix. But what about other values of k? For example, if $A^2 = 0$, and $A + A^2 = A^T$, then: $A + 0 = A^T \Rightarrow A^T = A$ So again, symmetric and nilpotent. So for any $k \geq n$, if A^k is skew-symmetric and eigenvalues are real only zero. So the condition forces: $A^T = A$ - $A^k = 0$ for some $k \geq n$ Therefore, only the zero matrix. So final boxed answer: $\boxed{0}$.