Noisy Data is Destructive to Reinforcement Learning with Verifiable Rewards

Yuxuan Zhu
UIUC
yxx404@illinois.edu

Daniel Kang
UIUC
Bridgewater AIA Labs
ddkang@illinois.edu

Abstract

Reinforcement learning with verifiable rewards (RLVR) has driven recent capability advances of large language models across domains. Recent studies suggest that improved RLVR algorithms allow models to learn effectively from incorrect annotations, achieving performance comparable to learning from clean data. In this work, we show that these findings are invalid because the claimed 100% noisy training data is "contaminated" with clean data. After rectifying the dataset with a rigorous re-verification pipeline, we demonstrate that noisy data is destructive to RLVR. We show that existing RLVR algorithm improvements fail to mitigate the impact of noisy data, achieving similar performance to that of the basic GRPO. Furthermore, we find that the model trained on truly incorrect annotations performs 8–10% worse than the model trained on clean data across mathematical reasoning benchmarks. Finally, we show that these findings hold for real-world noise in Text2SQL tasks, where training on real-world, human annotation errors causes 6–12% lower accuracy than clean data. Our results show that current RLVR methods cannot yet compensate for poor data quality. High-quality data remains essential.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) is a widely used post-training paradigm [10, 26], improving the reasoning capabilities of large language models (LLMs) [18–20, 22, 31, 32, 35, 36]. However, obtaining verifiable annotations at scale for reward calculation is labor-intensive, and ensuring their quality is difficult [5, 33, 37]. Recent studies suggest that data quality is secondary, arguing that models can learn equally effectively from cheaper, noisy data, leveraging algorithmic improvements such as reward shaping [4], objective correction [23], and clipping mechanisms [27]. This raises a fundamental question for the future of post-training: *To what extent can RLVR, with algorithmic improvements, tolerate annotation noise?*

A growing body of recent literature suggests a promising, yet counter-intuitive answer: RLVR appears robust to data noise. Several studies claim that LLMs can learn effectively from low-quality data or even random noise, achieving performance comparable to models trained on clean data [21, 27, 30]. For instance, Shao et al. [30] report that RLVR with 100% incorrect data annotations achieves accuracy on MATH-500 within 5% of the same model trained on clean data. Similarly, Lv et al. [21] suggest that clipped objectives enable learning from random rewards with only 3% performance loss. These findings increasingly imply that data quality is secondary to algorithmic design.

In this work, we argue that these findings are invalid by showing that the noisy data used in prior work is not 100% noisy. Through a rigorous empirical study, we demonstrate that high-quality data remains necessary and cannot be replaced by existing algorithmic improvements.

Re-verifying data reveals severe impact of noise. We identify a critical contamination issue in the training datasets used in prior work. Using a data re-verification pipeline leveraging GPT-5 Pro and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

manual verification (Section 3.3), we find that at least 16% of annotations labeled as "incorrect" in prior work are actually correct (Figure 2). This contamination inflated the performance of models trained on noise. After removing the correct annotations from the prior dataset, we find that training on 100% incorrect annotations degrades MATH-500 accuracy by 9% (8-10% on other benchmarks) compared to clean data (Figure 1). This performance is similar to training with format-only rewards (rewarding \boxed{}), empirically showing that RLVR with pure noise fails to boost reasoning. Furthermore, we show that training on random annotations results in accuracy lower than the base model (Figure 1).

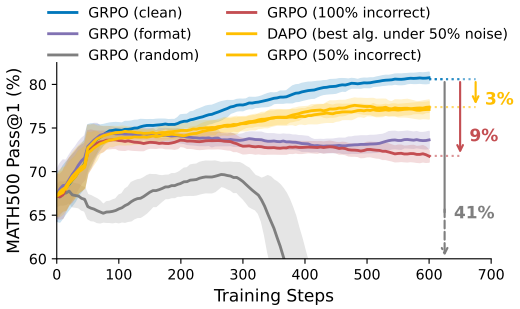


Figure 1: Noisy data significantly degrades RLVR performance, while existing algorithms fail to mitigate it. Training Qwen2.5-Math-7B on 100% incorrect annotations performs similarly to format-only rewards and 9% worse than clean data. Even the best-performing RLVR algorithmic improvements [4, 8, 17, 38, 40] trained on 50% noise only match baseline GRPO [31] and underperform the model trained on clean data.

Existing algorithms fail to compensate for noisy data. We evaluate state-of-the-art algorithmic improvements, including bias mitigation (Dr. GRPO [17], TIS [38], and PGFC [4]), adaptive clipping (SAPO [8]), and dynamic sampling (DAPO [40]), on noisy data. Contrary to prior work, we find that these carefully designed algorithms fail to compensate for true data noise. Under 50% noise, these algorithms perform similarly to GRPO [31] and underperform GRPO with clean data by over 3% on MATH-500 (Figure 1).

Real-world annotation noise is destructive to RLVR. Moving beyond synthetic noise, we use Text2SQL (i.e., translating natural language questions to SQL queries) tasks as a case study to illustrate the impact of real-world annotation errors. Text2SQL datasets contain naturally occurring human annotation errors due to the ambiguity of natural language and the multidimensionality of data. We manually corrected a Text2SQL dataset, BIRD [13], which is known to contain substantial annotation noise [16, 28, 37]. We find that training on the real-world, noisy dataset degrades performance by 6–12% compared to training on our corrected dataset across five base models.

We summarize our contributions as follows:

1. We identify and remove a hidden confound in prior work: correct annotations in a dataset described as 100% noisy. We construct a noisy dataset that enables a controlled empirical study of annotation noise in RLVR.
2. We show that existing algorithmic improvements for RLVR fail to mitigate the impact of noise. With 50% noise, the best-performing algorithm achieves similar accuracy to that of GRPO and 3% lower than clean data.
3. We demonstrate that the impact of data noise is more severe than previously reported. Training with 100% noise leads to similar or lower performance than training with format rewards and 8-10% lower than clean data.
4. Using Text2SQL tasks, we show that the presence of real-world noise is destructive to RLVR. Compared to a cleaned dataset, the real-world, noisy dataset lowers performance by 6–12%.

2 Background

Reinforcement learning with verifiable rewards. RLVR has emerged as a dominant paradigm for post-training LLMs, with recent large-scale successes achieving state-of-the-art results in various domains [10, 26]. Group relative policy optimization (GRPO) is the first algorithmic instantiation of RLVR [31]. In each iteration, for a question-answer pair (q, a), GRPO samples a group of outputs $\{\tau_1, \ldots, \tau_G\}$ from the model $\pi_{\theta_{old}}$ and estimates the advantage of output τ_i as follows:

$$\hat{A}_i = \frac{R(\tau_i, a) - \text{mean}\left(\{R(\tau_j, a)\}_{j=1}^{G}\right)}{\text{std}\left(\{R(\tau_j, a)\}_{j=1}^{G}\right)} \tag{1}$$

Then, GRPO optimizes the model by maximizing the following clipped objective:

$$\mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\left(\min\left(r_{i,t}\hat{A}_i, \text{clip}(r_{i,t}, 1\pm\epsilon)\hat{A}_i\right)\right)\right]$$

where $r_{i,t} = \pi_\theta(\tau_{i,t}|x, \tau_{i,<t})/\pi_{\theta_{old}}(\tau_{i,t}|x, \tau_{i,<t})$.

As shown, advantage estimates $\hat{A}_i$ are the key optimization signals in RLVR, which relies on the quality of annotations a (e.g., gold answers [31] and unit tests [18]). Since prior work assumes the availability of gold annotations [18–20, 31], the impact of noise in a remains underexplored.

The noise robustness hypothesis. Recent literature argues that RLVR is robust to reward noise. Shao et al. [30] observe that models trained with completely incorrect annotations (a) still improve on reasoning benchmarks significantly, up to 24% on MATH-500. They hypothesize that such reasoning improvements shown on benchmarks are due to the reasoning patterns learned in pre-training. Similarly, Lv et al. [21] suggest that RLVR improves LLM despite up to 40% randomly flipped reward signals, achieving a substantial 67% accuracy improvement on MATH-500. These findings increasingly imply the conclusion that RLVR has strong robustness to noise. Our findings challenge the validity of this robustness hypothesis. We demonstrate that the findings of Shao et al. [30] are primarily because their noisy dataset is contaminated by correct annotations (Section 3.3). Furthermore, we argue that Lv et al. [21] rely on a stochastic noise model that misaligns with the practical RLVR that typically uses deterministic, rule-based verifiers.

Mitigation strategies for noisy rewards. To mitigate the biased optimization under noisy data, Cai et al. [4] introduced Policy Gradient Backward Correction (PGBC) and Forward Correction (PGFC), while Mansouri et al. [23] derived a corrected objective to debias the policy gradient. Both theoretical analyses assume that reward corruption is independent of the sampled response conditional on the prompt and clean reward. Cai et al. [4] assume instance-independent noise rates, while Mansouri et al. [23] rely on reward flip rates that are independent of responses. However, real-world annotation errors produce reward flips that depend on the particular response. We diagnose this mismatch for PGFC in Appendix A.7. In Section 5, we empirically show that algorithmic mitigations, including PGFC, fail to provide significant improvements compared to GRPO on noisy data. In addition, experimental evidence from Shao et al. [30] and Park et al. [27] shows that the clipping mechanism can mitigate biased gradients caused by noisy data. In our study, with a truly noisy dataset, we show that even the advanced training stabilization using adaptive clipping [8] and dynamic sampling [40] fail to outperform the baseline GRPO (Section 5).

3 Empirical Study Design

In this section, we introduce the design of our empirical study, starting with three primary research questions that we aim to answer (§3.1). Then, we discuss the types of noise we study (§3.2). Next, we present our method of training data curation for the empirical study (§3.3). Finally, we introduce our implementation details of experiments (§3.4).

3.1 Research Questions

To rigorously evaluate the limits of noise tolerance in RLVR, we must decouple data quality from confounding factors such as base models and algorithm variants. Therefore, we design a comprehensive study spanning *seven* base models, *six* benchmarks, and *five* algorithms besides GRPO. We begin by revisiting prior work's claims with a synthetic noisy dataset, before extending our analysis to real-world noise. Our study centers on three research questions:

RQ1. Does the hypothesis that "RLVR is robust to noise" hold when training data is truly noisy, without being contaminated by correct annotations (Section 4)?

RQ2. Can state-of-the-art algorithmic improvements (e.g., clipping, bias mitigation) effectively recover the performance lost to severe data noise (Section 5)?

RQ3. Do findings from synthetic noise transfer to real-world data? Specifically, how do natural human errors, such as those seen in Text2SQL datasets [37], impact RLVR performance compared with a clean dataset (Section 6)?

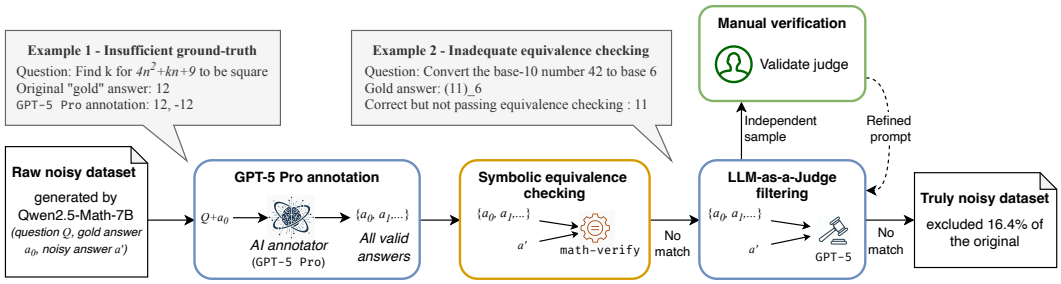


Figure 2: Data re-verification pipeline for synthesizing truly noisy dataset. As shown in the two examples, we identify the issues of insufficient ground-truth annotations and inadequate equivalence checking, and address them via a rigorous pipeline combining LLMs and manual verification.

3.2 Noise Models

According to the RLVR reward formulation (Equation (1)), an accurate reward signal relies on two components: a correct ground-truth annotation a and a reliable equivalence verifier $R(a, \cdot)$. Consequently, noise in RLVR arises from two failures: (1) annotation errors (a is incorrect), or (2) verifier errors (the function $R(\cdot, \cdot)$ is flawed). Based on this categorization, we study four noise models as covered by prior work [4, 21, 23, 30].

Model-generated annotations. To simulate scalable, low-quality data, prior work generates "noisy" annotations using base models [30]. We adopt this as our primary synthetic noise source. In addition, we apply a re-verification pipeline to ensure these annotations are genuinely incorrect, preventing the accidental inclusion of correct labels that invalidate the assumption.

Random annotations. In addition to weak annotations, we study random annotations, a noise model widely used in prior work [24, 42]. For example, we generate a random integer $n \in [1, 1000]$ as the answer to a mathematical problem similar to AIME problems [6]. This simulates a scenario where the data pipeline contains completely mismatched labels, serving as a lower-bound baseline for noise.

Format reward. We use the format reward setting in prior work to simulate a collapsed verifier that checks only for the format of the LLM's generation [30]. In this setting, a model receives a positive reward only for satisfying structural format requirements (e.g., enclosing an answer in `\boxed{}`). We use this noise model to benchmark LLMs that learn format adherence without reasoning.

Real-world noise. In practice, noise is often not purely random. It is systematic and ambiguous. Real-world datasets suffer from incorrect annotations, such as incorrect labels in ImageNet [2], incorrect "gold" queries in BIRD [37], or insufficient test coverage in SWE-Bench [39]. While verifier errors exist [12], they are often fixable engineering issues. In contrast, massive-scale annotation errors are a persistent bottleneck. Thus, the final part of our study (§6) investigates real-world annotation errors.

3.3 Data Curation

Following the noise models, we curate datasets for both controlled, synthetic settings and real-world scenarios. For synthetic noisy data, we follow prior work [4, 21, 23, 30] to use a mathematical reasoning dataset. For real-world noisy data, we use BIRD, a Text2SQL dataset that is known to contain over 50% annotation errors [1, 11, 28, 37].

Synthetic noisy data. We used the same noisy dataset constructed by Shao et al. [30], which comprises incorrect generations of Qwen2.5-Math-7B on the DeepScaleR dataset [20]. To inspect whether this noisy dataset has truly incorrect annotations, we investigated 20 randomly sampled examples and identified a high rate of invalid data. We found that 8/20 (40%) of the examples labeled as "incorrect" were actually correct. There are two primary causes for this data error:

1. *Insufficient ground-truth annotation*: A math problem can have multiple valid answers, while the ground-truth annotation only captures one of them (Example 1 of Figure 2). Consequently, the dataset marks a correct annotation that differs from the ground-truth as incorrect.
2. *Inadequate equivalence checking*: Weak symbolic equivalence verifiers may fail to detect semantic equivalence across differing formats (Example 2 of Figure 2).

To rigorously sanitize the noisy dataset, we propose the following multi-stage data re-verification pipeline, targeting at reducing the number of correct annotations.

1. *GPT-5 Pro Annotation*: To address insufficient ground-truth annotation, we used GPT-5 Pro to find all ground truth for each problem, given the problem text and the original ground-truth annotation.
2. *Symbolic Equivalence Checking*: We used `math-verify` [12] to check the target "incorrect" annotation against the set of correct annotations.
3. *LLM-as-a-Judge Filtering*: To mitigate the issue of inadequate equivalence checking due to different formats, we applied GPT-5 judge to evaluate the "incorrect" annotations that passed the symbolic check. We prompted GPT-5 to determine if each "incorrect" annotation is equivalent to any answer in the set of correct annotations. We attached the prompt in Appendix B.
4. *Manual Investigation*: To validate the reliability of the LLM judge, we use an iterative refinement process. We first manually inspected 100 random samples. When we identified any LLM judge errors, we incorporated the error patterns into the prompt and re-ran the judging process with a new independent sample. In our study, we conducted two iterations of refinement.

We justify the validity of using LLMs in the pipeline. First, while GPT-5 Pro may produce incorrect annotations, our objective is to remove correct annotations from the noisy dataset. Therefore, dropping a data point because it matches an incorrect annotation only reduces dataset size without contaminating the noise. Second, our manual investigation statistically bounds the error rate of LLM judge. In the final sample of 100 examples, we observed a 0% error rate. Modeling this as a binomial proportion, we establish that the true error is upper-bounded at 3% with 95% confidence [25].

In this pipeline, GPT-5 Pro identified 9.0% of initial noisy dataset with insufficient ground-truth annotations. Together with inadequate equivalence checking, we excluded 16.4% of the original noisy data, resulting in a set of 12,769 mathematical problems with incorrect annotations. In addition to this noisy dataset, we construct a clean dataset and a random-annotation dataset.

Real-world noisy data. We select Text2SQL tasks, translating natural language questions to SQL queries, as our target domain for curating real-world noisy dataset, since Text2SQL datasets inherently contain substantial annotation errors due to the semantic ambiguity of natural language and the data-dependent nature of writing SQL queries. Prior studies on Text2SQL datasets have shown that the widely used BIRD dataset [13] contain over 50% incorrect annotations [11, 37]. However, no existing work has curated a clean Text2SQL dataset to serve as a reliable reference for the impact of noise.

To quantify the impact of real-world noisy dataset on RLVR, we curated a clean dataset based on a random sample of 600 noisy Text2SQL instances from the BIRD Train set. For each Text2SQL instance, one of our authors proposed correction while a second independently verified the correction. We repeated this correction-verification pipeline until the verification passes. We resolved disagreement between correction and verification by involving another author. This process identified and corrected 372 (62%) noisy instances. We encountered and resolved two correction-verification conflicts. In the rest of this paper, we denote the original 600 sampled BIRD Train instances as BIRD-Original-600 and the corrected version as BIRD-Corrected-600.

3.4 Implementation

For relatively small-scale models (e.g., Qwen2.5-Math-7B), we used the SkyRL [9]. We conducted these experiments with four A100s, a 64-Core CPU, and 512 GB RAM. For large-scale models (e.g., DeepSeek-V3.1), we used the Tinker, leveraging LoRA to achieve high efficiency [29, 34]. We have submitted our curated datasets and code in the Supplementary Material.

4 Noisy Data Significantly Degrades RLVR Performance

In this section, we investigate the impact of data quality on RLVR based on our synthetic noisy dataset. We first describe the experimental settings and then present our findings.

4.1 Experimental Settings

Training. We used Qwen2.5-Math-7B as our base model and trained it using the GRPO [31]. For reward computation, we used `math-verify` and assigned reward=1 if the generated answer is verified

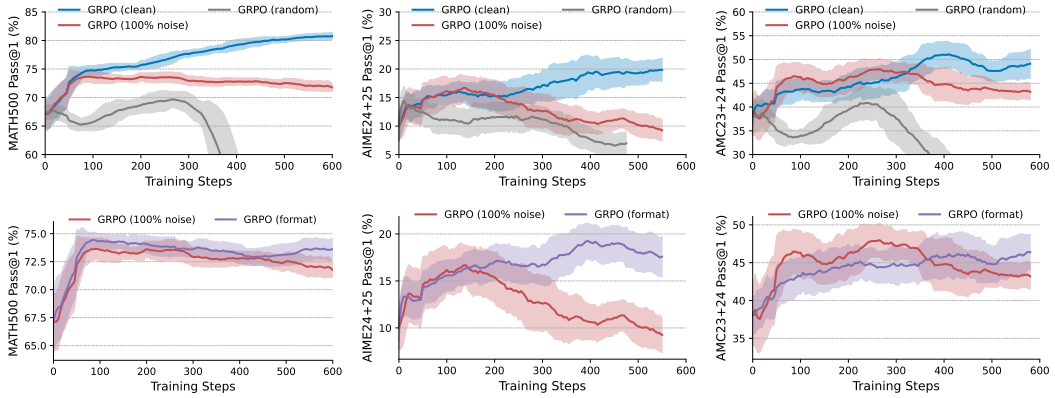


Figure 3: Both incorrect and random annotations significantly decrease the performance of RLVR: RLVR with incorrect annotations achieves comparable or lower accuracy (by 1.2–5.6%) than RLVR with format rewards, and lower accuracy (by 8.5–10.0%) than RLVR with clean data; RLVR with random annotations achieves significantly lower accuracy (by 6.7–27.8%) than the base model.

to match the solution and reward=0 otherwise. In the setting of format reward, we assign reward=1 if the generated sequence contains \boxed{} and reward=0 otherwise. We set the batch size to 64, the group size to 16, and the learning rate to 5×10^{-6}, while keeping other settings (including the number of epochs) consistent with the configuration in Shao et al. [30]. For the clean, noisy, and format-reward settings, we trained models for three epochs, as we observed no further improvements beyond this point (Appendix C). For the random annotation setting, we used early stopping at two epochs as our pilot runs indicated significant performance collapse at the end of the second epoch.

We also trained Nemotron 3 Nano [3] on the same dataset at 0%, 50%, and 100% annotation noise. For this backbone, we used Tinker with LoRA rank 32 and a learning rate of 5×10^{-5}, keeping the remaining experimental settings consistent with the Qwen experiments.

Evaluation. We performed pass@1 evaluations with a temperature of 0 and pass@k (k$\geq$2) evaluations with a temperature of 1. We include five benchmarks for evaluation: MATH-500 [14], AIME 2024 and 2025 [6], AMC 2023 and 2024 [7]. Among them, the AIME 2025 and AMC 2024 datasets were released after the last update date of the model weights of our base model, ensuring a contamination-free evaluation environment.

4.2 Findings

Answer to RQ1. *Noisy data impacts learning significantly, resulting in 8-10% lower accuracy than training on clean data and failing to outperform training with pure format rewards.*

Noisy data causes significant performance degradation. We quantify the performance gap between training on clean data and training on noisy data. As shown in Figure 3, training on 100% noise yields a substantially lower performance compared to the clean baseline, degrading accuracy by 9.0% on MATH-500, 10.0% on AIME, and 8.5% on AMC. We also find that training on 100% noise fails to outperform the format-reward baseline, suggesting that noisy data does not improve the capability of the base model. Furthermore, training on random annotations leads to catastrophic collapse, causing performance to drop significantly below that of the base model.

We attribute the accuracy degradation of the model trained on noisy data to a confirmation bias loop during optimization. As detailed in the training dynamics analysis in Appendix A.1, we find that the model easily exploits its pre-existing flawed priors, achieving a 149% higher average reward on noisy data while exhibiting a 9% drop in policy entropy, rather than exploring correct reasoning paths.

Noise sensitivity extends to Nemotron. In Table 1, we show the accuracy of fine-tuned Nemotron models on the five math benchmarks. We find that the average accuracy drops from 69.3% with clean data to 59.9% with 50% noise and 35.8% with 100% noise. These results demonstrate that the impact of noisy data extends beyond Qwen models.

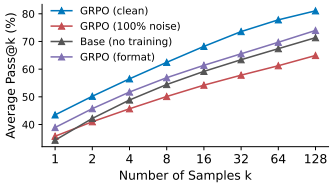


Figure 4: Noise leads to lower pass@k than the base model at larger tested sampling budgets, indicating lower empirical problem coverage.

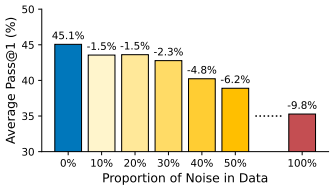


Figure 5: Noise results in increasing performance degradation (1.5–9.8%) as the noise proportion increases.

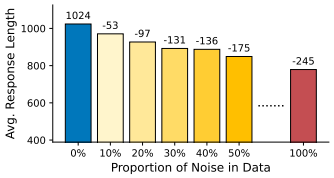


Figure 6: Training on noisy data results in shorter responses (5.2–23.9%). Response length is a secondary behavioral diagnostic.

Noisy data decreases pass@k. To determine if noisy data affects the range of problems a model can solve given multiple attempts, we evaluate the pass@k of models trained on noisy data. We find that clean data increases empirical problem coverage within the evaluated sampling budgets, while noisy data reduces it [41]. As shown in Figure 4, the model trained on clean data consistently outperforms all other baselines by 4.6–16.6% in terms of pass@k. However, the model trained on noisy data outperforms the base model only when $k = 1$, while underperforming the base model by up to 6.1% when $k > 1$.

Table 1: Nemotron-3-Nano-30B-A3B accuracy (%) under annotation noise.

Benchmark	100% noise	50% noise	0% noise
AIME 2024	11.3	39.3	50.7
AIME 2025	10.0	26.7	46.0
AMC 2023	54.5	83.5	91.5
AMC 2024	28.9	61.3	65.8
MATH-500	74.4	88.6	92.4
Average	35.8	59.9	69.3

Noisy data increasingly leads to shorter responses. For the fine-tuned Qwen2.5-Math-7B models, we measure average response length in tokens across the five math benchmarks. We find that training on noisy data leads to a significant reduction in generation length. In Figure 6, we show that models trained on noisy data produced responses that were 5.2% to 23.9% shorter than the model trained on clean data, indicating that noisy data discourages complex reasoning chains.

Even modest annotation noise degrades RLVR. We vary the proportion of incorrect annotations from 10% to 100% used to train Qwen2.5-Math-7B with GRPO and calculate average evaluation accuracy over the five math benchmarks. In Figure 5, we show a monotonic drop in accuracy as noise increases, with degradation ranging from 1.5% (at 10% noise) to 9.8% (at 100% noise). These results show that even datasets with modest annotation noise reduce RLVR accuracy, with the performance penalty increasing as incorrect annotations become more prevalent.

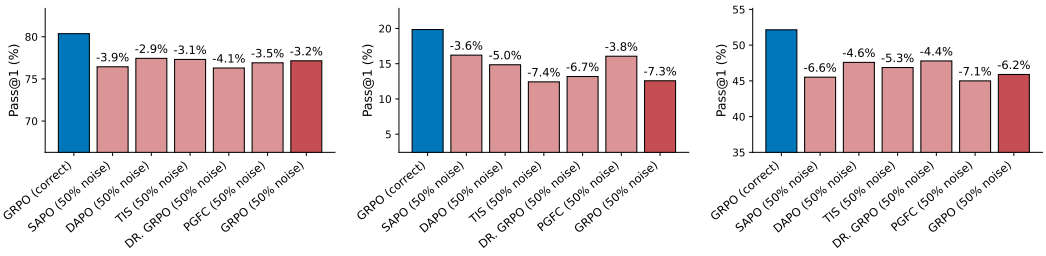
5 Algorithm Improvements Fail to Mitigate the Impact of Data Noise

In this section, we investigate whether existing algorithm improvements mitigate the impact of noisy data on RLVR. We first introduce our experimental settings and then discuss our findings.

5.1 Algorithm Selection

We evaluated a set of state-of-the-art RLVR variants. We categorized these methods into three algorithmic changes that hypothetically improve stability and robustness:

1. **Gradient bias mitigation**: GRPO suffers from bias caused by inherent policy loss [17], training-inference shifts [38], and noisy data [4]. Therefore, we include Dr. GRPO [17], truncated importance sampling (TIS) [38], and policy gradient with forward correction (PGFC) designed to address this bias, respectively.
2. **Dynamic sampling**: GRPO suffers from vanishing gradients when a rollout group has uniform rewards (all correct or all wrong). To address this, prior work proposed dynamic sampling, which discards uniform groups to focus on informative groups. Dynamic sampling may implicitly



(a) MATH-500. (b) AIME 2024 and 2025. (c) AMC 2023 and 2024.

Figure 7: Existing algorithmic improvements fail to mitigate the impact of noisy data. Under 50% noise, none of the improved algorithms achieved a $> 2\%$ accuracy improvement on all benchmarks simultaneously compared to GRPO with 50% noise, while all algorithms underperform the vanilla GRPO with clean data by 3.1–7.3%.

filter out failures caused by noise. Thus, we evaluate TIS [38] and DAPO [40], both of which incorporate dynamic sampling.

3. **Clipping mechanisms**: Prior research suggests that asymmetric clipping can improve learning from noisy data by regulating entropy [21]. Furthermore, adaptive clipping ratios based on group advantage can stabilize gradient updates when a batch of rollouts exhibits a variance of rewards. Therefore, we assess DAPO (asymmetric clipping) [40] and SAPO (adaptive clipping) [8].

In summary, we evaluate five algorithms to achieve a full coverage of the categories of state-of-the-art algorithmic improvements: Dr. GRPO (bias mitigation), TIS (bias mitigation + dynamic sampling), PGFC (bias mitigation), DAPO (dynamic sampling + clipping), and SAPO (clipping).

5.2 Experimental Settings

In addition to the settings of Section 4, we use the default hyperparameter recommendations for each algorithm based on their original publications. For TIS and DAPO, we used asymmetric clipping with an upper ratio of 0.28 and a lower ratio of 0.2, alongside an overlong buffer threshold of 2,048 tokens and a penalty factor of 1.0. Additionally, for TIS, we applied an importance sampling cap of 2.0. For SAPO, we used the recommended soft clipping temperatures of $\tau_+ = 1.0$ and $\tau_- = 1.05$. For PGFC, we set the false-negative correction parameter to 0.50, the dataset's annotation-noise fraction. In Appendix A.8, we report diagnostics showing that the intended mechanism of each method is active under these recommended settings.

5.3 Results

Answer to RQ2. *Algorithmic improvements fail to compensate for the noise in data. None of the SOTA variants (TIS, DAPO, SAPO, Dr. GRPO) outperformed GRPO by more than 1% under the 50% noise. None of them consistently achieved performance within 5% of GRPO on clean data.*

Algorithmic improvements fail to mitigate accuracy degradation. In Figure 7, we summarize the final accuracy achieved by each algorithm. Under 50% noise, none of the evaluated algorithms achieved a consistent improvement over the GRPO (50% noise) baseline. Specifically, no algorithm yielded more than a 2% accuracy gain across all benchmarks simultaneously. Furthermore, all algorithmic variants underperform the model trained on clean data, with accuracy gaps ranging from 3.1% to 7.3%. This suggests that current algorithmic interventions are insufficient to compensate for the destructive effects of noise. We defer a detailed analysis of the accuracy trend to Appendix A.4, which shows similar trends to Figure 7.

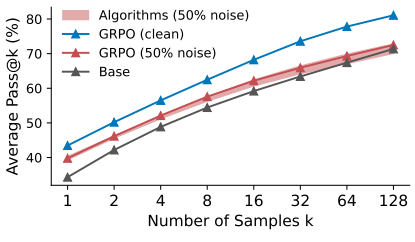


Figure 8: The evaluated algorithms have similar pass@k to GRPO under 50% noise and remain below clean GRPO.

Algorithmic improvements fail to recover pass@k. To determine if algorithmic improvements could enable the models to solve a wider range of problems given multiple attempts, we analyzed the average pass@k performance of these methods across all benchmarks. In Figure 8, we demonstrate that the performance of the improved algorithms (represented by the shaded red region) overlaps with the GRPO (50% noise) baseline. None of the algorithmic improvements consistently achieved a higher pass@k score than the vanilla baseline, and all remained significantly below the GRPO with clean data. Additionally, we analyzed the response length of the trained models. We found that none of the models trained via improved algorithms consistently generates longer responses than GRPO. We defer more detailed results and analysis to Appendix A.5 and A.6.

6 Impact of Real-world Annotation Noise

Moving beyond synthetic noisy data, we extend our analysis to the annotation errors found in real-world datasets. Specifically, we investigate the impact of naturally occurring annotation errors within the BIRD dataset [13] on RLVR performance.

6.1 Experimental Settings

Training. We applied multi-turn RLVR with iterative query refinement [15], and define a tri-level reward that yields 1 if the execution results of the generated SQL and the gold SQL match, -1 if the output misses ⟨solution⟩ tags, and 0 otherwise. We trained each model for ten epochs (Appendix C) with a batch size of 64, group size of 16, learning rate of 5×10^{-5}, and LoRA rank of 32.

Evaluation. Existing Text2SQL benchmarks suffer from substantial annotation errors [11, 16, 28, 37]. To ensure our reported performance differences come from noisy training data rather than benchmark errors, we used the BIRD Mini-Dev set independently verified by Arcwise [1]. This benchmark contains corrections for 161 (32.3%) of the original 498 instances. After independent, manual inspection by two of our authors, we find that the corrections by Arcwise are valid and sound. We performed all evaluations using greedy decoding (temperature=0).

6.2 Findings

Answer to RQ3. *Real-world noisy data is destructive to RLVR. Across different base models ranging from 32B to 685B parameters, training on noisy data results in up to 12% lower accuracy compared to training on clean data.*

Real-world noise leads to significant degradation. As shown in Figure 9, we find that training on BIRD-Corrected-600 demonstrates consistent and significant superiority over training on BIRD-Original-600. Models trained on the BIRD-Corrected-600 consistently outperform those trained on the original noisy data, with performance gaps ranging from a minimum of 5.7% to a maximum of 12.1%. This confirms that the impact of noisy data observed in synthetic settings transfers directly to realistic domains.

Algorithmic mitigation fails to handle real-world noise. We also tested whether PGFC [4], an algorithm specifically designed to mitigate impact of noise on RLVR, could recover the performance lost to real-world noise. Consistent with our findings from synthetic noise, we show that PGFC failed to yield consistent improvements over the GRPO. On GPT-OSS-120B-A5B, PGFC performed significantly worse than GRPO on noisy data. Overall, models trained with PGFC on BIRD-Original-600 remained 5.1-16.3% less accurate than those trained on clean data.

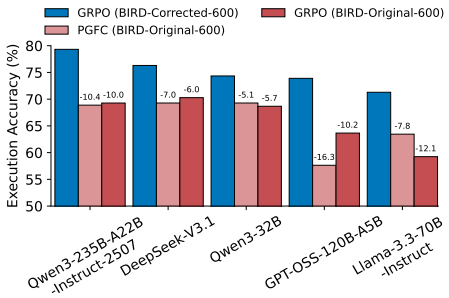


Figure 9: Real-world noise significantly degrades RLVR performance. Models trained on the noisy BIRD dataset (BIRD-Original-600) via GRPO results in 5.7–12.1% lower accuracy than BIRD-Corrected-600. PGFC [4], an algorithm designed for noisy data, fails to consistently improve over GRPO and remains 5.1–16.3% behind the clean data.

7 Conclusion

In this work, we demonstrate the significant impact of noisy data on RLVR. Contrary to prior studies, we developed a rigorous data re-verification pipeline to construct a set of truly noisy data and investigated real-world noise by correcting a subset of the noisy BIRD dataset. Empirically, we identify a consistent performance collapse of 8–10% due to synthetic noise and 6–12% due to real-world noise. We show that existing RLVR algorithms result in performance indistinguishable from the basic GRPO. These findings highlight that RLVR risks yielding suboptimal models without high-quality data, which has not yet been addressed by existing algorithmic interventions.

8 Limitations and Impact Statement

By demonstrating that algorithmic interventions fail against noisy RLVR data, we highlight the critical necessity of high-quality data curation. To prevent deploying degraded models in high-stakes domains like science and code generation, researchers must prioritize data quality over algorithmic complexity. A primary limitation of our study is its focus on outcome-based rewards and deterministic tasks (math and Text2SQL). Future work could investigate whether this noise sensitivity persists in open-ended domains or when using process-based reward models that verify intermediate steps.

Acknowledgments and Disclosure of Funding

This research was supported in part by a sponsorship from Bridgewater AIA Labs.

References

[1] Arcwise. Bird minidev - corrections, 2025. URL `https://docs.google.com/spreadsheets/d/1IGm9Otruey60ujUnl8AOkepY3qgWHdFJHnX7hQGUeCw`. Accessed: 2025-09-15.

[2] Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and Oord, A. v. d. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

[3] Blakeman, A., Grattafiori, A., Basant, A., Gupta, A., Khattar, A., Renduchintala, A., Vavre, A., Shukla, A., Bercovich, A., Ficek, A., et al. Nemotron 3 nano: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2512.20848*, 2025.

[4] Cai, X.-Q., Wang, W., Liu, F., Liu, T., Niu, G., and Sugiyama, M. Reinforcement learning with verifiable yet noisy rewards under imperfect verifiers. *arXiv preprint arXiv:2510.00915*, 2025.

[5] Chen, A., Li, A., Gong, B., Jiang, B., Fei, B., Yang, B., Shan, B., Yu, C., Wang, C., Zhu, C., et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

[6] Community, A. O. Aime problems and solutions, 2025. URL `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`. Accessed Dec. 10, 2025.

[7] Community, A. O. Amc problems and solutions, 2025. URL `https://artofproblemsolving.com/wiki/index.php?title=AMC_Problems_and_Solutions`. Accessed Dec. 10, 2025.

[8] Gao, C., Zheng, C., Chen, X.-H., Dang, K., Liu, S., Yu, B., Yang, A., Bai, S., Zhou, J., and Lin, J. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

[9] Griggs, T., Hegde, S., Tang, E., Liu, S., Cao, S., Li, D., Ruan, C., Moritz, P., Hakhamaneshi, K., Liaw, R., Malik, A., Zaharia, M., Gonzalez, J. E., and Stoica, I. Evolving skyrl into a highly-modular rl framework, 2025. Notion Blog.

[10] Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[11] Jin, T., Choi, Y., Zhu, Y., and Kang, D. Pervasive annotation errors break text-to-sql benchmarks and leaderboards. *VLDB 2026*, 2026.

[12] Kydlíček, H., Gandenberger, G., and Maksin, L. Math verify, 2025. URL https://github.com/huggingface/Math-Verify. Accessed Dec. 10, 2025.

[13] Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Geng, R., Huo, N., et al. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36:42330–42357, 2023.

[14] Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *ICLR*, 2023.

[15] Liu, S., Hegde, S., Cao, S., Zhu, A., Li, D., Griggs, T., Tang, E., Malik, A., Hakhamaneshi, K., Liaw, R., Moritz, P., Zaharia, M., Gonzalez, J. E., and Stoica, I. Skyrl-sql: Matching gpt-4o and o4-mini on text2sql with multi-turn rl, 2025. Notion Blog.

[16] Liu, X., Shen, S., Li, B., Tang, N., and Luo, Y. Nl2sql-bugs: A benchmark for detecting semantic errors in nl2sql translation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5662–5673, 2025.

[17] Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[18] Luo, M., Jain, N., Singh, J., Tan, S., Patel, A., Wu, Q., Ariyak, A., Cai, C., Venkat, T., Zhu, S., Athiwaratkun, B., Roongta, M., Zhang, C., Li, L. E., Popa, R. A., Sen, K., and Stoica, I. Deepswe: Training a state-of-the-art coding agent from scratch by scaling rl. https://pretty-radio-b75.notion.site/DeepSWE-Training-a-Fully-Open-sourced-State-of-the-Art-Coding-Agent-by-Scaling-RL-22281902c1468193aabbe9a8c59bbe33, 2025. Notion Blog.

[19] Luo, M., Tan, S., Huang, R., Patel, A., Ariyak, A., Wu, Q., Shi, X., Xin, R., Cai, C., Weber, M., Zhang, C., Li, L. E., Popa, R. A., and Stoica, I. Deepcoder: A fully open-source 14b coder at o3-mini level. https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51, 2025. Notion Blog.

[20] Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Li, L. E., Popa, R. A., and Stoica, I. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2, 2025. Notion Blog.

[21] Lv, A., Xie, R., Sun, X., Kang, Z., and Yan, R. The climb carves wisdom deeper than the summit: On the noisy rewards in learning to reason. *arXiv preprint arXiv:2505.22653*, 2025.

[22] Ma, X., Liu, Q., Jiang, D., Zhang, G., Ma, Z., and Chen, W. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*, 2025.

[23] Mansouri, O. E., Seddik, M. E. A., and Lahlou, S. Noise-corrected grpo: From noisy rewards to unbiased gradients. *arXiv preprint arXiv:2510.18924*, 2025.

[24] Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

[25] Newcombe, R. G. Two-sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in medicine*, 17(8):857–872, 1998.

[26] OpenAI. Introducing openai o3 and o4-mini, 2025. URL https://openai.com/index/introducing-o3-and-o4-mini/. Accessed Dec. 10, 2025.

[27] Park, J. R., Kim, J., Kim, G., Jo, J., Choi, S., Cho, J., and Ryu, E. K. Clip-low increases entropy and clip-high decreases entropy in reinforcement learning of large language models. *arXiv preprint arXiv:2509.26114*, 2025.

[28] Pourreza, M. and Rafiei, D. Evaluating cross-domain text-to-sql models and benchmarks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1601–1611, 2023.

[29] Schulman, J. and Lab, T. M. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

[30] Shao, R., Li, S. S., Xin, R., Geng, S., Wang, Y., Oh, S., Du, S. S., Lambert, N., Min, S., Krishna, R., et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

[31] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[32] Su, Y., Yu, D., Song, L., Li, J., Mi, H., Tu, Z., Zhang, M., and Yu, D. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*, 2025.

[33] Team, K., Bai, Y., Bao, Y., Chen, G., Chen, J., Chen, N., Chen, R., Chen, Y., Chen, Y., Chen, Y., et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

[34] Team, T. M. Tinker, 2025. URL `https://thinkingmachines.ai/tinker/`. Accessed Dec. 10, 2025.

[35] Wei, Y., Duchenne, O., Copet, J., Carbonneaux, Q., Zhang, L., Fried, D., Synnaeve, G., Singh, R., and Wang, S. I. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025.

[36] Wen, X., Liu, Z., Zheng, S., Ye, S., Wu, Z., Wang, Y., Xu, Z., Liang, X., Li, J., Miao, Z., et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

[37] Wretblad, N., Riseby, F., Biswas, R., Ahmadi, A., and Holmström, O. Understanding the effects of noise in text-to-sql: An examination of the bird-bench benchmark. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 356–369, 2024.

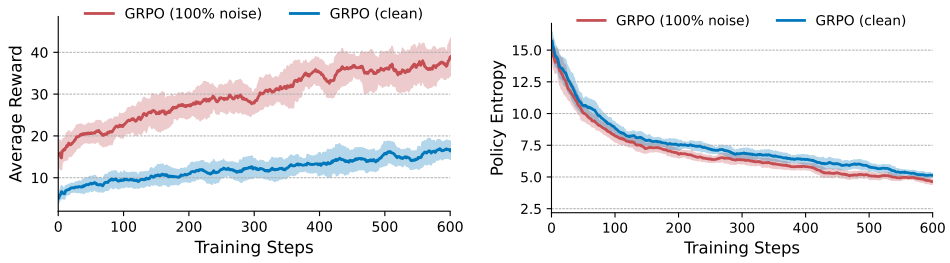
[38] Yao, F., Liu, L., Zhang, D., Dong, C., Shang, J., and Gao, J. Your efficient rl framework secretly brings you off-policy rl training, August 2025. URL `https://fengyao.notion.site/off-policy-rl`.

[39] Yu, B., Zhu, Y., He, P., and Kang, D. Utboost: Rigorous evaluation of coding agents on swe-bench. *ACL 2025*, 2025.

[40] Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[41] Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

[42] Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.



(a) Average reward.

(b) Policy entropy.

Figure 10: Training dynamics of RLVR under clean versus 100% noisy data. (a) The model achieves a 149% higher average reward when trained on noisy data, indicating that it easily exploits and satisfies its own flawed priors. (b) Training on noise induces a more rapid decline in policy entropy (9% lower on average), reflecting premature convergence and diminished exploration.

A Detailed Experimental Results

A.1 Training Dynamics

To understand the mechanism behind the severe performance degradation caused by noisy data, we analyze the training dynamics of the model. Specifically, we compare the average reward and policy entropy curves of the GRPO algorithm on 100% verified noise versus clean data (Figure 10).

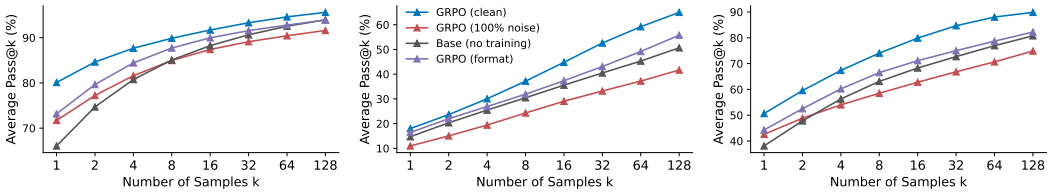
Reward Exploitation on Flawed Priors. The model finds it significantly easier to optimize for incorrect annotations than for correct ones. As shown in Figure 10a, during training, the model achieves 149% higher average rewards on noisy data compared to clean data. Because the synthetic noise in the dataset originates from the base model's own incorrect generations, these flawed reasoning paths already possess a high initial probability under the model's pre-trained distribution. Consequently, the model effortlessly rediscovers and satisfies these incorrect labels, exploiting its own flawed priors rather than learning novel, rigorous reasoning chains.

Premature Entropy Collapse. This ease of optimization actively harms the exploration necessary for RLVR. In Figure 10b, we illustrate the evolution of policy entropy, which serves as a proxy for the model's exploratory behavior. We observe that training on noise results in a consistently lower policy entropy by 9% on average, compared to training on clean data. The rapid collapse in entropy indicates that the model becomes overconfident in its generated outputs much earlier in the training process, prematurely curtailing its search space.

The Confirmation Bias Loop. Together, these dynamics reveal that applying RLVR to model-generated noise creates a pathological "confirmation bias loop." Because the model is heavily and easily rewarded for reasoning paths it already favors, it rapidly loses the incentive to explore alternative, correct solutions. Instead of expanding the model's reasoning capabilities, RLVR with noisy data merely reinforces and entrenches the model's pre-existing errors, leading to the collapsed reasoning lengths and degraded accuracy observed in our evaluations.

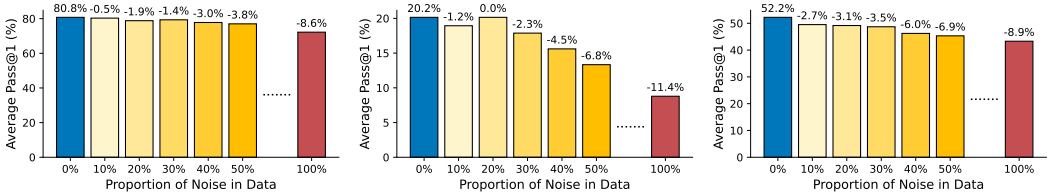
A.2 Pass@k Results of Models Trained on Noisy Data

We show the pass@k performance of models trained on noisy data in Figure 11. While the model trained on correct annotations (blue) consistently expands the solution space across all k, the model trained on 100% noise (red) exhibits failures consistently. On easier benchmarks like MATH-500 (Figure 11a), the noisy model performs comparably to the base model at $k = 1$, but falls behind as k increases. This trend is even more severe on complex reasoning benchmarks like AIME (Figure 11b) and AMC (Figure 11c), where the base model (grey) consistently outperforms the noisy model across nearly all values of k.



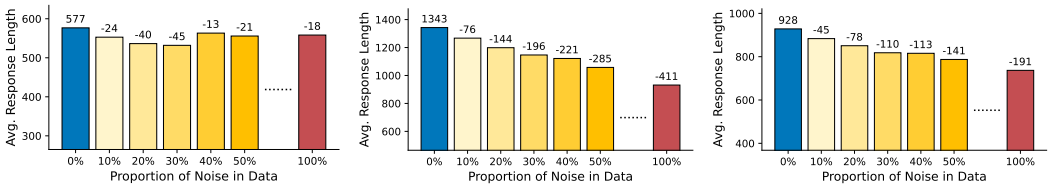
(a) MATH-500. (b) AIME 2024 and 2025. (c) AMC 2023 and 2024.

Figure 11: Noise leads to lower pass@k than the base model at larger tested sampling budgets, indicating lower empirical problem coverage within those budgets.



(a) MATH-500. (b) AIME 2024 and 2025. (c) AMC 2023 and 2024.

Figure 12: Training on noisy data results in increasing performance degradation (by up to 10.0%) as the noise proportion increases.



(a) MATH-500. (b) AIME 2024 and 2025. (c) AMC 2023 and 2024.

Figure 13: Training on noisy data results in shorter responses.

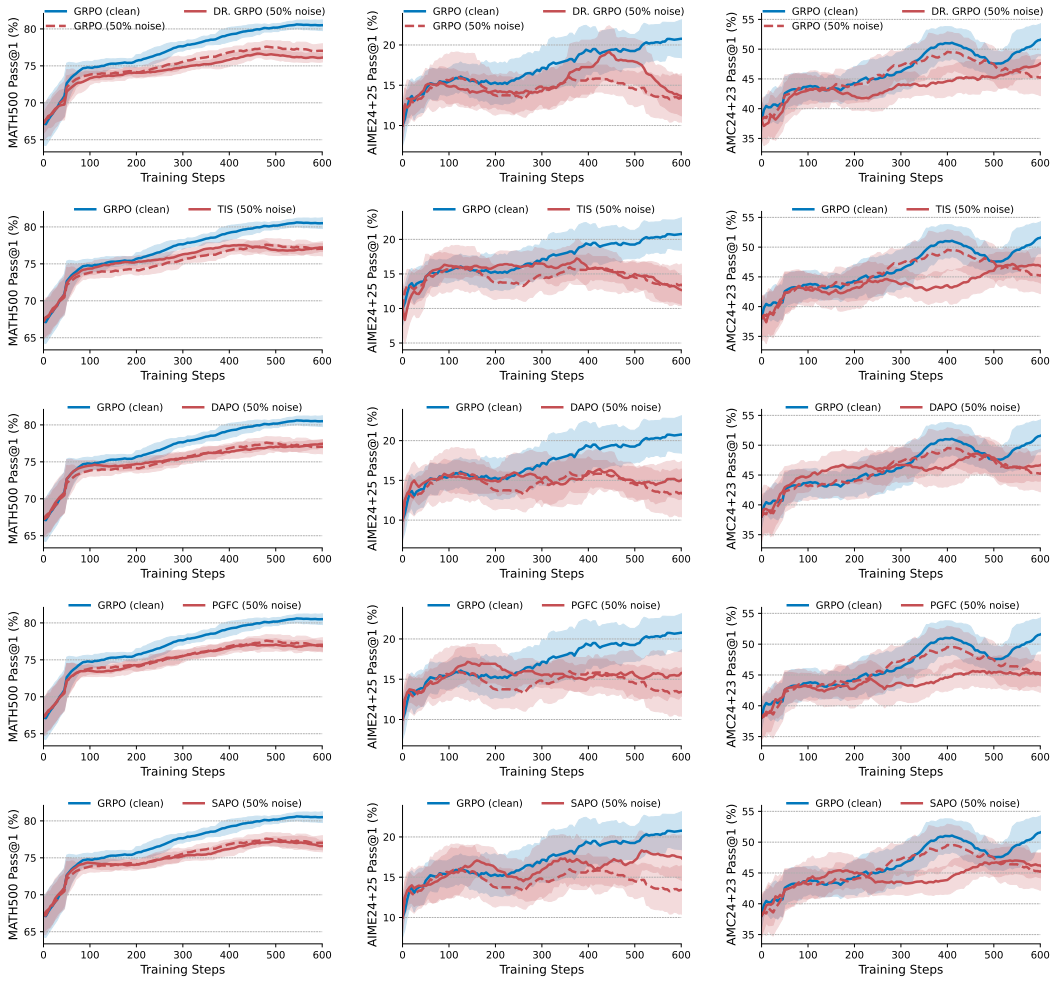
A.3 Results of Training on Different Noise Rates

We show the performance of models trained on data with different noise rates in Figure 12. As shown, we find a strict negative correlation: as the proportion of incorrect annotations rises, accuracy consistently declines. This degradation is most severe on the challenging AIME benchmark (Figure 12b), where 100% noise precipitates a 10.0% collapse in accuracy compared to the clean baseline. Similarly, MATH-500 and AMC suffer significant drops of 9.0% and 8.5%, respectively. Crucially, we observe no "safe" noise threshold; even moderate noise levels (e.g., 20–30%) result in measurable performance penalties, confirming that RLVR lacks intrinsic robustness to label noise.

We show the average generation length of models trained on data with different noise rates in Figure 13. We reveal a clear inverse relationship: as noise increases, responses become progressively shorter. This effect is most profound on the complex AIME benchmark (Figure 13b), where we find a steep decline from 1343 tokens (clean data) to just 932 tokens (100% noise), a loss of over 400 tokens or $\approx$30% of the response length. Even minor contamination (10% noise) causes a noticeable drop of 76 tokens.

A.4 Results of Different RLVR Algorithms

In Figure 14, we show the test accuracy trajectories (pass@1) over all training steps for all algorithms. We first show a sharp contrast between clean and noisy supervision. Across all five algorithmic variants (rows) and three benchmarks (columns), the model trained on clean data establishes a clear performance upper bound, maintaining a steady upward trajectory. In contrast, the advanced algorithms (solid red) fail to differ from the GRPO baseline trained on the same noisy data (dashed



(a) MATH-500. (b) AIME 2024 and 2025. (c) AMC 2023 and 2024.

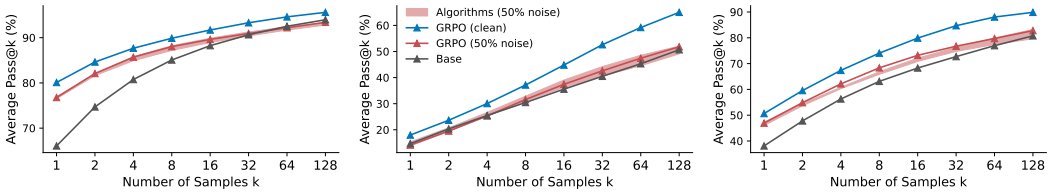
Figure 14: Existing algorithm improvements fail to mitigate the impact of noisy data. Under 50% noise, none of the improved algorithm achieved $> 2\%$ accuracy improvement on all benchmarks simultaneously compared to GRPO with 50% noise, while all algorithms underperform the vanilla GRPO with clean data by 3.1-7.3%

red). The trajectories are nearly indistinguishable, with overlapping confidence intervals throughout the process.

Specifically, on the challenging AIME benchmark (middle column), both the advanced algorithms and the vanilla noisy baseline exhibit signs of training collapse after 400 steps, whereas the clean model continues to improve. This confirms that current algorithmic state-of-the-art methods, including Dr. GRPO, TIS, DAPO, PGFC, and SAPO, offer no significant robustness against validated label noise.

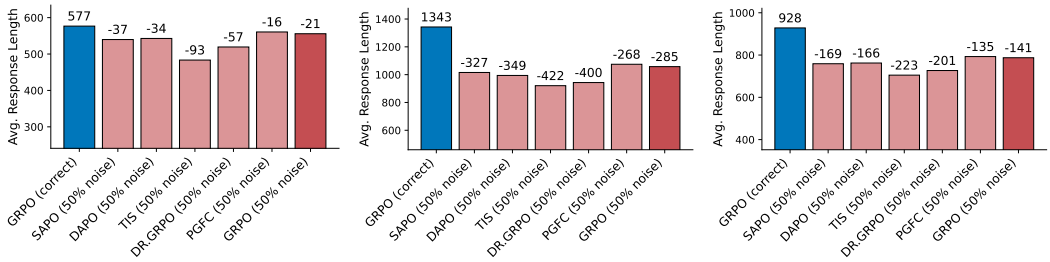
A.5 Pass@k Results of Different RLVR Algorithms

In Figure 15, we show the pass@k results of models trained via improved algorithms. We find that the shaded region (range of improved algorithms) effectively collapses onto the GRPO curve (50% noise), indicating that no algorithmic intervention successfully expands the set of solvable problems beyond the naive baseline. As sampling budget k increases, the gap between these algorithms and the clean-data model (blue) widens, particularly on the rigorous AIME benchmark (Figure 15b). This demonstrates that while these algorithms might stabilize training metrics, they fail to repair the fundamental damage noise inflicts on the model's ability to explore and find correct solutions.



(a) MATH-500. (b) AIME 2024 and 2025. (c) AMC 2023 and 2024.

Figure 15: The evaluated algorithms have similar pass@k to GRPO under 50% noise and remain below clean GRPO across the tested sampling budgets.



(a) MATH-500. (b) AIME 2024 and 2025. (c) AMC 2023 and 2024.

Figure 16: Training on noisy data with improved algorithms still results in shorter responses.

A.6 Generation Length Analysis of Different RLVR Algorithms

In Figure 16, we show the average response length across all trained models. We demonstrate a significant contraction in reasoning depth when noise is introduced. On the complex AIME benchmark (Figure 16b), the model trained on clean data generates long, detailed chains of thought (avg. 1343 tokens). In contrast, models trained on noisy data, regardless of the algorithm used, exhibits a massive reduction in length, dropping by over 300 tokens. This suggests that noisy rewards penalize complex exploration, encouraging the model to converge on shorter reasoning paths. Specifically, advanced algorithms like TIS and DAPO fail to reverse this trend, often producing responses even shorter than the GRPO baseline. This confirms that current algorithmic improvements do not restore the depth of thought characteristic of models trained on high-quality data.

A.7 PGFC under Annotation Noise

PGFC's directional-alignment guarantee assumes an instance-independent reward channel [4]. In a realistic dataset, each prompt q has a fixed annotation $\tilde{a}(q)$, which is either correct or incorrect. The clean and observed rewards, $R(\tau, a(q))$ and $R(\tau, \tilde{a}(q))$, differ according to both the prompt and the sampled response τ. This structure falls outside the assumed reward channel.

We re-score rollouts from the Qwen2.5-Math-7B PGFC run against both clean and 50%-noisy annotations. Among responses accepted under the clean annotations, the fraction rejected under the noisy annotations rises from 0.43 over the first 100 training steps to 0.59 over the final 100 steps. The fixed correction parameter of 0.50 therefore does not track the observed false-negative rate throughout training. PGFC's guarantee does not apply to this annotation-noise setting.

A.8 Algorithm Diagnostics

We check whether the intended mechanisms are active under these settings. For TIS, the mean absolute token-probability difference between the FSDP2 learner and vLLM sampler ranges from 0.0010 to 0.0016. DAPO maintains 15.8–22.0% higher mean policy entropy than GRPO. SAPO reduces the standard deviation of accuracy over the final 100 training steps by 27.2% relative to GRPO. Dr. GRPO introduces no additional hyperparameters: we remove sequence-length and advantage

standard-deviation normalization. These diagnostics show that the methods' intended mechanisms are active under the recommended settings.

B Prompts Used in the Re-verification Pipeline

We used the following prompt when acquiring annotations from GPT-5 Pro.

```
Question: {question}
If the question have multiple distinct and correct answers, provide all the
answers. Write the answers in LaTeX format. Do not include any explanations.
Answer(s):
```

We used the following prompt as the final prompt for judging the correctness of answers with GPT-5.

```
You are a highly intelligent and accurate math grader. Given a math question,
a ground truth answer, and a student's answer, determine whether the student's
answer is correct. You must follow the following guidelines:

1. If the question has multiple distinct but correct answers, the student only
needs to provide one of them to be considered correct. For example, if the
ground truth answer is "(k in 12, -12)" and the student's answer is "11", the
student's answer should be considered correct.

2. Ignore any format mistakes and only grade the mathematical meaning of the
answer. For example, if the ground truth answer is "(11)_6" and the student's
answer is "11", the student's answer should be considered correct. Another
example is if the ground truth answer is latex formatted and the student's answer
is not latex formatted but is mathematically equivalent, the student's answer
should be considered correct. Vice versa is also true.

3. If the student's answer is mathematically equivalent to the ground truth
answer, it should be considered correct.

4. If the student's answer uses fractions, decimals, or different
representations that are mathematically equivalent to the ground truth answer,
it should be considered correct. For example, if the ground truth answer is
"4\sqrt{2}" and the student's answer is "5.656854249492381", the student's answer
should be considered correct.

5. If the question is an Yes/No question and the ground truth answer is an
exact solution, the student's answer is correct as long as they answers "Yes"
or anything equivalent to "Yes". For example, for the question "Does there exist
a fraction equivalent to $\frac{7}{13}$ such that the difference between the
denominator and the numerator is 24?", the ground truth answer is "\frac{28}{52}"
and the student's answer is "Yes", the student's answer should be considered
correct.

6. If the student's answer is a simplification of the ground truth answer, it
should be considered correct. For example, if the ground truth answer is "2 +
2" and the student's answer is "4", the student's answer should be considered
correct. Vice versa is also true.

7. If the student's answer is mathematically inequivalent to the ground truth
answer, it should be considered incorrect. For example, if the ground truth
answer is "5" and the student's answer is "-5", the student's answer should be
considered incorrect.

Question: {question}
Ground Truth: {ground_truth}
Student Answer: {student_answer}
Is the student's answer correct? Answer 'Yes' or 'No'.
Answer:
```

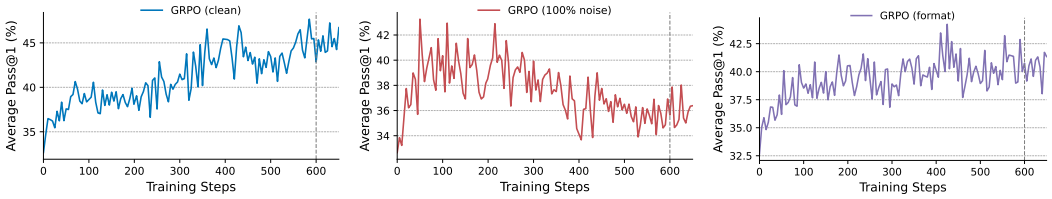


Figure 17: After three epochs (600 steps), no model shows improvements of average pass@1 of all benchmarks in the subsequent 50 steps.

C Determination of Training Duration

To establish a rigorous stopping criterion for math reasoning training, we monitored the average performance across all benchmarks over nearly 600 training steps. As illustrated in Figure 17, the model trained on clean data (left) exhibits a steady ascent, reaching peak performance stability around 600 steps (3 epochs). In contrast, the model trained on 100% noise peaks early (approx. step 200) before degrading, confirming that prolonged exposure to noise leads to overfitting and performance collapse. The format-only model stabilizes quickly and maintains a consistent pass@1. Consequently, we set the training duration to 600 steps for all experiments, as no model demonstrated significant improvements in the subsequent 50 steps.

Similarly, to determine the stopping criterion for Text2SQL training, we monitored the average training rewards. We did not monitor test accuracy on BIRD mini-Dev to prevent overfitting to a single benchmark. As illustrated in Figure 18, starting the tenth epoch, no model shows improvements in training rewards. Consequently, we set the training duration to 10 epochs for all Text2SQL experiments.

D Training Hyperparameters

We list all hyperparameters used in the math (Sections A.1–A.6) and Text2SQL experiments. Hyperparameters that differ across algorithms or models are highlighted in their own tables; everything else is shared.

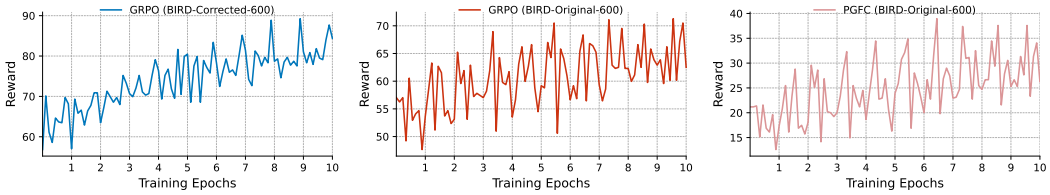
D.1 Math RLVR Hyperparameters

The Qwen math experiments fine-tune Qwen2.5-Math-7B with full-parameter updates on the DeepScaleR training set under the SkyRL training stack. Settings shared by the Qwen math runs are given in Table 2. Algorithm-specific deviations from this common configuration are listed in Table 3.

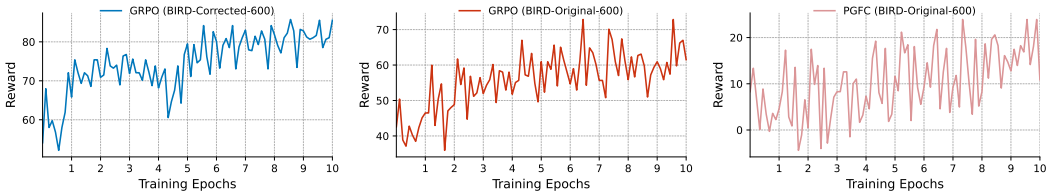
The additional Nemotron experiment uses `nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-BF16` with Tinker, LoRA rank 32, and a learning rate of 5×10^{-5} at 0%, 50%, and 100% noise (Section 4).

Loss aggregation. The "Loss aggregation" column in Table 3 controls how the per-token policy-gradient loss is reduced into a single scalar per gradient step. Let $\ell_{i,t}$ denote the token-level loss for the t-th token of the i-th rollout in the batch, and let $|x_i|$ be the response length of rollout i. The two modes used in our experiments are:

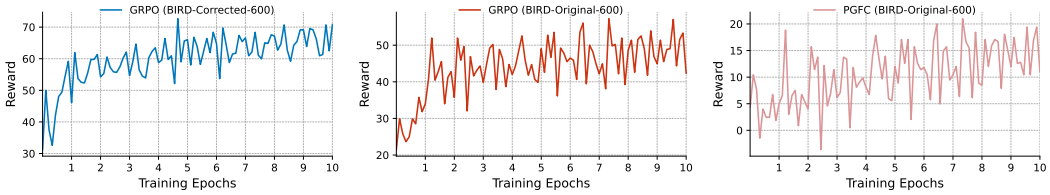
- **Token mean**: $\mathcal{L} = \frac{1}{\sum_i |x_i|} \sum_i \sum_{t=1}^{|x_i|} \ell_{i,t}$, i.e., a uniform average over every valid token in the batch. Long rollouts therefore contribute more to the gradient than short ones.
- **Sequence mean**: $\mathcal{L} = \frac{1}{B} \sum_{i=1}^{B} \frac{1}{T_{\max}} \sum_{t=1}^{|x_i|} \ell_{i,t}$, where B is the batch size and $T_{\max}$ is the configured maximum generation length (3072 in our setup). Dividing each rollout's token-sum by the same constant $T_{\max}$ removes the length bias that token-mean aggregation introduces, as proposed by Liu et al. [17].



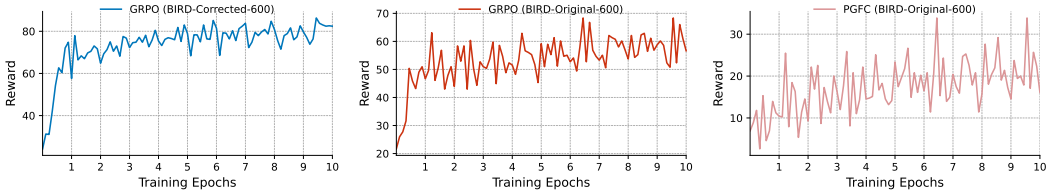
(a) Qwen3-235B-A22B-Instruct-2507.



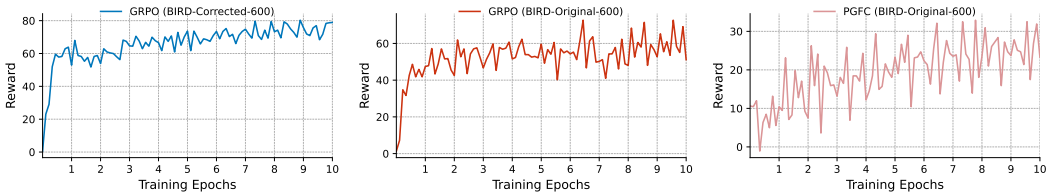
(b) DeepSeek-V3.1.



(c) Qwen3-32B.



(d) GPT-OSS-120B-A5B.



(e) Llama-3.3-70B-Instruct.

Figure 18: Starting the tenth epoch, no model shows improvements in terms of reward.

D.2 Text2SQL RLVR Hyperparameters

The Text2SQL experiments fine-tune five open-weight models with LoRA-style adapters via the Tinker SDK against the BIRD training/test split described in Section C. Because the training infrastructure (managed distributed inference + LoRA fine-tuning) is identical across models, every run uses the configuration in Table 4; the only run-level differences are the base model checkpoint, the training data file (clean BIRD-Corrected vs. noisy BIRD-Original), and the PGFC noise rate where

Table 2: Hyperparameters shared by the Qwen math RLVR runs. All algorithms use the GRPO advantage estimator on top of these settings; algorithm-specific overrides are listed in Table 3.

Category	Hyperparameter	Value
Model	Base model	Qwen/Qwen2.5-Math-7B
	Training strategy	FSDP2 (colocated train + generate)
	GPUs per node	4
	Gradient checkpointing	enabled
Optimization	Train batch size (prompts)	64
	Policy mini-batch size	64
	Micro-batch per GPU (fwd / train)	2 / 2
	Update epochs per batch	1
	Max grad norm	1.0
	KL loss coefficient	0
	Learning rate	5×10^{-6}
	Weight decay	0.01
Rollouts	Samples per prompt (group size)	16
	Max prompt length	1024 tokens
	Max generation length	3072 tokens
	Sampling temp. (train / eval)	1.0 / 0.0
	vLLM GPU memory utilization	0.8

Table 3: Algorithm-specific hyperparameters. Entries marked "–" use the shared default from Table 2, including the learning rate of 5×10^{-6} and the zero KL coefficient. For algorithms inherited from prior work we follow the hyperparameters recommended in the original papers.

Algorithm	Loss aggregation	Other
GRPO	token mean	—
PGFC	token mean	$\hat{\rho}_1 = 0.50$ (annotation-noise fraction)
Dr. GRPO	sequence mean	`grpo_norm_by_std=false`
DAPO	token mean	see below
TIS	token mean	DAPO + IS-ratio cap 2.0
SAPO	token mean	τ_+=1.0, τ_-=1.05

DAPO settings: ϵ_{low}=0.2, ϵ_{high}=0.28, c=10.0; dynamic sampling in *filter* mode with at most 30 sample batches; soft over-long penalty (buffer 2048 tokens, penalty factor 1.0); over-long filtering enabled. TIS inherits all DAPO settings on top of the importance-sampling ratio cap.

applicable. All runs stop after 10 epochs, at which point training reward has saturated for every model (Figure 18).

Importance-sampling policy loss. The Text2SQL training loop in the Tinker SDK separates the *sampling policy* q (the model that produced the rollout, served from a managed inference fleet) from the *learner policy* p_θ (the on-device weights being updated). These two distributions can drift apart between sampling and the gradient step because rollouts are generated on a separate inference endpoint and inference engines exhibit small numerical non-determinism relative to the trainer. To correct the resulting off-policy bias, the SDK's `importance_sampling` loss is applied at the token level: each token's advantage is multiplied by the learner-vs.-sampler probability ratio for that token, and the result is summed along the sequence,

$$\mathcal{L}_{\text{IS}}(\theta) \;=\; -\sum_{t=1}^{|x|} \frac{p_\theta(x_t \mid x_{<t})}{q(x_t \mid x_{<t})}\, A(x_t) \;=\; -\sum_{t=1}^{|x|} \exp\big(\log p_\theta(x_t \mid x_{<t}) - \log q(x_t \mid x_{<t})\big)\, A(x_t),$$

where $A(x_t)$ is the token-level advantage from group-relative reward centering (the same GRPO-style estimator used in the math experiments), $\log q(x_t \mid x_{<t})$ is recorded by the inference engine at sampling time, and $\log p_\theta(x_t \mid x_{<t})$ is recomputed by the learner during the forward pass. Unlike PPO, the ratio is *not* clipped: this yields the standard token-level REINFORCE-with-importance-sampling gradient and is appropriate when the sampler and learner are expected to remain close (only one optimizer step is taken per batch of rollouts in our setup). The token-level loss is summed within each sequence and averaged across the batch.

Table 4: Hyperparameters shared by every Text2SQL RLVR run. Runs with PGFC additionally set the `noise_rate` hyperparameter to 0.62; all other runs leave it unset.

Category	Hyperparameter	Value
Optimization	Batch size (prompts)	64
	Group size (rollouts per prompt)	16
	Learning rate	5×10^{-5}
	LoRA rank	32
	Number of epochs	10
	Loss function	importance sampling
Rollouts	Max input tokens	32,768
	Max output tokens per turn	3,072
	Per-rollout timeout	180 s
	Conversation prefix	enabled
	Curriculum learning	disabled
	Dynamic sampling	disabled
	Asynchronous mode	disabled
	Multi-turn iterative query refinement	enabled
Evaluation	Sampling temperature	0.0 (greedy)
	Test set	BIRD Mini-Dev (498 instances)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We discussed and justified all the claims in the following method or experiment sections.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations in Section 8.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We show the full dataset construction process in Section 3.3, and discussed the settings of hardware and hyparameters in Section 4, 5, and 6. We also submitted the code and data in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We submitted the data and code in the supplemental material, as well as the instruction to reproduce all results of this paper in the README.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specified the experimental setting and details in Section 4, 5, 6, Appendix C, and supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We reported the variation of accuracy in the figure and report the average accuracy of three checkpoints for each experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We reported the compute resources used in this paper in Section 3.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm the paper and our research conform within the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed the broader impacts of this work in Section 8.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper is based on previous open-source models and published datasets, which come with their original safeguards and do not require additional safeguard of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have credited the data and model used in this work in the supplementary material.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We included documentation of the new assets in the supplementary material.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work has no human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is not used in original, importance, or non-standard component of the core methods of this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.