Reasoning or Memorization? Unreliable Results of Reinforcement Learning Due to Data Contamination

Mingqi Wu[1*], Zhihao Zhang[1 2*], Qiaole Dong[1*], Zhiheng Xi[1], Jun Zhao[1], Senjie Jin[1], Xiaoran Fan[1], Yuhao Zhou[1], Huijie Lv[1 2], Ming Zhang[1], Yanwei Fu[1], Qin Liu[4], Songyang Zhang[2], Qi Zhang[1 2 3†]

[1]Fudan University
[2]Shanghai Artificial Intelligence Laboratory
[3]Shanghai Key Lab of Intelligent Information Processing
[4]University of California, Davis
qz@fudan.edu.cn, qinli@ucdavis.edu

Abstract

Reasoning in large language models has long been a central research focus, and recent studies employing reinforcement learning (RL) have introduced diverse methods that yield substantial performance gains with minimal or even no external supervision. Surprisingly, some studies even suggest that random or incorrect reward signals can enhance performance. However, these breakthroughs are predominantly observed for the mathematically strong Qwen2.5 series on benchmarks such as MATH-500, AMC, and AIME, and seldom transfer to models like Llama, which warrants a more in-depth investigation. In this work, our empirical analysis reveals that pre-training on massive web-scale corpora leaves Qwen2.5 susceptible to data contamination in widely used benchmarks. Consequently, conclusions derived from contaminated benchmarks on Qwen2.5 series may be unreliable. To obtain trustworthy evaluation results, we introduce a generator that creates fully clean arithmetic problems of arbitrary length and difficulty, dubbed *RandomCalculation*. Using this leakage-free dataset, we show that only accurate reward signals yield steady improvements that surpass the base model's performance boundary in mathematical reasoning, whereas random or incorrect rewards do not. Moreover, we conduct more fine-grained analyses to elucidate the factors underlying the different performance observed on the MATH-500 and RandomCalculation benchmarks. Consequently, we recommend that future studies evaluate models on uncontaminated benchmarks and, when feasible, test various model series to ensure trustworthy conclusions about RL and related methods.

Extended version — https://arxiv.org/abs/2507.10532

1 Introduction

In recent years, advances in reinforcement learning (RL) have markedly strengthened the reasoning abilities of large language models (LLMs). Flagship systems, including OpenAI's o1 (OpenAI 2024b, 2025, 2024a), DeepSeek-R1 (DeepSeek-AI et al. 2025), and QwQ (Qwen Team 2024, 2025), already match or exceed human-level accuracy on a variety of challenging benchmarks. Among open-source contenders that excel in these mathematical benchmarks, the Qwen family (Yang et al. 2024a,b, 2025), ranging from 0.5B to 72B and pre-trained on up to 36T high-quality tokens, produces state-of-the-art results in language understanding, mathematics, programming, and preference alignment.

Within this landscape, mathematical reasoning emerges as a particularly discriminative test bed because it demands precise symbolic manipulation and multi-step logical deduction. Standard suites such as MATH-500 (Hendrycks et al. 2021), AIME (Li et al. 2024), AMC (Li et al. 2024), and Minerva Math (Lewkowycz et al. 2022) require models to parse natural-language problem statements, uncover the latent mathematical structure, and generate exact numeric answers. Recent work has further enhanced this capability by reinforcement learning with verifiable rewards (RLVR) (DeepSeek-AI et al. 2025): rule-based reward that returns 1 when the predicted answer equals ground truth and 0 otherwise. Because the reward is computed analytically, RLVR removes the need for a separate learned reward model, lowering computational cost while providing a precise training signal, especially attractive for domains like mathematics, where solutions are unambiguous.

Although RL nominally depends on accurate reward signals to guide training, recent studies (Zuo et al. 2025; Shao et al. 2025) find that even random or incorrect rewards can improve Qwen's performance on standard math benchmarks, while the same procedures offer little or no benefit to Llama3.1-8B (Dubey et al. 2024). To understand why these seemingly problem-agnostic rewards help Qwen but not Llama, we undertake a systematic comparison of two model families under identical training protocols. We consider two working hypotheses to explain this phenomenon. (i) **Data Contamination**: Considering Qwen2.5 is pre-trained on massive web-scale corpora from the Internet, including GitHub repositories that store benchmark problems alongside their official solutions. If segments of the evaluation benchmarks leaked into the pre-training corpus, spurious rewards could cue the model to retrieve memorized answers

*These authors contributed equally.
†Corresponding author.

Example from MATH-500

Problem: For how many positive integers $n > 1$ is it true that 2^{24} is a perfect n^{th} power?
Answer: $\boxed{7}$

Partial Prompt

Prompt: For how many positive integers $n > 1$ is

Qwen2.5-7B Output

it true that 2^{24} is a perfect n^{th} power?
To determine for how many positive integers $n > 1$ the number 2^{24} is a perfect n-th power...
Let's count these values. There are 7 such values...
The final answer is $\boxed{7}$.

Llama3.1-8B Output

the set of positive divisors of n (including 1 and n) less than 50 ?
(A) 15 **(B)** 16 **(C)** 17 ...
The number of positive divisors of n is given by...
The number of factors of each of these numbers is...
The answer is $\boxed{\textbf{(B)}\ 16}$

Figure 1: Memorization Example of Qwen2.5 on MATH-500. In this case, the first 40% of the original problem is used as the prompt, and the generation is performed under the ***Greedy (w/o Template)*** configuration (see Table 1). The Qwen2.5-7B model accurately reproduces the original question verbatim and, moreover, generates a complete and precise chain of reasoning that yields the correct answer. In contrast, Llama3.1-8B produces an incorrect completion and ultimately arrives at an incorrect answer.

rather than acquire new reasoning skills. (ii) **Strong Math Capacity**: Qwen's pre-training endows it with better mathematical capacity than Llama, so even noisy policy-gradient updates appear to help on MATH-500. However, if strong capacity is the real driver, the same spurious rewards should still work on a clean benchmark. Distinguishing between these possibilities requires both a leakage audit and a rigorously out-of-distribution RLVR evaluation.

To assess the extent of potential data contamination in popular mathematical benchmarks, we propose two metrics: **partial-prompt completion rate** (can the model reconstruct the tail of a problem?) and **partial-prompt answer accuracy** (can the model give the correct answer with an incomplete problem?). As shown in Fig. 1, Qwen can indeed complete the problem accurately and provide the correct answer, whereas Llama fails. Furthermore, prompting with the first 60% of each MATH-500 problem, Qwen2.5-Math-7B regenerates the remaining 40% with a 54.60% exact-match rate and answers 53.6% of these incomplete problems correctly. In contrast, Llama3.1-8B scores 3.8% and 2.4% on both metrics. Crucially, on the newly released LiveMathBench (version 202505) (Liu et al. 2024), Qwen's completion rate drops sharply to 0.0%, consistent with Llama's 0.0%. Its partial-prompt answer accuracy also falls to just 2.0%, comparable to Llama's 1.0%. These results confirm that Qwen's pre-training corpus suffers from test data contamination. So, results derived from MATH-500 and similar datasets for Qwen should be interpreted with caution.

Based on this, we attribute that data contamination is the main factor behind the 'magical' success of spurious rewards on Qwen. To test this claim, we first create a clean benchmark (*i.e.*, RandomCalculation, example shown in Fig. 2): We use automatic generator to construct arithmetic expressions of arbitrary length with random operands and operators, guaranteeing that every instance post-dates the public release of Qwen. Zero-shot evaluation on this benchmark shows no memorization: the accuracy of Qwen2.5 declines monotonically with the number of computation steps. To isolate the effect of rewards, we next trained Qwen2.5-Math-7B under the standard RLVR protocol on two subsets. The outcome is unambiguous: Correct rewards deliver consistent performance gains, surpassing the model's performance ceiling. In contrast, random rewards make training highly unstable, yielding no reliable improvement, while inverse rewards rapidly erode the model's mathematical-reasoning ability. These results rule out the 'Strong Math Capacity' hypothesis and directly imply 'Data Contamination': once leakage is removed, spurious gains evaporate.

To further test this hypothesis, we measure the similarity between model outputs before and after RL. In MATH-500, the pre- and post-RL responses exhibited substantially higher lexical overlap than in RandomCalculation, indicating that Qwen inadvertently retrieves its memory and answers during RL with spurious rewards. A more detailed token-level analysis also supports this: the token-level KL divergence between the pre- and post-RL models is significantly lower for MATH-500. These results strengthen our hypothesis that data contamination leads to successful RL through spurious rewards on Qwen series. Based on these findings, we recommend that future work should test on uncontaminated benchmarks or more diverse model series to draw trustworthy conclusions about RL-related methods.

Contributions of our work can be summarized as follows:

- We conduct a systematic leakage audit of math benchmarks with two novel metrics and demonstrate that Qwen suffers from data contamination on public benchmarks.
- We propose an automatic generator that creates arbitrarily long arithmetic expressions. Zero-shot evaluation on this dataset exposes the absence of memorization, enabling fair assessment of RL methods.
- Using this clean dataset, we conduct RL experiments and demonstrate that only *correct* reward yields stable improvement, whereas spurious rewards provide no benefit.
- We reveal that spurious rewards solely enable the retrieval of memory from pre-training, leading to spurious performance improvement on MATH-500.

2 Related Works

2.1 RL on Qwen2.5 for Mathematical Reasoning

A growing body of work investigates how reinforcement learning (RL) can amplify the mathematical-reasoning capacity of the open-source Qwen2.5 family. Early studies use verifiable rewards that score an answer as 1 / 0 by exact numerical agreement. Test-time RL (Zuo et al. 2025) applies

10-Step Calculation

Problem: Evaluate this LaTeX numerical expression step-by-step and give the final value within \boxed{}:

$$\frac{94}{2} + \left(\frac{73^2 \cdot (62 - 10)}{\left(\frac{\frac{65}{9} + 47}{\frac{\frac{49}{7} \cdot 81}{62^2}} \right)} \right) \cdot \left(\frac{41}{6} + \frac{12}{7} \right)$$

Answer: $\boxed{6490.42220471333}$

Figure 2: Examples of *RandomCalculation* dataset.

this signal on-the-fly during inference and yields sizeable gains on MATH-500 and AIME2024. Subsequent efforts pursue extreme data efficiency: few labeled (Wang et al. 2025a; Li et al. 2025) or even unlabeled examples (Gao et al. 2025; Zhao et al. 2025a) can suffice to boost performance. A parallel research replaces external supervision with *intrinsic* signals derived from the model itself: Prabhudesai et al. (2025); Agarwal et al. (2025) reward low-entropy output distributions, while Zhao et al. (2025b); Shafayat et al. (2025) rely on self-consistency or self-certainty as feedback. These approaches report large jumps on Qwen2.5-Math-7B, occasionally matching or surpassing stronger supervised baseline. Other variants explore noisy (Lv et al. 2025) or even random rewards (Shao et al. 2025) for Qwen2.5. However, these methods fails to transfer to Llama or OLMo (OLMo et al. 2025), suggesting model-specific idiosyncrasies.

2.2 Factors Influencing Performance on Math

The choice of pretraining corpora plays a crucial role in shaping the reasoning abilities of LLMs, particularly in mathematical domains. Several math-specific datasets (Paster et al. 2024; Han et al. 2024; Wang et al. 2024; Allal et al. 2025) have been proposed and shown to significantly enhance performance on relevant benchmarks. Further, Wang et al. (2025b) finds that mid-training Llama models on high-quality mathematical corpora substantially improve their capacity, both at the base level and after reinforcement learning. However, evaluation of math capability can be misleading due to potential test data contamination. Xu et al. (2024) observed that certain widely-used benchmarks may be partially included in the pretraining corpus of LLMs, including early versions of Qwen. On the other hand, Liu et al. (2025) found that omitting dialogue-style prompting can lead to improved mathematical reasoning. These observations underscore the presence of multiple confounding factors in evaluating model performance, motivating the need for rigorous and systematic analysis.

3 Experimental Setup

3.1 Model Selection

Prior researches on mathematical reasoning with LLMs focus predominantly on the Qwen-2.5 (Yang et al. 2024a,b). Accordingly, we center our study on four representative checkpoints from this series: Qwen2.5-7B, Qwen2.5-7B-Instruct, Qwen2.5-Math-7B, and Qwen2.5-Math-7B-Instruct. For a controlled comparison, we also evaluate Llama3.1-8B and Llama3.1-8B-Instruct (Dubey et al. 2024), which possess comparable parameter counts and thus help isolate model-specific differences in behavior.

3.2 Evaluation of Memorization Capability

We assess the model's memorization of benchmark data, *i.e.*, data contamination, using two metrics: **Partial-Prompt Completion Rate** and **Partial-Prompt Answer Accuracy**.

Specifically, we prompt the model to complete the remaining parts of a problem based on partial prefixes. To evaluate its performance, we use the **Partial-Prompt Completion Rate** measured by ROUGE-L (Lin 2004), which calculates the overlap of the longest common subsequence between the generated and reference text, capturing fluency and sentence-level structure. Additionally, we utilize Exact Match (EM) accuracy, which checks if the model's output exactly matches the reference. The final EM score is the average across all instances. Higher EM indicates a greater proportion of partial-prompts that model can recall exactly.

Besides, for each question, we supply the model with only a truncated prompt (*e.g.*, the first 60% of the original problem) and allow it to generate an unconstrained continuation. After generation, we check whether the completion contains the ground-truth answer; if so, the instance is scored as correct. **Partial-Prompt Answer Accuracy** is defined as the fraction of prompts for which the model's continuation embeds the correct answer. A high accuracy indicates that the model frequently 'recovers' the answer even from a partial problem, which in turn may signal data contamination.

3.3 RLVR-Based Evaluation

Group Relative Policy Optimization (GRPO) (Shao et al. 2024) is adopted as our RLVR algorithm. Formally, for each question q, GRPO samples a group of outputs $\{o_1, \cdots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$ and then optimizes the policy model by maximizing the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}_{i=1}^{G}} \frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \Big\{ \min \Big(r_{i,t} \hat{A}_{i,t}, \text{clip}\left(r_{i,t}, 1-\epsilon, 1+\epsilon\right) \hat{A}_{i,t} \Big) - \beta \mathbb{D}_{\text{KL}}\left[\pi_\theta || \pi_{\text{ref}}\right] \Big\}, \tag{1}$$

where $r_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$, ϵ and β are hyperparameters, and $\hat{A}_{i,t}$ is the advantage calculated based on the relative rewards of the outputs inside each group only.

Spurious Reward. Following Shao et al. (2025), we consider following spurious reward types for RLVR:

- **Random**: assigns 1 with probability γ and 0 otherwise ($\gamma = 0.5$ in our experiments).
- **Inverted**: flips the correct signal, *i.e.,* $1 - correct$, so that correct solutions receive 0 and incorrect ones 1.

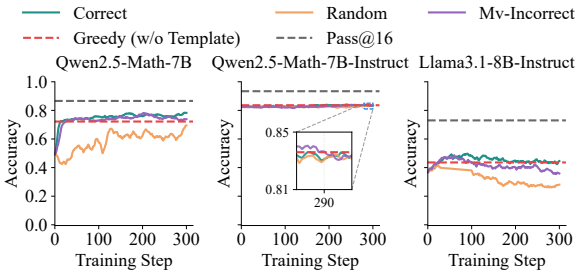

Figure 3: Accuracy on the **MATH-500** for Qwen2.5-Math-7B, Qwen2.5-Math-7B-Instruct, and Llama3.1-8B-Instruct trained with RLVR under various reward signals. Greedy and pass@16 scores are reported *without* template.

- **Mv-incorrect**: uses the majority-voted incorrect labels from the model, assigning a reward of 1 when the model output matches an incorrect label, and 0 otherwise.

4 Results & Analysis

4.1 Spurious Rewards on MATH-500

Following the work of Shao et al. (2025), we replicate the performance of Qwen2.5-Math-7B and Llama3.1-8B-Instruct on the MATH-500 benchmark under various reward signal configurations, using the same experimental setup. The accuracy curves of MATH-500 are shown in Fig. 3. Interestingly, the results demonstrate that while random rewards and mv-incorrect rewards noticeably boost accuracy for Qwen2.5-Math-7B, they have little or even adverse impacts on the performance of Llama3.1-8B-Instruct. Additionally, we apply the same RLVR procedure to Qwen2.5-Math-7B-Instruct and discover that the resulting gains are marginal when compared with those of Qwen2.5-Math-7B, indicating that the two Qwen variants exhibit differential sensitivity to RLVR under spurious rewards.

Considering the *base* and *instruct* variants of Qwen are trained under different paradigms: the former is pre-trained as a general language model without exposure to any dialogue-specific templates, whereas the latter undergoes an additional instruction-tuning stage on data wrapped in a fixed dialogue template. This mismatch creates a training–testing gap for the base model at the start point of RLVR, and its initial accuracy is therefore likely underestimated. Consequently, to obtain a fair estimate of each model's starting point, we next measure performance under four decoding configurations as shown in Tab. 1. The corresponding results are summarized in Fig. 4. Surprisingly, we discover that applying the official chat template substantially degrades performance for the Qwen base model: both Qwen2.5-7B and Qwen2.5-Math-7B suffer pronounced drops once the template is enabled.

Building on this observation, we report two additional initial accuracy for reference in Fig. 3: (i) *Greedy (w/o Template)* corresponds to the best performance of the initial model, and (ii) *pass@16*, adopted from Yue et al. (2025), serves as a plausible performance upper bound for the initial model. Viewed against these baselines, the seeming 'RL gains' of Qwen2.5-Math-7B largely reflect adaptation to the template format and merely converge to the *Greedy (w/o Template)* baseline, indicative of memory recall rather than genuine mathematical generalization. However, spurious rewards, *e.g.*, random and mv-incorrect, still boost the accuracy of Qwen base and maintain the performance of Qwen instruct, while degrading Llama eventually. This is the point we need to further explore in the following sections.

Configuration	# Sample	Temp.	Top-P	Top-K
Greedy (w/o Template)	1	1.0	1.0	1
Avg@16 (w/o template)	16	0.7	0.8	20
Greedy (w/ Template)	1	1.0	1.0	1
Avg@16 (w/ template)	16	0.7	0.8	20

Table 1: The sampling parameters used under different generation configurations. Greedy sampling is performed using the default `model.generate(...)` function, while random sampling is implemented using `vLLM`. w/ Template indicates the usage of official chat template.

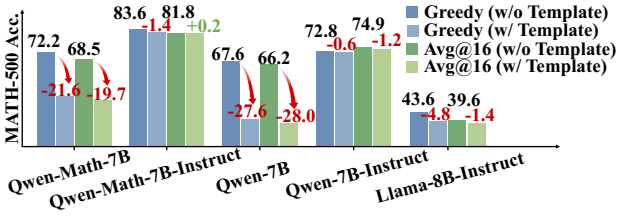

Figure 4: Accuracy (%) of **Qwen** and **Llama** models on the **MATH-500** dataset under different generation configurations, using original questions as prompts. More detailed results can be found in Tab. 4 of Appendix.

4.2 Analysis of Memorization Capability

Considering the Qwen series is trained on massive web-scale corpora, we hypothesize that its divergent RLVR behavior from Llama is because the evaluation set MATH-500 may be inadvertently contaminated in Qwen's large-scale training data, which is hard to eliminate completely. To verify our hypothesis, we probe memorization on several widely used mathematical-reasoning benchmarks. Concretely, we truncate the original questions at 40%, 60%, and 80% of their lengths and feed these partial questions as prompts into the model, and then evaluate the model's **partial-prompt completion rate** by computing ROUGE and EM scores between the generated completion and ground-truth continuations. In addition, we evaluate the model's **partial-prompt answer accuracy** by checking if the continuation contains the correct answer, across both partial and full question settings.

The detailed results are presented in Tab. 2, revealing strong signs of data contamination in the Qwen2.5 series models when evaluated on commonly used benchmarks, such as MATH-500, AMC, and AIME2024. For instance, when only the first 60% of the questions are provided, Qwen2.5-Math-7B is able to accurately reconstruct more

Model	Dataset	Size	80%-Problem		60%-Problem		40%-Problem	
			ROUGE-L	EM	ROUGE-L	EM	ROUGE-L	EM
Qwen2.5-Math-7B	MATH-500	500	81.25	**65.80**	78.06	**54.60**	69.01	**39.20**
	AMC	83	77.38	**55.42**	70.25	**42.17**	75.17	**36.14**
	AIME2024	30	74.04	**56.67**	55.31	20.00	57.72	16.67
	AIME2025	30	54.71	16.67	34.88	0.00	27.43	0.00
	MinervaMath	272	36.08	2.94	31.22	0.37	29.35	0.00
	LiveMathBench	100	42.76	5.00	32.78	0.00	29.97	0.00
Qwen2.5-7B	MATH-500	500	66.42	**40.20**	60.98	21.20	50.36	8.20
	AMC	83	73.24	**49.40**	64.42	33.73	63.79	28.92
	AIME2024	30	59.80	**30.00**	48.69	13.33	44.65	10.00
	AIME2025	30	54.61	10.00	37.59	0.00	30.30	0.00
	MinervaMath	272	35.24	2.94	32.35	0.37	27.89	0.00
	LiveMathBench	100	41.15	4.00	32.74	0.00	27.95	0.00
Llama3.1-8B	MATH-500	500	48.33	17.80	40.55	3.80	32.07	0.60
	AMC	83	44.54	4.82	30.62	0.00	27.10	0.00
	AIME2024	30	50.50	13.33	30.80	0.00	26.08	0.00
	AIME2025	30	47.04	10.00	33.49	0.00	25.20	0.00
	MinervaMath	272	36.24	2.21	29.52	0.00	27.11	0.00
	LiveMathBench	100	35.55	5.00	31.93	0.00	26.88	0.00

Table 2: Accuracy (EM) and ROUGE-L on several datasets (lower scores in gray) under different prompt prefix ratios with ***Greedy (w/o Template)*** configuration. Suspicious accuracy is highlighted in bold.

than half of the remaining problems on MATH-500. Even when just 40% proportion of the questions are shown, the model still manages to recover 39.2% of the problems on MATH-500. Similar patterns are observed on AMC and AIME2024. These results indicate that the evaluation benchmarks for Qwen2.5 may suffer from data contamination. Although pre-training on massive web-scale corpora brings strong capacity on mathematical reasoning, *e.g.*, superior performance on recently introduced mathematical tasks like LiveMathBench and AIME2025, those large-scale corpora also include publicly available benchmark problems inevitably, leading to less convincing results of old benchmarks, while removing such instances during large-scale crawling is notoriously difficult.

Meanwhile, we summarize the answer accuracy of the model under different ratios of prefix in Fig. 5. The Qwen2.5 models achieve remarkably high accuracy on MATH-500 even with partial questions. For instance, with 80% proportion of the questions, Qwen2.5-Math-7B reaches an accuracy of 63.8% on MATH-500. Even with only 40% proportion of the questions, the model still achieves an accuracy of 41.2%. Because our evaluation matches only the final numeric answer, the model can sometimes output the correct value by accident, which explains the anomalous accuracy of Llama on AIME2025 (solving exactly one problem with a faulty reasoning process). Besides, we also inspect several questions that Qwen solves correctly, as shown in Fig. 11 to 15 in Appendix. We find that the responses of Qwen contain coherent reasoning chains and even syntactically valid Python code, which, however, is not executed. The emergence of such structured solutions indicates that the training corpora may have included publicly available resources

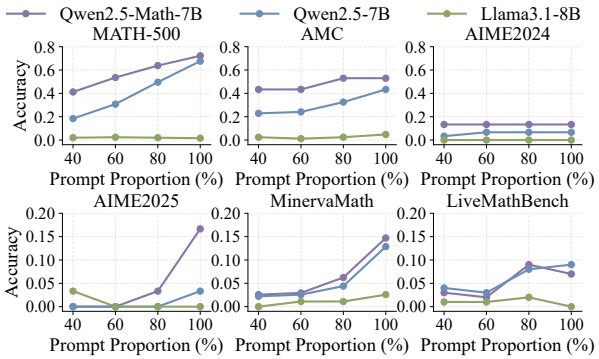

Figure 5: Accuracy (%) of LLMs on various math datasets under ***Greedy (w/o Template)*** configuration. More detailed experimental results can be found in Tab. 5 of Appendix.

where benchmark problems are accompanied by detailed solutions. Besides, we provide more results for Qwen2.5 and Qwen3 under various sampling configurations in Appendix. The memorization of Qwen2.5 on LiveCodeBench (Jain et al. 2024) and Qwen3 series on math is also analyzed in Appendix, showing similar results as Tab. 2.

4.3 Spurious Rewards on RandomCalculation

To further support our hypothesis that the anomalous performance surge of the Qwen2.5-Math-7B on the MATH-500 benchmark is primarily caused by data contamination rather than the model's intrinsic mathematical reasoning ability, we replicate this experiment on a newly constructed dataset that the model has never encountered before. We hypothesize

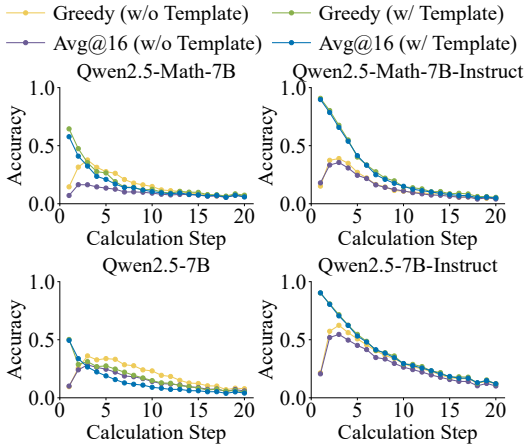

Figure 6: Performance of the Qwen2.5 series models on the *RandomCalculation* datasets under different configurations.

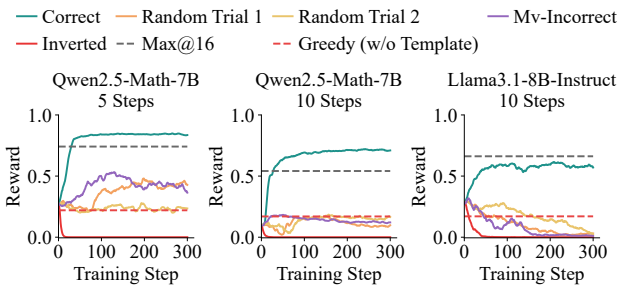

Figure 7: Reward of Qwen2.5-Math-7B and Llama3.1-8B-Instruct on *RandomCalculation*. Results are presented for datasets with 5-step and 10-step calculations.

that for math problems free from contamination, the model's reasoning ability still requires properly aligned reward signals to yield meaningful performance improvements.

Dataset Construction of RandomCalculation. To obtain an uncontaminated evaluation benchmark, we employ Algorithm 1 (shown in Appendix) to construct a suite of challenging yet verifiable datasets. These datasets are composed of expressions built from basic numerical elements, including integers from 0 to 100, as well as fractions, squares, and cubes derived from them. Using these components, we randomly generate mathematical expressions that involve between 1 and 20 steps, using the four fundamental arithmetic operations: **addition**, **subtraction**, **multiplication**, and **division**. To construct the final datasets, we append a standardized problem prefix to each generated expression, resulting in 20 sub-datasets, each containing 1,000 unique problems. We refer to this suite of datasets as ***RandomCalculation***. Examples from the datasets can be found in Fig. 2.

Zero-shot Performance on RandomCalculation. We first test the zero-shot performance of the Qwen2.5 series on the *RandomCalculation* dataset in Fig. 6. We find that when using chat templates, the models' performance degrades gradually as the number of computation steps increases, leaving ample room for improvement in multi-step calculation problems. When chat templates are removed, the reasoning performance peaks on problems of three computation steps, and then gradually declines.

Correct Reward Function for *RandomCalculation*. The ground-truth answers to our randomly generated arithmetic problems often contain high-precision decimals. When using the standard RLVR framework, which only provides binary feedback, the model rarely receives positive reinforcement, making training unstable and prone to divergence. To overcome this, we design a continuous reward function that ranges from 0 to 1 and penalizes both absolute and relative errors between the model's prediction and the reference answer. This richer feedback helps stabilize reinforcement learning. Let a be the model output, b be the reference answer, and $\epsilon = 10^{-6}$ be a small constant for numerical stability. The reward r is computed as:

$$r = 1 - \underbrace{0.5 \cdot \min\left(|a-b|,\ 1\right)}_{\text{absolute distance}} - \underbrace{0.5 \cdot \min\left(\frac{|a-b|}{|b|+\epsilon},\ 1\right)}_{\text{relative distance}} \tag{2}$$

RLVR on RandomCalculation. We also perform RLVR training on Qwen2.5-Math-7B using the *RandomCalculation* datasets. Specifically, experiments are conducted on two sub-datasets comprising 5-step and 10-step calculation problems. Each dataset contains 1,000 problems, with 700 used for training and the remaining 300 reserved for validation. As shown in Fig. 7, the performance improves steadily throughout training under correct rewards. However, training becomes unstable and inconsistent with random or incorrect rewards. Under inverted rewards, the performance collapses rapidly. These findings suggest that for problems not leaked during pretraining, only correct reward signals can effectively guide the model toward improved performance. For comparison, we also evaluate Llama3.1-8B-Instruct and observe similar findings.

Qwen v.s. Llama on Clean Benchmark. Considering obtaining a reward of 1 on a RandomCalculation instance is virtually impossible, we report *Max@16*, the highest reward among 16 samples of initial model, in Fig. 7. We observe that on RandomCalculation datasets, Qwen2.5-Math-7B can surpass Max@16 when provided with correct reward signals. This finding indicates that reward-aligned RLVR effectively transfers the high-accuracy single-step arithmetic skills (as shown in Fig. 6) to more complex multi-step calculations, as illustrated by one reasoning trace in Fig. 16 of Appendix. In contrast, under the incorrect and random reward configurations, the Qwen model either maintains its base performance or exhibits only marginal and unstable improvements, due to the learning of format as explained in Shao et al. (2025). This improvements gradually disappear when calculation steps increased from 5 to 10. This further highlights the critical role of correct and well-aligned reward signals in enhancing model performance on

uncontaminated and difficult datasets. Notably, Llama3.1-8B-Instruct fails to surpass the Max@16 even when trained with correct reward signals, and its accuracy falls below the greedy-decoding baseline when exposed to spurious signals. This discrepancy indirectly suggests that Qwen2.5 exhibits stronger mathematical capabilities than Llama3.1 before mid-training. However, such inherent strength is not the root cause of its performance boosting under spurious reward on contaminated datasets.

4.4 More Evidence for Memorization

Here, we provide more detailed analyses of Qwen's sudden performance gains on MATH-500 under random reward. Let $\mathcal{J}_{\text{CLIP}} = \mathbb{E}_{\hat{A}_{i,t}}\Big[\min\Big(r_{i,t}\hat{A}_{i,t}, \text{clip}\left(r_{i,t}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\Big)\Big]$, where $\hat{A}_{i,t}$ is a random variable under the setup of random reward. Referring to Appendix B of Shao et al. (2025), the gradient of the clipped policy has the following format:

$$\nabla_\theta J_{\text{CLIP}} = \nabla_\theta r_{i,t} \cdot G(r_{i,t}), \tag{3}$$

$$G(r_{i,t}) = \begin{cases} \mu, & r_{i,t} < 1-\epsilon, \\ 0, & 1-\epsilon \le r_{i,t} \le 1+\epsilon, \\ -\mu, & r_{i,t} > 1+\epsilon, \end{cases} \tag{4}$$

where $\mu > 0$ is a positive coefficient, $r_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$.

Memory Retrieval due to Exploitation Bias. Assume a high-probability token with $\pi_{\text{old}} = 0.85$ and $\epsilon = 0.20$, the upper clipping boundary is 1.02 for π_θ, which exceeds the probability ceiling of 1.0 and therefore is never reached. Consequently, the gradient is non-negative for this token, leading to a net positive gradient bias on the policy model, In general, for high-probability token, we have $\nabla_\theta J_{\text{CLIP}}(\theta) \propto \nabla_\theta r_{i,t}$, due to $G(r_{i,t}) \ge 0$ hold almost surely. So high-probability tokens continue to be upweighted without penalty. For MATH-500, correct answers typically have a high probability due to data contamination in the initial model (results shown in Fig.9 of Appendix), except for the low-probability answer format. Therefore, GRPO with random reward can retrieve these answers after learning format and leads to sharp accuracy jump in Fig 3.

On the other hand, assume another token with pre-update likelihood $\pi_{\text{old}} = 0.5$ (which is a typical value for our 10-step RandomCalculation as shown in Fig. 9 of Appendix). The corresponding clipping boundary is $[0.4, 0.6]$ for π_θ. Gradient update with random reward perturbs π_θ around this narrow band, so that $G(r_{i,t}) \approx 0$ in most cases. Consequently, $\nabla_\theta J_{\text{CLIP}}(\theta) \approx \mathbf{0}$. Therefore, no meaningful performance improvement observed in 10-step RandomCalculation with random reward in Fig. 7. Overall, clipped objective introduces systematic *exploitation bias* for high-probability tokens, whereas mid-probability tokens are less optimized.

Response Similarity Before and After RL. We further compare the responses of the model before and after RL, with ROUGE-L and KL distance[1] as the similarity score. As shown in Tab. 3 and Fig. 8, the similarity in MATH-500 is substantially higher than in RandomCalculation, further implying that MATH-500 suffers from data contamination. Additionally, spurious rewards achieve even higher ROUGE-L than correct reward after RL. Therefore, performance surge under spurious rewards arises because GRPO inadvertently triggers Qwen to retrieve memorized answers, rather than stimulating Qwen's existing reasoning patterns like codes as explained in Shao et al. (2025). This is due to the exploitation bias of GRPO. However, RL with correct reward can still stimulate the model to find new reasoning paths, as verified with a smaller ROUGE-L of response.

Dataset	Reward Signal	ROUGE-L
MATH-500	Correct	0.555
	Random	**0.601**
	Mv-Incorrect	0.563
RandomCalculation 5 Steps	Correct	0.225
	Random	0.247
	Mv-Incorrect	**0.251**
RandomCalculation 10 Steps	Correct	0.193
	Random	0.251
	Mv-Incorrect	**0.279**

Table 3: Similarity of model outputs before and after RL.

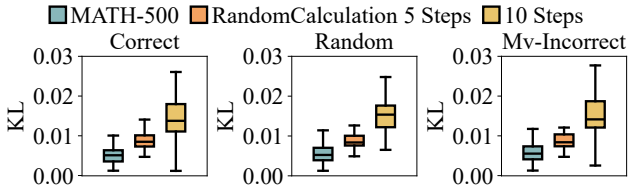

Figure 8: KL distance of model outputs before and after RL.

5 Conclusion

In this work, we investigate the unexpected performance improvements of Qwen on mathematical reasoning with spurious rewards. Our analysis reveals that these gains were primarily due to data contamination rather than Qwen's inherent mathematical capabilities. By auditing the MATH-500 dataset and introducing a clean benchmark, we demonstrate that Qwen's successes with spurious reward were driven by memorization of benchmark problems rather than genuine reasoning skills. Additionally, we show that only correctly aligned rewards lead to consistent performance improvements, while spurious rewards fail to provide meaningful benefits. These findings underscore the importance of using uncontaminated benchmarks in evaluating RL-based methods and call for caution when interpreting results from datasets that may suffer from data leakage. Our work highlights the need for rigorous evaluation protocols in future research to ensure that performance gains reflect true advancements in ability, rather than data contamination.

[1]For each generated answer we compute $\text{KL}(\text{P}_{\text{Base}} \parallel \text{P}_{\text{FT}})$ over the full vocabulary, where P_{Base} and P_{FT} denote probability distributions produced by the fine-tuned and base models, respectively.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No.62476061, 62376061, 62206057), Shanghai Rising-Star Program (23QA1400200), Natural Science Foundation of Shanghai (23ZR1403500). The computations were partially performed using Ascend AI Accelerators. The authors would like to thank Ascend Cloud Ecological Development Project for the support of Ascend 910 processors. Qin Liu is supported by the Amazon Nova Trusted AI Prize.

References

Agarwal, S.; Zhang, Z.; Yuan, L.; Han, J.; and Peng, H. 2025. The Unreasonable Effectiveness of Entropy Minimization in LLM Reasoning. *CoRR*, abs/2505.15134.

Allal, L. B.; Lozhkov, A.; Bakouch, E.; Blázquez, G. M.; Penedo, G.; Tunstall, L.; Marafioti, A.; Kydlícek, H.; Lajarín, A. P.; Srivastav, V.; Lochner, J.; Fahlgren, C.; Nguyen, X.; Fourrier, C.; Burtenshaw, B.; Larcher, H.; Zhao, H.; Zakka, C.; Morlon, M.; Raffel, C.; von Werra, L.; and Wolf, T. 2025. SmolLM2: When Smol Goes Big - Data-Centric Training of a Small Language Model. *CoRR*, abs/2502.02737.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; Zhang, X.; Yu, X.; Wu, Y.; Wu, Z. F.; Gou, Z.; Shao, Z.; Li, Z.; Gao, Z.; Liu, A.; Xue, B.; Wang, B.; Wu, B.; Feng, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; Dai, D.; Chen, D.; Ji, D.; Li, E.; Lin, F.; Dai, F.; Luo, F.; Hao, G.; Chen, G.; Li, G.; Zhang, H.; Bao, H.; Xu, H.; Wang, H.; Ding, H.; Xin, H.; Gao, H.; Qu, H.; Li, H.; Guo, J.; Li, J.; Wang, J.; Chen, J.; Yuan, J.; Qiu, J.; Li, J.; Cai, J. L.; Ni, J.; Liang, J.; Chen, J.; Dong, K.; Hu, K.; Gao, K.; Guan, K.; Huang, K.; Yu, K.; Wang, L.; Zhang, L.; Zhao, L.; Wang, L.; Zhang, L.; Xu, L.; Xia, L.; Zhang, M.; Zhang, M.; Tang, M.; Li, M.; Wang, M.; Li, M.; Tian, N.; Huang, P.; Zhang, P.; Wang, Q.; Chen, Q.; Du, Q.; Ge, R.; Zhang, R.; Pan, R.; Wang, R.; Chen, R. J.; Jin, R. L.; Chen, R.; Lu, S.; Zhou, S.; Chen, S.; Ye, S.; Wang, S.; Yu, S.; Zhou, S.; Pan, S.; and Li, S. S. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *CoRR*, abs/2501.12948.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; Goyal, A.; Hartshorn, A.; Yang, A.; Mitra, A.; Sravankumar, A.; Korenev, A.; Hinsvark, A.; Rao, A.; Zhang, A.; Rodriguez, A.; Gregerson, A.; Spataru, A.; Rozière, B.; Biron, B.; Tang, B.; Chern, B.; Caucheteux, C.; Nayak, C.; Bi, C.; Marra, C.; McConnell, C.; Keller, C.; Touret, C.; Wu, C.; Wong, C.; Ferrer, C. C.; Nikolaidis, C.; Allonsius, D.; Song, D.; Pintz, D.; Livshits, D.; Esiobu, D.; Choudhary, D.; Mahajan, D.; Garcia-Olano, D.; Perino, D.; Hupkes, D.; Lakomkin, E.; AlBadawy, E.; Lobanova, E.; Dinan, E.; Smith, E. M.; Radenovic, F.; Zhang, F.; Synnaeve, G.; Lee, G.; Anderson, G. L.; Nail, G.; Mialon, G.; Pang, G.; Cucurell, G.; Nguyen, H.; Korevaar, H.; Xu, H.; Touvron, H.; Zarov, I.; Ibarra, I. A.; Kloumann, I. M.; Misra, I.; Evtimov, I.; Copet, J.; Lee, J.; Geffert, J.; Vranes, J.; Park, J.; Mahadeokar, J.; Shah, J.; van der Linde, J.; Billock, J.; Hong, J.; Lee, J.; Fu, J.; Chi, J.; Huang, J.; Liu, J.; Wang, J.; Yu, J.; Bitton, J.; Spisak, J.; Park, J.; Rocca, J.; Johnstun, J.; Saxe, J.; Jia, J.; Alwala, K. V.; Upasani, K.; Plawiak, K.; Li, K.; Heafield, K.; Stone, K.; and et al. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.

Gao, Z.; Chen, L.; Zhou, J.; and Dai, B. 2025. One-shot Entropy Minimization. arXiv:2505.20282.

Han, X.; Jian, Y.; Hu, X.; Liu, H.; Wang, Y.; Fan, Q.; Ai, Y.; Huang, H.; He, R.; Yang, Z.; and You, Q. 2024. InfiMM-WebMath-40B: Advancing Multimodal Pre-Training for Enhanced Mathematical Reasoning. *CoRR*, abs/2409.12568.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In Vanschoren, J.; and Yeung, S., eds., *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Jain, N.; Han, K.; Gu, A.; Li, W.-D.; Yan, F.; Zhang, T.; Wang, S.; Solar-Lezama, A.; Sen, K.; and Stoica, I. 2024. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. arXiv:2403.07974.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V. V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; Wu, Y.; Neyshabur, B.; Gur-Ari, G.; and Misra, V. 2022. Solving Quantitative Reasoning Problems with Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Li, J.; Beeching, E.; Tunstall, L.; Lipkin, B.; Soletskyi, R.; Huang, S.; Rasul, K.; Yu, L.; Jiang, A. Q.; Shen, Z.; et al. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13: 9.

Li, P.; Skripkin, M.; Zubrey, A.; Kuznetsov, A.; and Oseledets, I. 2025. Confidence Is All You Need: Few-Shot RL Fine-Tuning of Language Models. arXiv:2506.06395.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Liu, J.; Liu, H.; Xiao, L.; Wang, Z.; Liu, K.; Gao, S.; Zhang, W.; Zhang, S.; and Chen, K. 2024. Are Your LLMs Capable of Stable Reasoning? *CoRR*, abs/2412.13147.

Liu, Z.; Chen, C.; Li, W.; Qi, P.; Pang, T.; Du, C.; Lee, W. S.; and Lin, M. 2025. Understanding R1-Zero-Like Training: A Critical Perspective. *CoRR*, abs/2503.20783.

Lv, A.; Xie, R.; Sun, X.; Kang, Z.; and Yan, R. 2025. The Climb Carves Wisdom Deeper Than the Summit: On the Noisy Rewards in Learning to Reason. arXiv:2505.22653.

OLMo, T.; Walsh, P.; Soldaini, L.; Groeneveld, D.; Lo, K.; Arora, S.; Bhagia, A.; Gu, Y.; Huang, S.; Jordan, M.; Lambert, N.; Schwenk, D.; Tafjord, O.; Anderson, T.; Atkinson, D.; Brahman, F.; Clark, C.; Dasigi, P.; Dziri, N.; Guerquin, M.; Ivison, H.; Koh, P. W.; Liu, J.; Malik, S.; Merrill, W.; Miranda, L. J. V.; Morrison, J.; Murray, T.; Nam, C.; Pyatkin, V.; Rangapur, A.; Schmitz, M.; Skjonsberg, S.; Wadden, D.; Wilhelm, C.; Wilson, M.; Zettlemoyer, L.; Farhadi, A.; Smith, N. A.; and Hajishirzi, H. 2025. 2 OLMo 2 Furious. *CoRR*, abs/2501.00656.

OpenAI. 2024a. Hello GPT-4o.

OpenAI. 2024b. Learning to Reason with LLMs.

OpenAI. 2025. Introducing OpenAI o3 and o4-mini.

Paster, K.; Santos, M. D.; Azerbayev, Z.; and Ba, J. 2024. OpenWebMath: An Open Dataset of High-Quality Mathematical Web Text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Prabhudesai, M.; Chen, L.; Ippoliti, A.; Fragkiadaki, K.; Liu, H.; and Pathak, D. 2025. Maximizing Confidence Alone Improves Reasoning. arXiv:2505.22660.

Qwen Team. 2024. QwQ: Reflect deeply on the boundaries of the unknown.

Qwen Team. 2025. QwQ-32B: Embracing the Power of Reinforcement Learning.

Shafayat, S.; Tajwar, F.; Salakhutdinov, R.; Schneider, J.; and Zanette, A. 2025. Can Large Reasoning Models Self-Train? arXiv:2505.21444.

Shao, R.; Li, S. S.; Xin, R.; Geng, S.; Wang, Y.; Oh, S.; Du, S. S.; Lambert, N.; Min, S.; Krishna, R.; Tsvetkov, Y.; Hajishirzi, H.; Koh, P. W.; and Zettlemoyer, L. 2025. Spurious Rewards: Rethinking Training Signals in RLVR. arXiv:2506.10947.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *CoRR*, abs/2402.03300.

Wang, Y.; Yang, Q.; Zeng, Z.; Ren, L.; Liu, L.; Peng, B.; Cheng, H.; He, X.; Wang, K.; Gao, J.; Chen, W.; Wang, S.; Du, S. S.; and Shen, Y. 2025a. Reinforcement Learning for Reasoning in Large Language Models with One Training Example. *CoRR*, abs/2504.20571.

Wang, Z.; Li, X.; Xia, R.; and Liu, P. 2024. MathPile: A Billion-Token-Scale Pretraining Corpus for Math. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Wang, Z.; Zhou, F.; Li, X.; and Liu, P. 2025b. OctoThinker: Mid-training Incentivizes Reinforcement Learning Scaling. arXiv:2506.20512.

Xu, R.; Wang, Z.; Fan, R.; and Liu, P. 2024. Benchmarking Benchmark Leakage in Large Language Models. *CoRR*, abs/2404.18824.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025. Qwen3 Technical Report. *CoRR*, abs/2505.09388.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2024a. Qwen2.5 Technical Report. *CoRR*, abs/2412.15115.

Yang, A.; Zhang, B.; Hui, B.; Gao, B.; Yu, B.; Li, C.; Liu, D.; Tu, J.; Zhou, J.; Lin, J.; Lu, K.; Xue, M.; Lin, R.; Liu, T.; Ren, X.; and Zhang, Z. 2024b. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. *CoRR*, abs/2409.12122.

Yue, Y.; Chen, Z.; Lu, R.; Zhao, A.; Wang, Z.; Yue, Y.; Song, S.; and Huang, G. 2025. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? *CoRR*, abs/2504.13837.

Zhao, A.; Wu, Y.; Yue, Y.; Wu, T.; Xu, Q.; Yue, Y.; Lin, M.; Wang, S.; Wu, Q.; Zheng, Z.; and Huang, G. 2025a. Absolute Zero: Reinforced Self-play Reasoning with Zero Data. *CoRR*, abs/2505.03335.

Zhao, X.; Kang, Z.; Feng, A.; Levine, S.; and Song, D. 2025b. Learning to Reason without External Rewards. arXiv:2505.19590.

Zuo, Y.; Zhang, K.; Qu, S.; Sheng, L.; Zhu, X.; Qi, B.; Sun, Y.; Cui, G.; Ding, N.; and Zhou, B. 2025. TTRL: Test-Time Reinforcement Learning. *CoRR*, abs/2504.16084.