Deconstructing the "Aha Moment": A Systematic Study of Reflective Reasoning in RLVR

Anonymous ACL submission

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has catalyzed spontaneous reflective reasoning in Large Language Models (LLMs), colloquially termed the "aha moment." However, the functional utility of this mechanism remains debated, with studies questioning whether it represents true reasoning or merely Superficial Self-Reflection (SSR). In this work, we investigate the origins, utility, and failure modes of reflection in R1-Zero-like training. First, we characterize the emergence of reflection, showing that while rooted in pre-training, it is significantly amplified during RLVR as an adaptive strategy to navigate increasing task complexity. Second, through intervention experiments, we reveal a critical misalignment: explicitly rewarding reflection increases its frequency and length but fails to yield commensurate accuracy gains, indicating that behavioral mimicry does not equate to cognitive correctness. Third, we diagnose the root cause of this inefficiency as "toxic context," characterized by flawed initial reasoning coupled with excessive reflective redundancy that dilutes attention and confuses the model. To counteract this, we propose Distilled Reflection Prompting, a two-stage framework that distills flawed reasoning and verbose reflections into actionable directives. By reconstructing the context and prompting the model to re-reason, this intervention mitigates attention dilution and yields a significant recovery in accuracy. Our findings demystify the "aha moment" and provide a practical pathway to unlock the latent potential of reflective reasoning.

1 Introduction

Recent progress in large language models (LLMs) especially those trained with Reinforcement Learning with Verifiable Rewards (RLVR) has been marked by the emergence of compelling "System 2"-like reasoning behaviors (Guo et al., 2025; Zeng et al., 2025; Wang et al., 2025b; Li et al., 2025). In R1-Zero-style training, models such as DeepSeek-R1-Zero often produce longer chains of thought (CoT) featuring backtracking, verification, and occasional self-correction, a pattern colloquially described as the "aha moment" (Guo et al., 2025; Liu et al., 2025c). In practice, the community frequently treats sudden increases in response length and the appearance of reflective cues (e.g., "Wait," "Let me recheck") as evidence of emergent reasoning ability (Wei et al., 2022; Wang et al., 2022; Kojima et al., 2022).

However, recent studies question whether this "aha moment" marks a genuine transition from statistical next-token prediction to autonomous error correction, or merely reflects statistical artifacts (Liu et al., 2025b; Shah et al., 2025; Yue et al., 2025). Critics point out that reflective patterns are observable even at epoch 0, suggesting that these capabilities originate from pre-training rather than being induced by R1-Zero-like training (Shah et al., 2025). Some analyses even challenge the existence of the "aha moment" as a distinct phenomenon in this context (Liu et al., 2025b). Furthermore, Liu et al. (2025b) identify a phenomenon termed Superficial Self-Reflection (SSR), where reflective text mimics the stylistic form of reasoning but fails to constructively revise errors, often leading to reasoning loops or the degradation of originally correct solutions. Consequently, whether the "aha moment" genuinely exists as a functional mechanism in R1-Zero-like training, and whether it actively contributes to correctness, remains an unresolved question.

To comprehensively understand this reflective capability and demystify the mechanisms behind it, we structure our exploration into three phases: (1) how reflection arises, (2) whether it genuinely aids reasoning, and (3) why it often fails and how to fix it. **First**, regarding the origin of reflection, while prior work attributes these capabilities to

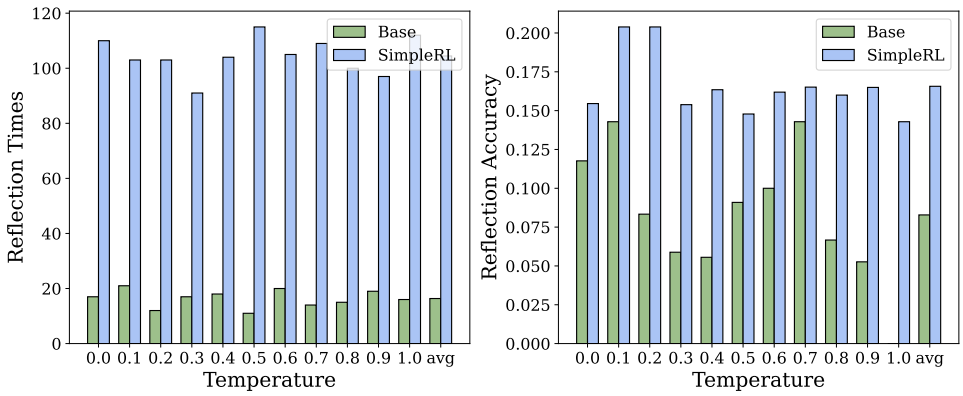

Figure 1: **RLVR increases reflection frequency and reflection-conditional reliability under controlled decoding. Left:** Reflection occurrences across a temperature sweep. **Right:** $P(\text{correct} \mid \text{reflection})$ across the same sweep. Here, "Base" denotes the Qwen2.5-3B base model, while "SimpleRL" represents its RLVR-trained counterpart. All settings (prompt, top-p, $T_{\max}$, sampling budget) are held constant between the Base model and the RLVR model.

pre-training, we specifically examine the impact of RLVR. We verify that RLVR significantly enhances reflection propensity, primarily because the model learns to deploy self-reflection as an adaptive strategy when encountering increasingly difficult problems during training. **Second**, we investigate the utility of this behavior. By introducing an explicit reflection reward to encourage frequency, we observe a counter-intuitive result: as reflection frequency increases, accuracy does not improve commensurately. This suggests that mere behavioral reflection does not automatically translate to effective reasoning. **Finally**, to understand this disconnect, we analyze why reflection often yields limited gains. We attribute this to two factors: the intrinsic task difficulty and a "toxic context"—driven by flawed initial trajectories coupled with verbose reflections—that dilutes attention and confuses the model. Crucially, by exclusively targeting this "toxic context" to mitigate cognitive interference, we demonstrate that it is possible to unlock the true quality of reflection, thereby enabling the model to effectively solve more complex problems.

Our main contributions are summarized as follows:

- **Characterizing the Origins:** We systematically demonstrate that RLVR amplifies reflection not as a pre-training artifact, but as an adaptive strategy for complex problem-solving.
- **Evaluating Reflection Utility:** We reveal a critical disconnect between reflection frequency and reasoning accuracy through explicit reward shaping.
- **Proposing Distilled Reflection Prompting:** We identify "toxic context" as a primary failure mode and introduce a two-stage framework to mitigate cognitive interference, significantly restoring model performance.

2 Related Work

RLVR and R1-Zero-like Paradigms. RLVR scales post-training by deriving supervision from deterministic outcomes (e.g., math exact matches, code unit tests) to bypass human annotation bottlenecks (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Le et al., 2022; Bai et al., 2022; Ouyang et al., 2022; Rafailov et al., 2023). Its "R1-Zero-like" instantiation uniquely eliminates supervised fine-tuning, applying policy gradients (e.g., GRPO) directly to base models to enforce structured reasoning (e.g., `<think>` tags) (Schulman et al., 2015, 2017; Guo et al., 2025; Zeng et al., 2025). While this recipe distinctively elicits verbose reasoning traces and emergent "aha moments," it remains debated whether these methods genuinely expand reasoning boundaries or merely improve sampling efficiency for latent capabilities (Yue et al., 2025). This uncertainty necessitates diagnostics beyond simple accuracy metrics to probe their underlying generation dynamics.

The "Aha Moment": Phenomenology and Proxies. The "aha moment" describes an abrupt shift in RLVR training toward System-2-like behaviors, including extended CoT, backtracking, and self-correction (Yao et al., 2022; Kojima et al., 2022; Wei et al., 2022; Zelikman et al., 2022; Suzgun et al., 2023). Lacking ground-truth reasoning la-

bels, prior work frequently relies on heuristic proxies (detailed in Appendix): (i) abrupt length increases, (ii) frequent reflective cues (e.g., "Wait"), and (iii) aggregate performance spikes. However, these proxies are highly confounded. Prompting and stochastic decoding can elicit similar verification patterns even in untrained base models (Wang et al., 2022; Zhou et al., 2022; Yao et al., 2023; Yu et al., 2025). Furthermore, increased rationale length may merely reflect reward shaping and exploration-exploitation dynamics rather than genuine discrete capability jumps (Dutta et al., 2024; Yue et al., 2025; Shah et al., 2025). These confounders underscore the need to replace superficial proxies with outcome-aware metrics that distinguish the mere propensity to reflect from the effectiveness of that reflection.

Reflection Dynamics: Behavior vs. Effectiveness. Disentangling "aha" phenomena requires separating observable reflection behavior from effective reflection, strictly operationalized as verifiable transitions from incorrect intermediate states to correct final answers (Uesato et al., 2022; Lightman et al., 2023). This distinction is vital, as reflective language often emerges as pre-training artifacts rather than genuine reasoning (Zelikman et al., 2024; Mondorf and Plank, 2024; Pirozelli et al., 2024; Yeo et al., 2025). Recent studies identify Superficial Self-Reflection (SSR)—where models mimic verification by looping or overturning correct solutions without constructive revision—which inflates keyword-based metrics (Liu et al., 2025b; Wu et al., 2025; Wang et al., 2025a; Liu et al., 2025a). Furthermore, multi-path sampling strategies (e.g., self-consistency) confound evaluation by boosting aggregate accuracy without intrinsic trajectory improvement. Consequently, analyzing RLVR demands controlled decoding and granular metrics—separately quantifying reflection propensity, correction success rates (CSR), and SSR prevalence—to stop conflating verbosity with reasoning competence.

3 Analysis of Reflection in RLVR

In this section, we conduct an in-depth investigation into the "aha moment" within RLVR training. We employ a stepwise diagnostic framework covering three key dimensions: (1) **how reflection arises** (analyzing difficulty, decoding, and training distributions), (2) **whether reflection helps** (comparing baseline RLVR vs. reflection-reward RLVR), and (3) **why reflection fails** (examining length effects and counterfactual ablations). Through this analysis, we aim to demystify the reflection phenomenon and establish a path for transforming superficial reflective behaviors into genuine reasoning competence.

Benchmarks and Setup. We utilize Qwen2.5-3B (Base) (Qwen et al., 2024) as our primary backbone across all analyses and foundational experiments. To further validate the cross-model generalizability of our findings, we additionally incorporate Llama-3.2-3B-Instruct (Grattafiori et al., 2024) in a select subset of evaluations. Both models are trained via an R1-Zero-like RLVR framework. To evaluate reasoning performance across a spectrum of difficulty, we utilize the following benchmarks: **GSM8K** (Li et al., 2024), **MATH500** (Hendrycks et al., 2021), **AMC** (Li et al., 2024), **Minerva Math** (Lewkowycz et al., 2022), and **OlympiadBench** (He et al., 2024).

Metrics. We categorize our evaluation measurements into two dimensions:

- **Surface-level Metrics:** We measure the response length $L(y)$ and the binary reflection status $\text{IsReflective}(y) \in \{0, 1\}$. Consistent with our reward formulation, $\text{IsReflective}(y) = 1$ if the response contains any predefined reflective patterns, and 0 otherwise.
- **Outcome-aware (Conditional) Metrics:** To assess the efficacy of reflection, we examine the probability of correctness given that reflection occurred. Let $\text{Correct}(y) \in \{0, 1\}$ denote the verifier outcome. We compute:

$$P(\text{Correct}(y) = 1 \mid \text{IsReflective}(y) = 1).$$

  We further stratify this metric by difficulty bins to mitigate spurious correlations (as reflection is disproportionately triggered by harder instances).

3.1 How Reflection Arises: Complexity as a Catalyst

We begin by examining the natural emergence of reflection under standard RLVR conditions.

RLVR Setup. We study RLVR under an R1-Zero-like framework, where a base model is optimized via verifiable rewards. Each input problem x is wrapped in a fixed template to elicit a reasoning trace followed by a final answer. The policy π_θ generates outputs $y = (y_{1:T})$ using nucleus sampling (top-p) with temperature τ, sub-

ject to a token budget $T_{\max}$. We employ a strictly **rule-based reward function** $r(x, y)$ that evaluates solely the correctness of the final answer; crucially, this function does *not* incentivize response length, formatting structure, or specific reflection cues. Optimization is performed using **Group Relative Policy Optimization (GRPO)** (Shao et al., 2024) with KL regularization to the reference policy. Unless otherwise stated, we keep the prompt template (see Appendix B), decoding hyperparameters, and sampling budget fixed across comparisons to ensure that observed behavioral shifts are attributable solely to the intervention.

Identifying explicit reflection. Our analysis focuses on *explicit reflection*, i.e., textual segments indicating rethinking, verification, or attempted correction. We operationalize reflection using a lightweight cue-based detector: a generation is marked reflective if it contains at least one cue token from a curated set (e.g., *wait*, *recheck*, *let me verify*, *try again*). We count reflection occurrences $K(y)$ by matching cues case-insensitively and collapsing trivial repetitions (e.g., identical consecutive cues) to reduce degenerate inflation. The curated list of reflection cues is shown in Appendix B.

RLVR amplifies latent reflection capabilities across decoding temperatures. To characterize the propensity for reflection, we first perform a controlled sweep over decoding temperatures using Qwen2.5-3B (Figure 1). We observe that while the base model possesses latent reflective capabilities, RLVR training consistently amplifies this behavior across the entire temperature spectrum. Crucially, even at low temperatures where the base model rarely reflects, the RL-trained model exhibits a significantly higher frequency of reflective cues. This indicates that RLVR does not necessarily create reasoning skills from scratch, but rather unlocks and incentivizes the extensive use of "System 2" verification patterns that were dormant or infrequent in the base model.

Reflection is an adaptive behavior driven by task complexity. To elucidate why standard RLVR improves reflection capabilities without explicit incentives, we analyze the relationship between reflection and problem difficulty. **First**, stratifying the MATH500 dataset by difficulty level (Figure 2) reveals a distinct selection bias: the average difficulty of problems that trigger reflection is significantly higher than the dataset baseline. This suggests that the model learns to deploy reflection as a specialized strategy for high-uncertainty instances. Based on this, we **hypothesize** that enhanced reflection emerges as an adaptive response during RLVR exploration: the model discovers that direct reasoning paths are insufficient for complex problems, prompting it to engage in self-correction. The subsequent rewards obtained from solving these previously intractable instances serve to reinforce this reflective mechanism. **Second**, to verify this hypothesis, we compare Qwen2.5-3B models trained on simple versus complex curricula across training checkpoints (Figure 3). The model trained on complex tasks exhibits a markedly higher propensity for reflection compared to its counterpart. Together, these results confirm that the complexity of the training distribution is a primary catalyst for the emergence of reflective capabilities in RLVR.

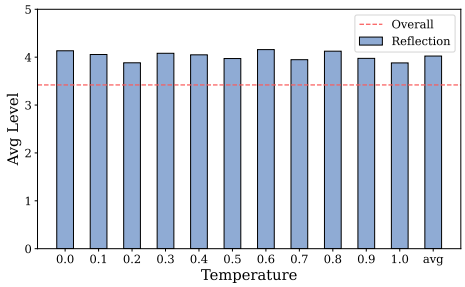

Figure 2: **Reflection is disproportionately triggered by more challenging problems.** Here, "Reflection" denotes the average difficulty of the subset of problems that triggered a reflection, while "Overall" refers to the average difficulty of the entire dataset.

3.2 Does Reflection Help: Incentives Increase Frequency but Not Correctness

Having established that RLVR naturally enhances reflective capabilities, we now investigate a critical follow-up question: is this ability genuinely beneficial to the model? To explore this, we introduce an explicit reflection reward to the rule-based RLVR framework. Our experimental design serves a two-fold objective: first, to verify whether this reward effectively incentivizes the model to reflect more frequently; and second, to investigate whether promoting reflection translates into improved problem-solving capabilities on complex tasks. Accordingly, in this section, we systematically compare the reflection patterns and reasoning accuracy of baseline RLVR against variants trained with reflection-based rewards.

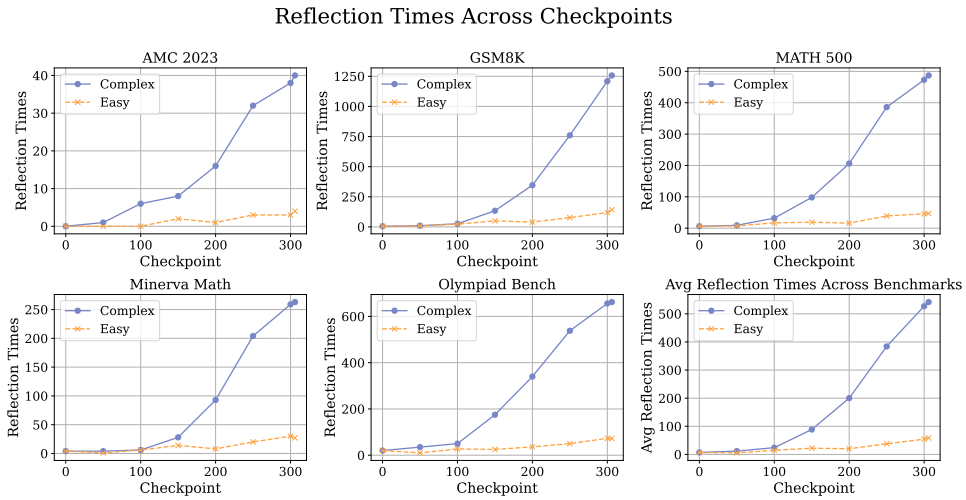

Figure 3: **Task complexity promotes reflection propensity.** We train two Qwen2.5-3B models using (i) simpler tasks and (ii) more complex tasks, then track reflection frequency across checkpoints on multiple benchmarks. Here, "Complex" refers to the model trained on complex problems, while "Easy" refers to the model trained on simple problems. Models trained on harder distributions exhibit systematically higher reflection rates.

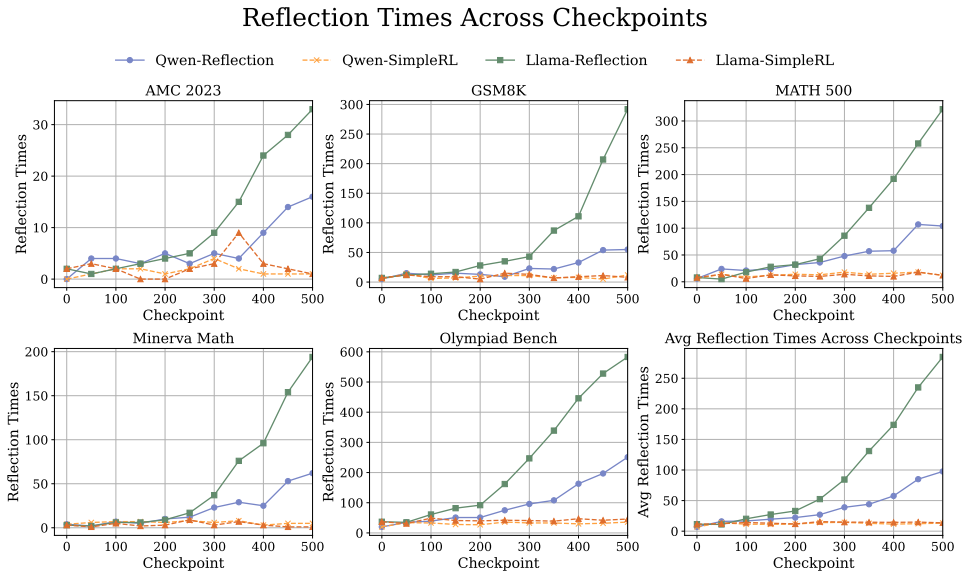

Figure 4: **Reflection reward significantly increases reflection occurrences across checkpoints.** Incorporating a reflection bonus into RLVR substantially increases reflection frequency compared to the baseline across both architectures. In the legend, the prefixes ('Qwen', 'Llama') indicate the backbone models, while the suffixes ('-Reflection', '-SimpleRL') denote variants trained with the auxiliary reflection reward and the standard outcome-only RLVR baseline, respectively.

Baseline RLVR objective. Following the standard RLVR setup outlined in Section 3.1, we instantiate the baseline reward based solely on the correctness of the final answer. While the original DeepSeek-R1 implementation incorporates a format reward to enforce structural constraints (Guo et al., 2025), we omit this component for two reasons. First, as noted by Zeng et al. (2025), explicit format rewards may hinder the emergence of reasoning capabilities, particularly in smaller models. Second, empirical observations suggest that formatting compliance converges rapidly, rendering a dedicated reward term redundant. Consequently, our baseline reward function is simplified as:

$$r(x, y) = \mathbb{I}(y \text{ is correct}),$$

where $\mathbb{I}(\cdot)$ denotes the indicator function for solution correctness. By relying solely on a binary outcome signal, this setup forces the model to implicitly discover effective reasoning patterns—such as reflection—without direct supervision on the reasoning process itself.

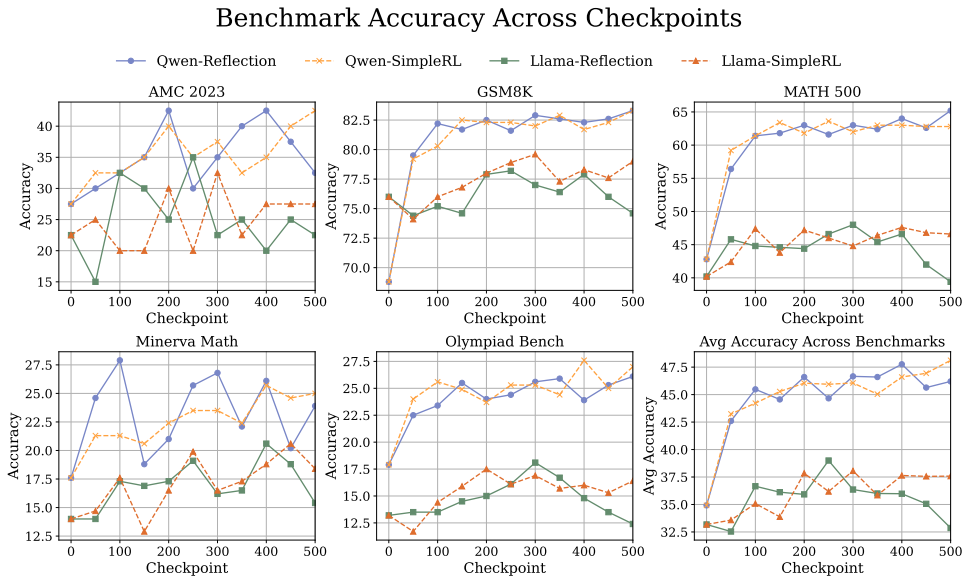

Figure 5: **Increased reflection frequency does not guarantee higher benchmark accuracy.** Despite substantial increases in reflection volume, the overall accuracy of the reflection-rewarded variants ('-Reflection') remains stagnant compared to their standard RLVR counterparts ('-SimpleRL') across both Qwen and Llama architectures.

Reflection-reward RLVR intervention. To explicitly incentivize reflective behavior, we augment the baseline reward function with an auxiliary term:

$$r'(x, y) = r(x, y) + \lambda_{\text{ref}} \cdot \mathbb{I}(\text{IsReflective}(y)),$$

where λ_{ref} is a hyperparameter controlling the magnitude of the bonus, and $\mathbb{I}(\text{IsReflective}(y))$ is an indicator function that activates when the model exhibits reflective patterns. As defined previously, we detect reflection by identifying specific keywords or cues within the reasoning trace. To ensure a fair comparison and isolate the impact of this intervention, all other training configurations remain identical to the baseline RLVR setup.

Reflection incentives decouple behavior from performance. A comparative analysis between the baseline (SimpleRL) and reflection-rewarded (Reflection) variants across both Qwen and Llama architectures reveals a critical misalignment:

- **Behavioral Surge (Figure 4):** The auxiliary reward successfully conditioned both models, leading to a significant amplification of reflection frequency. Across all evaluated benchmarks, both architectures exhibited a massive behavioral surge, frequently generating tens to hundreds of reflection steps per instance. This demonstrates that the policies effectively optimized for the reflection metric.
- **Performance Plateau (Figure 5):** However, quantity did not translate to quality. Despite this massive increase in reflection frequency, benchmark accuracy remained stagnant across both architectures. As illustrated in Figure 5, the reflection-rewarded variants failed to outperform their standard RLVR counterparts, and in challenging settings like AMC23, performance was negatively impacted.
- **Superficial Mimicry (Appendix Figure 9):** Analysis of the conditional accuracy $P(\text{Correct}(y) = 1 \mid \text{IsReflective}(y) = 1)$.—defined as the proportion of correctly answered questions among the instances where reflection was triggered—reveals that the induced reflections are largely decorative in both models. These reflection-rewarded variants appear to generate textual patterns that mimic reasoning (e.g., self-checking prompts), seemingly to harvest rewards. Yet, as these patterns do not reliably drive the reasoning process toward correct solutions, this behavior suggests a form of superficial mimicry rather than genuine cognitive verification.

3.3 Why Reflection Fails: The Role of Redundancy and How to Fix It

The results in Section 3.2 present a paradox: reflection is incentivized and produced in abundance, yet accuracy stagnates. In this section, we focus on diagnosing why the model struggles to perform effective reflection. Furthermore, we investigate whether addressing these failure modes can un-

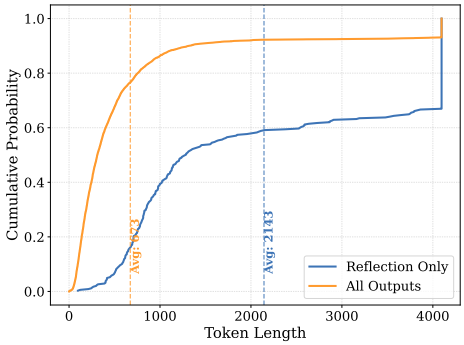

Figure 6: **Responses triggering reflection are significantly longer.** Here, "All Outputs" denotes the length distribution for the full dataset, while "Reflection Only" represents the subset triggering reflection.

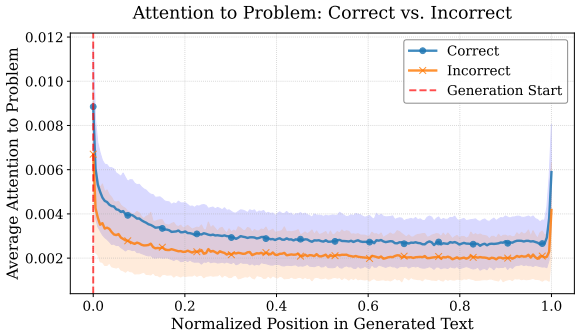

Figure 7: **Attention dynamics over normalized generation.** The plot shows mean attention mass to problem tokens as a function of normalized generation position ($0.0 =$ start, $1.0 =$ end).

lock the latent potential of reflection, enabling the model to successfully tackle problems that were previously out of reach.

Intrinsic problem difficulty acts as the primary bottleneck. First, as established in our prior statistical analysis (Figure 2) , reflection is disproportionately triggered by instances with higher intrinsic complexity. Consequently, the model exhibits a high baseline failure rate on these problems, independent of whether reflection is employed. Therefore, the observation that reflection fails to salvage accuracy does not imply that the mechanism is futile; rather, it suggests that the incremental benefit provided by reflection is currently insufficient to bridge the substantial capability gap required to solve these inherently difficult tasks.

"Toxic context" acts as cognitive interference. Intuitively, explicit reflection typically increases sequence length because the model must explore tentative paths, backtrack, and revise intermediate conclusions. We empirically verify this by plotting the cumulative distribution function (CDF) of token lengths for outputs *with* and *without* detected reflection (Figure 6). The distribution shows a pronounced rightward shift, confirming that reflective behavior is strongly coupled with length inflation.

This intrinsic verbosity introduces two critical failure modes. **(1) Context Exhaustion:** longer traces consume a substantial fraction of the token budget, increasing the risk of premature truncation before a solution is reached. **(2) Attention Dilution:** as the model generates extensive intermediate text, its attention mechanism may progressively drift away from the original problem statement. To validate this **Attention Dilution** hypothesis, we quantify the attention weights assigned to problem tokens by the generated tokens, averaging across all heads in the last three Transformer layers. Figure 11 reveals a discernible inverse correlation: reasoning traces with longer reflective segments exhibit systematically lower average attention to the problem context. Furthermore, tracking attention dynamics over normalized generation positions (Figure 7) shows that while all samples experience an initial decay, correct solutions maintain significantly higher attention levels throughout the generation process. These patterns corroborate the view that excessive verbosity acts as *cognitive interference*—diluting problem focus and exacerbating the risk of constraint forgetting and ungrounded "corrections."

Mitigating the Impact of "Toxic Context" via Distilled Reflection Prompting While an intuitive solution to verbosity is penalizing sequence length in the RLVR objective, our empirical results show this is counterproductive (see Appendix C for formulation details and degradation analysis). This suggests that the core bottleneck is not merely the raw token count, but likely the cognitive interference induced by a "toxic context"—specifically, the combination of an initially flawed reasoning trajectory and the subsequent verbose reflection that diffuses the model's attention.

To restore reasoning efficacy, we propose *Distilled Reflection Prompting*. As illustrated in Figure 8, we partition the reasoning sequence into two segments: the initial flawed trajectory (Part A) and the subsequent verbose reflection (Part B). While Part A contains misleading steps, it encapsulates critical diagnostic signals of the error; meanwhile, Part B holds valuable self-correction insights but triggers severe attention dilution due to its exces-

Model	Method	GSM8K		MATH		Minerva Math		OlympiadBench	
		Cond.	Overall	Cond.	Overall	Cond.	Overall	Cond.	Overall
Llama-Reflection	Standard Reflection	9.5	71.5	8.8	26.6	12.0	14.3	3.0	6.6
	Distilled (Ours)	**26.7**	**75.3**	**13.6**	**29.6**	**13.0**	**15.0**	**5.5**	**8.7**
Qwen-Reflection	Standard Reflection	18.9	83.3	18.4	52.4	5.7	24.2	10.2	26.0
	Distilled (Ours)	**62.1**	**85.8**	**24.5**	**53.8**	**17.3**	**27.2**	**14.0**	**27.5**

Table 1: Performance comparison of the standard reflection mechanism versus our *Distilled Reflection Prompting*. We evaluate across four mathematical reasoning benchmarks, reporting both conditional accuracy (accuracy restricted to paths where reflection inherently occurred) and overall accuracy. "Llama-Reflection" and "Qwen-Reflection" denote the respective models trained with reflection-based rewards. Notably, our distilled variant consistently outperforms the standard mechanism across all evaluated settings.

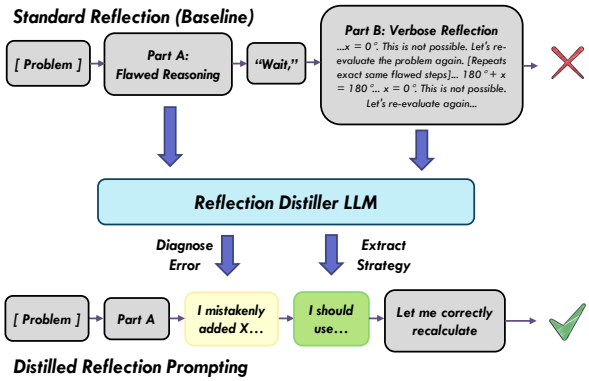

Figure 8: **Overview of Distilled Reflection Prompting.**

sive length.

We introduce an auxiliary LLM (Qwen3-Next-80B-A3B-Instruct) as a "Distiller" to perform a dual-pronged extraction: (1) isolating specific failure points from Part A, discarding misleading steps while preserving diagnostic insights; and (2) extracting actionable revision directives from Part B, drastically reducing sequence length without sacrificing semantic utility. We then reconstruct the reasoning context by retaining the original problem and Part A, but replacing the verbose Part B with these distilled signals. These signals are injected via a first-person persona to seamlessly prompt the model to resume reasoning (exact prompts are detailed in Appendix B). By actively streamlining the reasoning chain, we filter out detrimental noise and shorten the context window, thereby mitigating attention dilution and forcing the model to resume reasoning along a cleaner trajectory.

We evaluate this intervention across four mathematical reasoning datasets (GSM8K, MATH, Minerva Math, and OlympiadBench) using two distinct models previously trained on reflection-based rewards: Llama-3.2-3B-Instruct-reflection-reward and Qwen2.5-3B-reflection-reward. We compare the performance of the standard reflection mechanism against our proposed Distilled Reflection Prompting variant in terms of both conditional and overall accuracy (see Table 1). Crucially, we apply this intervention exclusively to reasoning paths that inherently trigger a reflection process.

The results demonstrate that distillation yields significant improvements in both conditional and overall accuracy across all models and datasets. This finding highlights a pivotal insight: while the model's intent to reflect is inherently valuable, the execution often fails due to the noise embedded in flawed trajectories and the subsequent diffusion of attention. By compressing reflective content into precise, actionable directives, our method mitigates cognitive interference and re-anchors the model's attention, unlocking its latent capacity for self-correction.

4 Conclusion

In this work, we conducted a systematic dissection of the "aha moment" in RLVR-trained models. We established that reflection acts as an adaptive strategy for navigating complex problem landscapes, yet naively incentivizing this behavior leads to SSR—increasing verbosity without improving correctness. We identified the root cause as a "toxic context," where flawed initial trajectories coupled with the inherent redundancy of verbose reflections severely interfere with the model's attention mechanism. Crucially, by introducing *Distilled Reflection Prompting* to distill this noise into actionable directives for re-reasoning, we confirmed that the model's self-corrective potential is genuine but heavily suppressed by its own unrefined output. Our findings suggest that unlocking the full power of reflective reasoning requires moving beyond simple reward engineering to more sophisticated inference-time interventions that systematically mitigate cognitive interference.

Limitations

First, our validation relies primarily on empirical analysis rather than theoretical formalization. While we have statistically characterized why reflective capabilities improve during RLVR, we lack a fundamental theoretical proof for these emergent behaviors. Given that the theoretical framework for Reinforcement Learning in the context of Large Language Models remains nascent, the precise internal dynamics driving these updates are not yet fully transparent. Establishing a rigorous theoretical explanation for these phenomena remains a long-term goal for the community.

Second, the challenge of effective reflection on inherently difficult tasks remains partially unresolved. While our proposed interventions mitigate the negative impact of "toxic context," they do not fundamentally alter the intrinsic complexity of the hardest problems, which continue to bottleneck performance. A comprehensive investigation into the linguistic or logical nature of these intractable tasks is beyond the scope of this work. We leave the further exploration of these frontiers to future research, aimed at developing more robust mechanisms for handling high-complexity reasoning.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. 2024. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *arXiv preprint arXiv:2402.18312*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. 2022. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Xiaoyuan Liu, Tian Liang, Zhiwei He, Jiahao Xu, Wenxuan Wang, Pinjia He, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025a. Trust, but verify: A self-verification approach to reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2505.13445*.

Zichen Liu, Changyu Chen, Wenjun Li, Tianyu Pang, Chao Du, and Min Lin. 2025b. There may not be aha moment in r1-zero-like training — a pilot study. https://oatllm.notion.site/oat-zero. Notion Blog.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025c. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Philipp Mondorf and Barbara Plank. 2024. Beyond accuracy: evaluating the reasoning behavior of large language models–a survey. *arXiv preprint arXiv:2404.01869*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Paulo Pirozelli, Marcos M José, Paulo de Tarso P. Filho, Anarosa AF Brandão, and Fabio G Cozman. 2024. Assessing logical reasoning capabilities of encoder-only transformer models. In *International Conference on Neural-Symbolic Learning and Reasoning*, pages 29–46. Springer.

A Yang Qwen, Baosong Yang, B Zhang, B Hui, B Zheng, B Yu, Chengpeng Li, D Liu, F Huang, H Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint*.

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

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

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

Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, and 1 others. 2025. Rethinking reflection in pre-training. *Preprint*.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Qian Wang, Zhanzhi Lou, Zhenheng Tang, Nuo Chen, Xuandong Zhao, Wenxuan Zhang, Dawn Song, and Bingsheng He. 2025a. Assessing judging bias in large reasoning models: An empirical study. *arXiv preprint arXiv:2504.09946*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, and 1 others. 2025b. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xuyang Wu, Jinming Nian, Ting-Ruen Wei, Zhiqiang Tao, Hsin-Tai Wu, and Yi Fang. 2025. Does reasoning introduce bias? a study of social bias evaluation and mitigation in llm reasoning. *arXiv preprint arXiv:2502.15361*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*.

Zhaojian Yu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2025. Humaneval pro and mbpp pro: Evaluating large language models on self-invoking code generation task. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13253–13279.

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

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. 2024. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

Appendix

A Experimental Details

Training Details. We conduct all training experiments for a single epoch on a compute node equipped with 4 NVIDIA A800 (80GB) GPUs. For data generation, we set the prompt batch size to 16 and generate 4 rollouts per prompt, with a maximum sequence length of 2048 tokens. The optimization step utilizes a mini-batch size of 64. Regarding hyperparameters, we employ a default sampling temperature of 1.0, a PPO clip ratio of 0.2, and a KL penalty coefficient of 1×10^{-4}.

Benchmarks. Our experiments utilize a suite of reasoning benchmarks spanning a wide spectrum of difficulty: **GSM8K** for grade-school math (Li et al., 2024), **MATH500** (a subset of the MATH dataset), which includes granular difficulty labels (Levels 1–5) (Hendrycks et al., 2021), **AMC** (sourced from American Mathematics Competitions) for challenging high-school problems (Li et al., 2024), **Minerva Math** for advanced technical reasoning (Lewkowycz et al., 2022), and **OlympiadBench** for elite competition-level problems (He et al., 2024). These benchmarks provide (1) deterministic verifiers for final-answer correctness and (2) heterogeneous difficulty distributions, enabling us to study reflection propensity as a function of uncertainty and capability gaps. Unless otherwise stated, all reported metrics are computed under identical decoding configurations and the same per-instance sampling budget.

B Prompts

B.1 Reflection Cues

To detect explicit reflection during the reasoning process, we utilize the following keyword cues:

```
"wait", "recheck", "rethink", "try again",
    "re-evaluate", "re-examine", "check
    again", "think again", "let's correct
    it and verify the steps again"
```

B.2 RLVR Standard Training Template

For RLVR training and evaluation, we format the input using the following standard template:

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{problem}

Please reason step by step, and put your
    final answer within \boxed{}.<|im_end|>
<|im_start|>assistant
```

B.3 Distilled Reflection Prompting Framework

As detailed in the main text, our Distilled Reflection Prompting framework utilizes a dual-pronged extraction strategy to mitigate the cognitive interference induced by a "toxic context." Because the core objective of this distillation process is precise information extraction and summarization rather than open-ended problem-solving, we explicitly opt against using specialized "thinking" or long-reasoning models. Instead, we select a highly capable instruction-tuned model, Qwen3-Next-80B-A3B-Instruct, deployed across four NVIDIA RTX A6000 (48GB) GPUs. This model choice ensures high semantic accuracy and strict adherence to structural constraints. To minimize generation randomness and guarantee reproducibility, we set the sampling temperature to 0.0 for all distillation tasks.

Furthermore, our extraction templates are designed with a structured Chain-of-Thought (CoT) mechanism. The Distiller is instructed to first generate an `<analysis>` block, which serves as a diagnostic scaffold to accurately pinpoint the failure cause or correction strategy. However, since our ultimate goal is to drastically shorten the context window of the reconstructed trajectory, the `<analysis>` block is entirely discarded post-generation. We exclusively extract the concise, 1-3 sentence text enclosed within the `<summary>` tags. Crucially, to ensure a seamless and natural concatenation when reinjecting these distilled insights back into the model's ongoing reasoning trajectory, we explicitly constrain the Distiller to phrase the summary from a strict first-person perspective.

The inputs for these two concurrent extractions are formatted using the following templates:

Failure Cause Extraction (from Part A)

```
<|im_start|>system
You are a helpful math reasoning
    diagnostician. You must strictly follow
    the requested XML output
    format.<|im_end|>
<|im_start|>user
Read the following math problem and the
```

```
    corresponding partial reasoning. The
    reasoning contains a mistake that the
    solver later realizes. Identify the
    specific calculation or logical error
    in this text.

Please output your response strictly in the
    following XML format:
<analysis>
Briefly analyze the solver's steps and
    locate the exact point of failure.
</analysis>
<summary>
Provide a brief summary of the error (1-3
    sentences) from a strictly first-person
    perspective (e.g., 'I mistakenly added
    X instead of Y.'). Output plain text
    only. Avoid using the phrase 'the
    solver'.
</summary>

Question:
{raw_instruction}

Reasoning:
{part_a}<|im_end|>
<|im_start|>assistant
```

Revision Directive Extraction (from Part B)

```
<|im_start|>system
You are a helpful math reasoning
    diagnostician. You must strictly follow
    the requested XML output
    format.<|im_end|>
<|im_start|>user
Read the following math problem and a
    reflection text where a solver realizes
    a mistake. Extract the high-level
    correction strategy or insight,
    ignoring any messy or redundant
    calculations.

Please output your response strictly in the
    following XML format:
<analysis>
Evaluate the reflection and identify the
    core strategy change.
</analysis>
<summary>
Provide a brief summary of the strategy (1-3
    sentences) from a strictly first-person
    perspective (e.g., 'I should use the
    quadratic formula instead.'). Output
    plain text only. Avoid using the phrase
    'the solver'.
</summary>

Question:
{raw_instruction}

Reflection:
{part_b}<|im_end|>
<|im_start|>assistant
```

B.4 Reconstructed Reasoning Trajectory

After extracting the distilled failure cause (sum_a) and revision directive (sum_b), we completely discard the original verbose reflection (Part B). We then reconstruct the reasoning context by replacing this toxic segment with our concise, first-person signals.

By reinjecting these distilled insights, we effectively bypass the cognitive interference of the original trajectory and prompt the model to resume its step-by-step reasoning along a cleaner, error-aware path. The input format for this final regeneration phase is structured as follows:

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
You are a helpful assistant.<|eot_id|>
<|start_header_id|>user<|end_header_id|>
{raw_instruction}

Please reason step by step, and put your
    final answer within \boxed{}.<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
{part_a}

Wait, let me recheck. {sum_a} To fix this,
    {sum_b} Let me correctly recalculate
    step-by-step from here:
```

C Extended Empirical Results

C.1 Detailed Analysis of Superficial Mimicry

To quantitatively support our observation of *superficial mimicry* discussed in the main text, we introduce the metric of **conditional accuracy**. It evaluates the actual problem-solving success rate specifically on instances where the model triggered at least one explicit reflection event:

$$P(\text{Correct}(y) = 1 \mid \text{IsReflective}(y) = 1)$$

As shown in Figure 9, despite the significant surge in reflection frequency under the auxiliary reward, the conditional accuracy remains disproportionately low across both Qwen and Llama architectures on all evaluated benchmarks. This supplementary data formally confirms our hypothesis: the induced reflection cues are largely decorative patterns optimized for reward harvesting, rather than genuine cognitive verification steps that lead to correct answers.

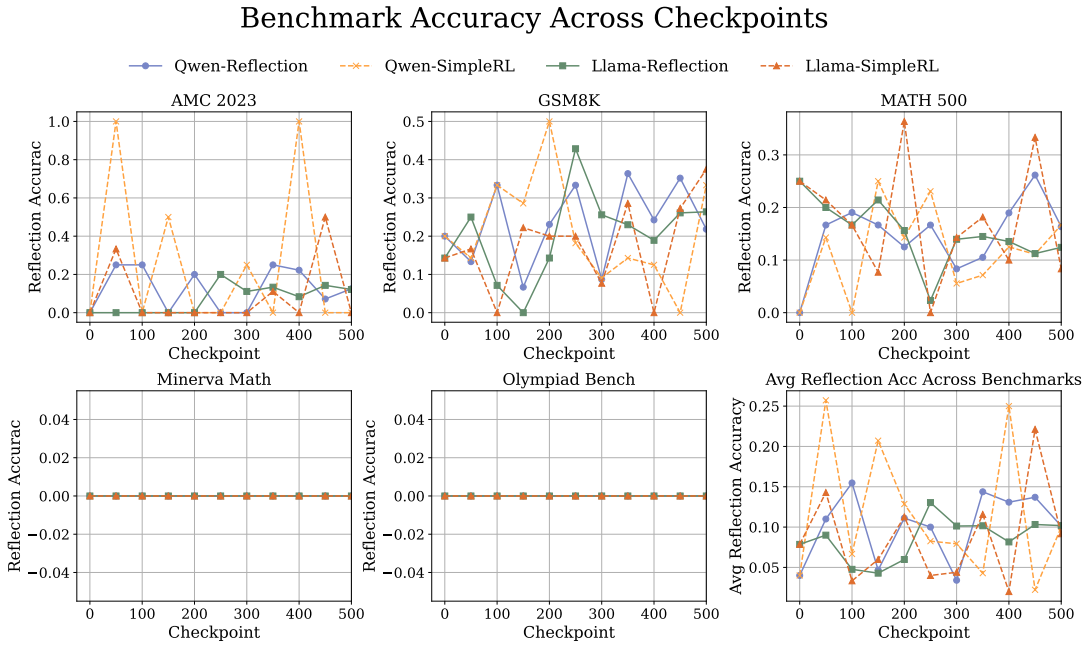

Figure 9: **Increased reflection frequency does not translate to higher accuracy on reflected samples.** We report the accuracy specifically for questions that triggered a reflection event across different benchmarks. The results indicate that while the reflection reward effectively boosts the number of reflections, the success rate on these reflection-triggering instances remains low.

C.2 Formulation and Degradation Analysis of Length Penalization

As mentioned in the main text, an intuitive approach to curb the verbosity of reflections is to incorporate a length penalty directly into the RLVR objective, mirroring the reflection reward structure. Our initial goal was to induce the model to perform reflections that are both distinct (rewarded) and concise (penalized). To achieve this, we modified the reward function as follows:

$$r_{\text{len}}(x, y) = s\big(L(y),\ \text{Acc}(x, y)\big) + \lambda_{\text{ref}} \cdot \mathbb{I}(\text{IsReflective}(y)), \tag{1}$$

where $L(y)$ denotes the output length in tokens, $\text{Acc}(x, y) \in \{0, 1\}$ represents the verifier correctness, and $\mathbb{I}(\text{IsReflective}(y))$ is the binary indicator for reflective patterns.

The length-shaping term $s(\cdot)$ utilizes a cosine schedule based on the progress ratio $p(y) = \min(L(y)/T_{\max}, 1)$. We define the shaping function piecewise:

$$s(L, \text{Acc}) = \begin{cases} s_{\text{corr}}\big(p(y)\big), & \text{Acc} = 1, \\ s_{\text{wrong}}\big(p(y)\big), & \text{Acc} = 0, \end{cases} \tag{2}$$

where the component functions are given by:

$$s_{\text{corr}}(p) = m_c + \frac{1}{2}(M_c - m_c)\big(1 + \cos(\pi p)\big), \tag{3}$$

$$s_{\text{wrong}}(p) = m_w + \frac{1}{2}(M_w - m_w)\big(1 + \cos(\pi p)\big). \tag{4}$$

For our experiments, we configured $(m_c, M_c) = (0.5, 1.0)$ to reward brevity in correct responses, and $(m_w, M_w) = (-0.5, -1.0)$ to tightly bound penalties for incorrect verbosity. The reflection bonus $\lambda_{\text{ref}} \cdot \mathbb{I}(\text{IsReflective}(y))$ was retained, with all other training configurations held fixed to strictly isolate the effect of length regularization.

Crucially, empirical results (Figure 10) reveal that this strategy is highly counter-productive. Rather than streamlining the reasoning, the length penalty frequently degrades overall performance compared to the baseline. This negative outcome robustly supports our hypothesis discussed in the main text: the core bottleneck limiting reasoning efficacy is not merely the raw token count. Forcing coarse length constraints disrupts the model's structural logic. Instead, the performance drop stems from the cognitive interference induced by the combination of flawed reasoning and verbose corrections—a "toxic context" that cannot be resolved by simple length penalization.

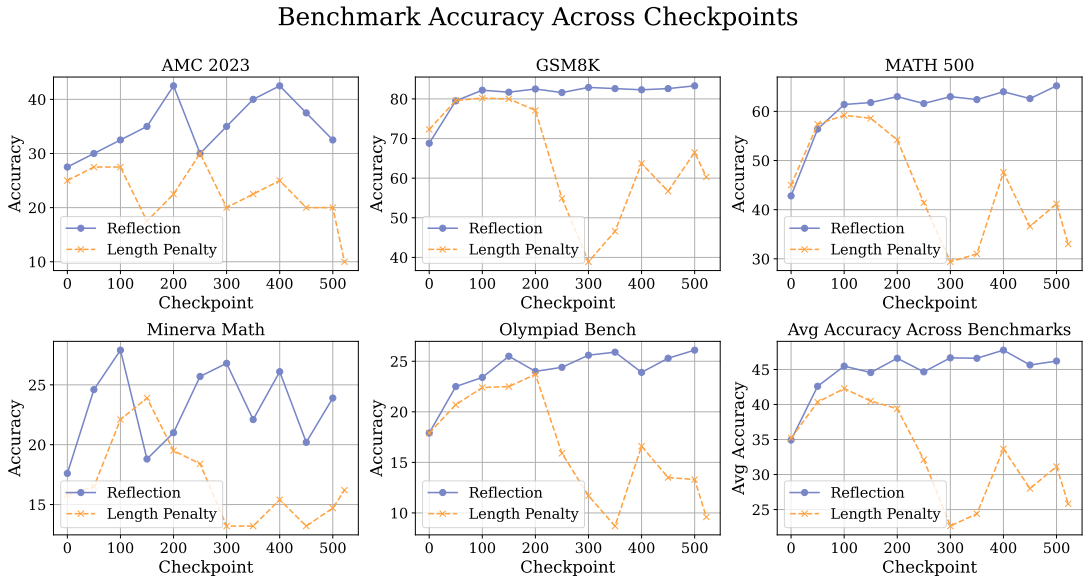

Figure 10: **Length regularization proves counter-productive for reasoning accuracy.** We compare the performance of the length-shaping intervention against the baseline RLVR model. The results demonstrate that explicitly penalizing verbosity fails to improve—and frequently degrades—reasoning accuracy. This confirms that coarse length constraints alone are insufficient to cultivate effective reflective capabilities or mitigate the negative effects of verbose reasoning.

C.3 Empirical Analysis of Attention Dilution

To quantitatively substantiate the **Attention Dilution** hypothesis discussed in the main text, we extract and analyze the internal attention matrices during the model's generation process. Specifically, we investigate whether an increase in reflective verbosity systematically reduces the model's focus on the original problem statement.

We conducted this analysis on a random subset of 100 instances from the MATH500 benchmark (comprising 74 correct and 26 incorrect solutions). To quantify "problem focus," we measured the attention weights assigned to the original problem tokens by the newly generated tokens. Since deeper layers in Transformer architectures typically capture higher-level semantic reasoning, we computed the average attention scores across all attention heads within the final three layers (layers 33–35) of the model.

As illustrated in Figure 11, mapping the number of reflection tokens against this average problem attention reveals a discernible inverse correlation. The least-squares fit confirms a downward trend: as the length of the reflection segment grows, the model's attention to the core problem diminishes. This empirical visualization provides direct, model-internal evidence for our claim that excessive verbosity acts as cognitive interference—diluting problem focus and making the model vulnerable to ungrounded "corrections."

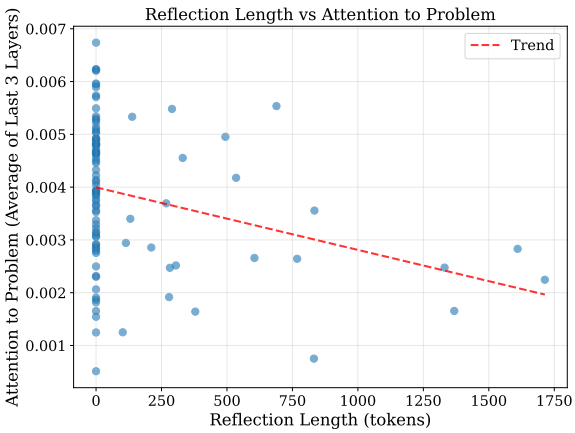

Figure 11: **Inverse correlation between reflection length and problem attention.** Scatter plot of 100 sampled instances from MATH500. The x-axis indicates the number of reflection tokens, and the y-axis represents the average attention weight paid to the problem tokens (averaged over all heads in layers 33–35). The dashed least-squares fit line visually confirms the Attention Dilution hypothesis: longer reflections systematically degrade the model's focus on the initial constraints.