ConSteer-RL: Steering Reasoning Capabilities in Large Language Models via Confidence-Aware Reinforcement Learning

Anonymous ACL submission

Abstract

Reinforcement Learning from Verifiable Rewards (RLVR) has recently become a key paradigm for improving the reasoning abilities of Large Language Models (LLMs), yet it remains limited by sparse binary rewards and its ignorance of model-internal uncertainty. In this paper, we propose **ConSteer-RL**, a simple yet effective framework that integrates token-level confidence signals derived from model log-probabilities into RLVR training. Specifically, building upon the Group Relative Policy Optimization (GRPO) framework, we construct a confidence-aware reward by aggregating per-token probabilities into a scalar confidence score and incorporating it into an awareness-based reward shaping mechanism that penalizes overconfident errors while reinforcing correct and confident reasoning. Experimental results demonstrate that ConSteer-RL consistently outperforms strong GRPO baselines, achieving average improvements of 2.3%–4.0% across different model scales. The relevant code and models will be released publicly.

1 Introduction

RLVR has emerged as a prominent paradigm for enhancing the reasoning abilities of large language models, particularly on tasks requiring long chain-of-thought (CoT) reasoning such as mathematical reasoning (Wang et al., 2026b; He et al., 2025), code generation (Jiang et al., 2025, 2026), and question answering (Burgess et al., 2026; Zhao et al., 2025; Qi et al., 2025; Zeng et al., 2026, 2025b; Zhao et al., 2026). Unlike approaches that rely on human preference annotations, RLVR leverages rule-based or model-based verifiers to automatically assess response correctness, thereby enabling scalable, efficient, and highly automated training.

Prior work such as DeepSeek-R1 (Guo et al., 2025) shows that strong reasoning capabilities can emerge without explicit supervision on intermediate reasoning steps, suggesting that RLVR alone is sufficient to induce coherent COT reasoning. In parallel, advances in policy optimization algorithms have further improved the effectiveness and stability of RLVR training. In particular, GRPO (Guo et al., 2025) introduces group relative advantage estimation as an alternative to learned value functions, reducing computational overhead while improving training stability and sample efficiency (Shao et al., 2024). These developments have significantly advanced LLM performance on reasoning tasks, establishing RLVR as a key paradigm for enhancing complex reasoning abilities and enabling scalable post-training optimization across a wide range of complex reasoning tasks.

However, existing RLVR methods still suffer from two key limitations. *(1) The binary correctness reward is extremely sparse*, providing only a terminal ± 1 supervision signal at the end of generation (Wang et al., 2026a). This makes credit assignment particularly challenging for long chain-of-thought reasoning, especially on competition-level problems where minor intermediate errors can lead to incorrect final answers (Liu et al., 2026a). *(2) Most existing approaches also fail to exploit the model's internal confidence*, treating high- and low-confidence predictions equivalently in the reward space. This design does not adequately capture the uncertainty structure of model predictions, leading to a lack of effective constraints on erroneous confidence patterns during training, and consequently limiting the model's ability to learn reliable reasoning behaviors.

To address these issues, we propose **ConSteer-RL**, a framework that integrates model-internal confidence into RLVR. As shown in Figure 1, we extract token-level confidence from log probabilities during rollouts and convert it into an additional reward component that rewards high-confidence correct responses while penalizing high-confidence incorrect ones. This confidence-aware reward is further integrated with GRPO for stable optimiza-

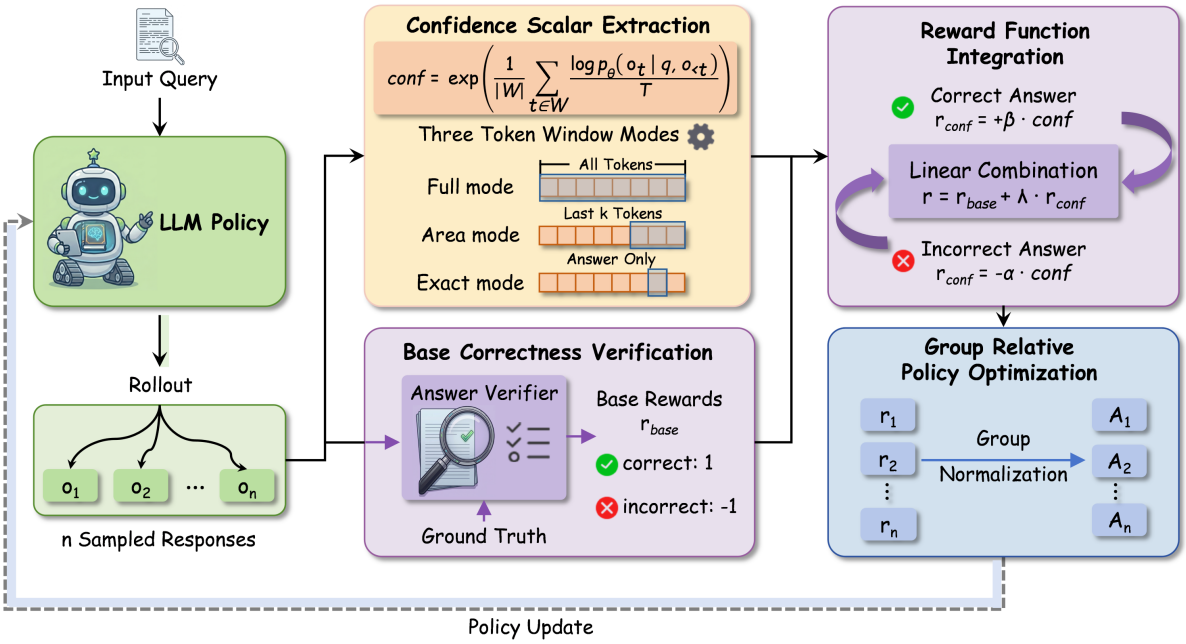

Figure 1: **Overview of ConSteer-RL.** We augment RLVR by *(1)* extracting a scalar confidence signal from per-token log probabilities, *(2)* formulating a composite reward that integrates correctness with confidence-aware shaping, and *(3)* optimizing the resulting objective with GRPO for stable policy learning.

tion. Compared with conventional RLVR methods, ConSteer-RL provides a more fine-grained and confidence-aware learning signal, encouraging more reliable reasoning behaviors. Importantly, ConSteer-RL introduces no additional reward models, auxiliary verifiers, or extra inference overhead, instead leveraging token probabilities naturally produced during rollout generation to construct an efficient confidence-aware optimization signal.

In a nutshell, our contributions are as follows:

- We propose ConSteer-RL, a reinforcement learning framework that explicitly incorporates model-internal confidence signals into RLVR training, enabling simultaneous optimization of task performance and confidence-aware reasoning without requiring additional human annotations.

- We introduce a confidence-aware reward shaping mechanism that jointly models correctness and predictive uncertainty, effectively penalizing overconfident errors while reinforcing reliable reasoning trajectories.

- Extensive experiments across multiple model families and seven mathematical reasoning benchmarks demonstrate that ConSteer-RL consistently improves performance, with particularly strong gains on challenging competition-level tasks.

2 Related Works

RL with Verifiable Rewards. RLVR has become a widely adopted paradigm for improving the reasoning capabilities of large language models. Most existing methods are based on policy optimization algorithms, particularly PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024). Recent works, including Dr. GRPO (Liu et al., 2025), DAPO (Yu et al., 2026), GSPO (Zheng et al., 2025), SAPO (Gao et al., 2025), and GDPO (Liu et al., 2026b) further improve training stability, reward design, and optimization objectives. Despite their effectiveness, these methods primarily optimize for final-answer correctness while largely neglecting the model's internal confidence during reasoning, and therefore fail to fully exploit the fine-grained uncertainty signals present throughout the generation process. In contrast, our method addresses this limitation by incorporating confidence-aware rewards into the RLVR framework, enabling the training process to benefit jointly from both answer correctness and reasoning confidence constraints.

Confidence Estimation and Optimization. Confidence estimation quantifies the reliability of model outputs and provides complementary signals

for improving LLM reasoning. Existing methods can be broadly divided into two main approaches. Black-box methods rely only on model outputs. For example, consistency-based methods estimate confidence from the agreement among multiple sampled responses (Wang et al., 2022; Manakul et al., 2023; Cheng et al., 2024; Yang et al., 2024b; Han et al., 2024), while prompting-based methods ask models to express confidence in natural language through prompts (Wei et al., 2022; Xiong et al., 2024). Although simple and compatible with closed-source models, these methods are limited by the model's intrinsic self-awareness ability, often leading to unreliable confidence estimation (Ji et al., 2023; Bani-Harouni et al., 2025). White-box methods require access to internal model representations or parameters. Some approaches train probe classifiers on hidden-layer features to predict the probability of answer correctness (Azaria and Mitchell, 2023). Our method instead is based on the assumption that high-probability tokens correspond to high-confidence predictions (Kuhn et al., 2023; Huang et al., 2023; Duan et al., 2024). We estimate confidence directly from token-level log probabilities during generation, without introducing additional training modules or inference overhead, making the method lightweight and efficient.

Confidence-aware optimization further utilizes confidence signals to improve model performance, and can be divided into inference-time and training-time approaches. Inference-time methods, such as DeepConf (Fu et al., 2025), dynamically evaluate reasoning paths during inference and improve performance by filtering low-confidence trajectories and aggregating high-confidence ones. However, these methods cannot fundamentally resolve the model's lack of uncertainty awareness. In contrast, our method performs training-time awareness optimization by incorporating confidence-aware objectives into reinforcement learning, enabling intrinsic optimization of the model's reasoning behavior.

3 ConSteer-RL

Human reasoning is naturally accompanied by self-monitoring: a solver continuously assesses their level of certainty at each step and adjusts subsequent decisions accordingly. Inspired by this metacognitive process, we propose ConSteer-RL, a framework that augments the RLVR training paradigm with an explicit, model-derived confidence signal. A scalar confidence score is first derived from per-token log probabilities produced during rollout (§3.1). This confidence is then transformed into an additive reward component that penalizes high-confidence mistakes while rewarding self-consistent and reliable successes (§3.2). Finally, the resulting composite reward, which integrates both task correctness and confidence signals, is optimized with Group Relative Policy Optimization to achieve stable policy updates (§3.3).

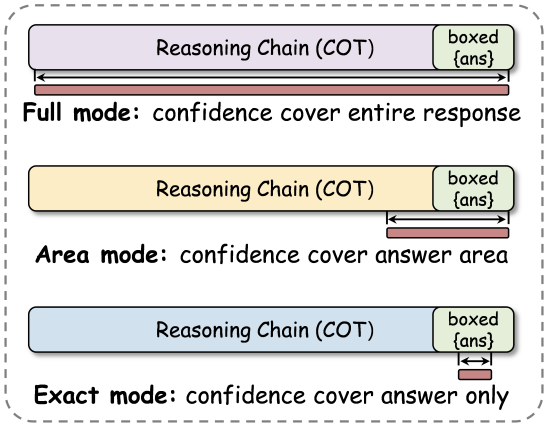

Figure 2: Definitions of token-window modes.

3.1 Confidence Scalar Extraction

A key component of ConSteer-RL is confidence extraction, which aims to quantify the model's certainty over its generated reasoning trajectory using a scalar confidence score. During rollout generation, vLLM provides per-token log probabilities $\log p_\theta(o_t \mid q, o_{<t})$, where q denotes the input query, o_t represents the t-th generated token, and $o_{<t}$ denotes the previously generated tokens. We aggregate these token-level probabilities into a unified confidence estimate by averaging the log probabilities over a predefined window $\mathcal{W}$ and scaling the result with a temperature hyperparameter T:

$$conf = \exp\left(\frac{1}{|\mathcal{W}|}\sum_{t\in\mathcal{W}}\frac{\log p_\theta(o_t \mid q, o_{<t})}{T}\right) \quad (1)$$

As shown in Figure 2, we design three token-window modes to define $\mathcal{W}$:

- **Full Mode**: $\mathcal{W}$ covers the entire generated response, providing a global estimate of the model's overall confidence throughout the reasoning process.
- **Area Mode**: $\mathcal{W}$ is restricted to the last K generated tokens, emphasizing the final answer

composition stage while filtering out intermediate scratchpad noise.

- **Exact Mode**: For tasks containing explicitly formatted boxed answers (e.g., `\boxed{...}`), $\mathcal{W}$ is the token span inside the boxed region, enabling confidence estimation to focus directly on the predicted final answer. It falls back to area mode if no box is found.

All three modes share the same formulation, differing only in the choice of token window $\mathcal{W}$.

3.2 Reward Function Integration

Equipped with the confidence score $conf$, we propose a confidence-aware reward shaping mechanism that augments the sparse binary correctness signal with an additional reward dimension, thereby yielding a more informative learning signal. Let a denote the model prediction and $a^\star$ the ground-truth. A rule-based answer verifier is first applied to determine whether the model output matches the ground truth, producing a base reward signal:

$$r_{base} = \mathbb{1}[a = a^\star] - \mathbb{1}[a \neq a^\star] \tag{2}$$

We then compute a confidence reward r_{conf} as:

$$r_{conf} = \beta\, conf \cdot \mathbb{1}[a = a^\star] - \alpha\, conf \cdot \mathbb{1}[a \neq a^\star] \tag{3}$$

The penalty coefficient α controls the strength of the penalty applied when the model produces an incorrect answer with high confidence; larger values of α impose stronger penalties on such overconfident errors, thereby discouraging incorrect yet high-confidence predictions. The bonus coefficient β determines the additional reward assigned to correct answers with high confidence; increasing β places greater emphasis on reinforcing reliable and consistent reasoning trajectories, encouraging the model to remain both accurate and confidently assertive when making correct predictions.

The final reward is defined as a linear combination of the base correctness score and a confidence shaping term:

$$r = r_{base} + \lambda \cdot r_{conf} \tag{4}$$

where $\lambda > 0$ controls the overall contribution of the confidence signal. This formulation integrates predictive confidence with task correctness into a unified learning objective, enabling the policy to jointly optimize task performance and uncertainty handling in a principled manner.

3.3 Group Relative Policy Optimization

We adopt Group Relative Policy Optimization (Shao et al., 2024) as our policy optimization algorithm. For each query q, the current policy π_θ independently samples n responses $\{o_1, \ldots, o_n\}$. GRPO dispenses with a learned value function and uses the within-group reward statistics as a baseline instead, making it more sample-efficient.

Each response o_i receives the confidence-aware composite reward $\tilde{r}_i = r_{base,i} + \lambda \cdot r_{conf,i}$ (defined in §3.2), where $r_{base,i} \in \{+1, -1\}$ and $r_{conf,i}$ is the confidence-dependent shaping term. GRPO normalizes these rewards within each query group to obtain stable advantages. For the group $\{\tilde{r}_j\}_{j=1}^{n}$ associated with query q, let μ_q and σ_q denote the group mean and standard deviation:

$$\mu_q = \frac{1}{n}\sum_{j=1}^{n}\tilde{r}_j \qquad \sigma_q = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(\tilde{r}_j - \mu_q)^2 + \delta} \tag{5}$$

where $\delta > 0$ is a small constant for numerical stability. The normalized advantage for response o_i is:

$$\hat{A}_i = \frac{\tilde{r}_i - \mu_q}{\sigma_q} \tag{6}$$

This normalization standardizes rewards within each group, stabilizing gradients and reducing the impact of varying question difficulty.

The policy is updated by maximizing the following policy gradient objective:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{q=1}^{G}\frac{1}{n}\sum_{i=1}^{n}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\hat{A}_i, \operatorname{clip}\big(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_i\Big) \tag{7}$$

where the importance sampling ratio is defined as

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} \tag{8}$$

Here, G denotes the number of queries, n is the number of sampled responses per query, and $|o_i|$ is the length of response o_i, ϵ specifies the clipping range to ensure conservative policy updates. The term $\pi_\theta(o_{i,t} \mid q, o_{i,<t})$ represents the probability of generating token $o_{i,t}$ conditioned on the query q and preceding tokens $o_{i,<t}$.

Confidence-aware rewards combined with advantage normalization form a closed optimization loop. High-confidence but incorrect responses receive stronger penalties, which are then reflected

Algorithm 1 ConSteer-RL Training Procedure

Require: Training set $\mathcal{D}$, policy π_θ, group size n, temperature T, coefficients α, β, λ.
Ensure: Updated policy parameters θ.

```
for each training iteration. do
    Sample query batch {q_1, ..., q_G} ~ D.
    for each query q do
        Sample n responses {o_i}_{i=1}^n ~ π_θ(· | q).
        Extract confidence conf_i using window mode W_i.
        Compute rewards (r_{base,i}, r_{conf,i}, r̃_i).
        Normalize group rewards {r̃_i}_{i=1}^n into advantages
            {Â_i}_{i=1}^n.
    end for
    Update policy π_θ via GRPO using advantages Â_i.
end for
return θ
```

through negative advantage signals that suppress similar trajectories in subsequent updates. Conversely, high-confidence correct responses are assigned positive advantages, reinforcing their corresponding reasoning paths and increasing their likelihood in future sampling.

To better illustrate the overall training pipeline of ConSteer-RL, the complete procedure is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Setups

Models and Datasets. To demonstrate the general applicability of ConSteer-RL, we conduct experiments across a diverse range of model scales including Qwen2.5-Math-7B (Yang et al., 2024a), Qwen3-4B-Base, Qwen3-8B-Base (Yang et al., 2025), performing direct reinforcement learning using the GRPO framework. We train our models on the DAPO-Math-17k dataset (Yu et al., 2025), which contains diverse problems spanning multiple difficulty levels. For evaluation, we consider seven challenging benchmarks, including MATH500 (Hendrycks et al., 2020), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), as well as AIME24, AIME25, AIME26, and AMC23 (He et al., 2024). These benchmarks cover a broad spectrum of difficulty, providing a comprehensive assessment of the models' mathematical reasoning ability and generalization performance.

Implementation Details. All reinforcement learning experiments are conducted using verl (Sheng et al., 2025) on 8 × A100 GPUs. We employ math-verify as the verifier during training and simpleRL verifier (Zeng et al., 2025a) for final evaluation. During training, we sample 8 responses

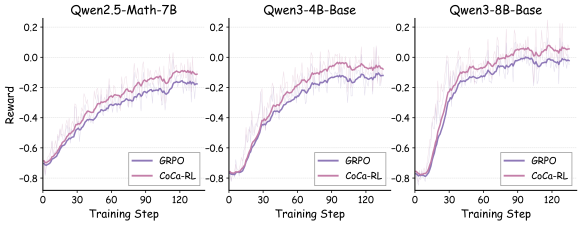

Figure 3: Comparison of training reward curves between GRPO and ConSteer-RL. ConSteer-RL rewards are rescaled to align with the range of GRPO for visualization purposes.

per prompt at a temperature of 1.0 with a batch size of 128. The model is optimized with a learning rate of 1×10^{-6} and a mini-batch size of 64. For evaluation, we report Avg@1 at temperature 0 for Math500, Minerva Math, and OlympiadBench, and Avg@32 at temperature 0.7 with top_p set to 0.8 for AIME24/25/26 and AMC23. No confidence-based interventions are applied during inference to ensure a clean evaluation setting. To prevent potential reward hacking, where a group of entirely incorrect trajectories may nevertheless receive spurious positive rewards due to lower confidence, we incorporate the Dynamic Sampling mechanism from DAPO (Yu et al., 2026). Further details are provided in Appendix B.

4.2 Experimental Results

Table 1 presents the main experimental results, from which we draw several key observations. First, ConSteer-RL consistently outperforms GRPO across all benchmarks and model families, demonstrating strong robustness across scales. Second, the performance improvements are stable and non-trivial, with average gains of 4.0% for Qwen2.5-Math-7B, 2.6% for Qwen3-4B-Base, and 2.3% for Qwen3-8B-Base. Third, the improvements are particularly pronounced on challenging competition-level benchmarks, such as the AIME series. This can be attributed to the intrinsic difficulty of such tasks, where long-horizon reasoning exacerbates the sparsity of binary reward signals, limiting their effectiveness for providing meaningful supervision at intermediate reasoning steps. In contrast, our confidence-based signal provides denser and more fine-grained feedback, which more effectively guides policy optimization over complex intermediate reasoning trajectories.

As illustrated in Figure 3, ConSteer-RL consistently outperforms the vanilla GRPO baseline across all three model scales. The reward curve

Table 1: **Main Results Comparison.** We report the performance of three models across seven mathematical reasoning benchmarks. ConSteer-RL achieves the best overall performance across all model scales, consistently outperforming both the Baseline and GRPO. The best results are highlighted in **bold**.

Model	Math500	Minerva math	Olympiad Bench	AIME24	AIME25	AIME26	AMC23	Avg.
Qwen2.5-Math-7B								
Baseline	65.0	17.6	25.2	16.7	6.7	6.7	35.0	24.7
GRPO	77.6	34.9	37.9	23.3	10.0	10.0	60.0	36.2
ConSteer-RL	**78.4**	**40.1**	**38.5**	**26.7**	**13.3**	**16.7**	**67.5**	**40.2**
Δ_{GRPO}	**+0.8**	**+5.2**	**+0.6**	**+3.4**	**+3.3**	**+6.7**	**+7.5**	**+4.0**
Qwen3-4B-Base								
Baseline	71.4	26.5	36.1	12.0	6.8	10.7	41.6	29.3
GRPO	83.6	42.6	46.7	23.2	23.4	14.0	63.0	42.4
ConSteer-RL	**85.8**	**45.2**	**47.0**	**26.5**	**26.6**	**20.2**	**63.8**	**45.0**
Δ_{GRPO}	**+2.2**	**+2.6**	**+0.3**	**+3.3**	**+3.2**	**+6.2**	**+0.8**	**+2.6**
Qwen3-8B-Base								
Baseline	76.6	37.9	40.6	10.2	13.3	6.2	51.7	33.8
GRPO	86.0	46.7	51.7	25.4	24.9	20.7	69.2	46.4
ConSteer-RL	**86.6**	**49.3**	**53.9**	**26.6**	**29.8**	**23.8**	**70.9**	**48.7**
Δ_{GRPO}	**+0.6**	**+2.6**	**+2.2**	**+1.2**	**+4.9**	**+3.1**	**+1.7**	**+2.3**

Table 2: Generalization evaluation of ConSteer-RL under different training method settings.The best results are highlighted in **bold**.

Method	AIME24	AIME25	AIME26	AMC23	Avg.
GSPO	24.1	11.1	9.4	63.3	27.0
ConSteer-RL	**27.1**	**13.8**	**10.4**	**66.7**	**29.5**
DAPO	25.5	12.3	11.5	65.4	28.7
ConSteer-RL	**29.2**	**15.6**	**14.5**	**68.2**	**31.9**
SAPO	24.6	10.7	10.2	61.2	26.7
ConSteer-RL	**27.7**	**11.5**	**12.8**	**63.1**	**28.8**

rises more steeply in the early stage of training, indicating faster initial learning, and maintains a persistent advantage throughout the entire optimization process. Notably, the performance gap continues to widen in later stages, ultimately converging to a substantially higher reward plateau. This pattern suggests that the improved confidence-aware reward modeling in ConSteer-RL provides stronger and more stable optimization signals, which not only accelerates early-stage learning but also enables more effective exploration in the later stages of training, leading to stronger final performance.

To further validate the generalizability of the ConSteer-RL method, we extend its application to GSPO (Zheng et al., 2025), DAPO (Yu et al., 2026), and SAPO (Gao et al., 2025) built upon the Qwen2.5-Math-7B backbone, and conduct experiments under identical settings for fair comparison. As shown in Table 2, incorporating ConSteer-RL consistently leads to stable and uniform performance gains across all three baseline algorithms. This demonstrates that the proposed method is not only compatible with different RL optimization frameworks but also exhibits strong transferability and broad applicability across algorithmic variants.

4.3 Ablation Studies

To better understand the impact of key design choices in ConSteer-RL, we conduct ablation studies on the temperature scaling factor T and the token-window mode. For narrative clarity, this section grounds the detailed analysis in the Qwen2.5-Math-7B model, results for the Qwen3 series exhibit consistent trends and are detailed in Appendix A.1 and A.2.

Effect of Temperature Scaling T. The temperature scaling factor T rescales token log probabilities in confidence computation, thereby affecting the resulting confidence distribution. To isolate this effect, all variants are evaluated under the Full token-window mode. As reported in Table 3, $T = 0.8$ yields the best performance.

We further examine this effect through empirical confidence distributions shown in Figure 4. Under the default setting $T = 1.0$, pretrained models tend to assign consistently high probabilities to generated tokens. As a result, the resulting confi-

Table 3: Ablation on Temperature Scaling

Benchmark	T = 1.0	T = 0.8	T = 0.5
Math500	75.0	**78.4**	75.4
Minerva Math	41.2	40.1	**41.5**
OlympiadBench	**38.7**	38.5	37.9
AIME24	20.0	**26.7**	20.0
AIME25	10.0	**13.3**	6.7
AIME26	10.0	**16.7**	6.7
AMC23	**67.5**	**67.5**	**67.5**
Average	37.5	**40.2**	36.5

Table 4: Ablation on Token-Window Mode

Benchmark	Full	Area	Exact
Math500	75.0	74.2	**76.4**
Minerva Math	41.2	**41.5**	39.0
OlympiadBench	**38.7**	37.8	37.3
AIME24	20.0	22.2	**24.7**
AIME25	**10.0**	**10.0**	5.1
AIME26	10.0	**13.3**	**13.3**
AMC23	**67.5**	59.8	57.6
Average	**37.5**	37.0	36.2

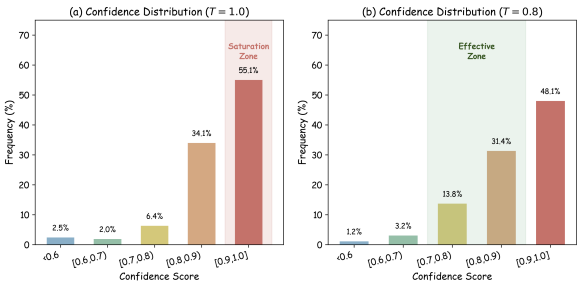

Figure 4: Effect of temperature scaling T on the confidence distribution. Lowering T effectively dispersing and deconcentrating the confidence mass.

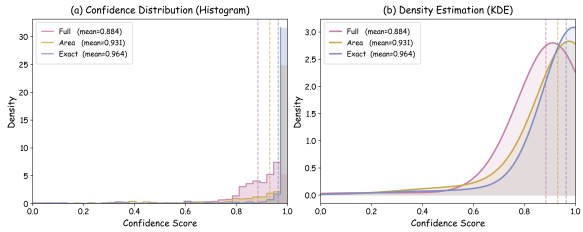

Figure 5: Analysis of token-window modes on empirical confidence distributions. Restricted modes exhibit severe confidence collapse, whereas the Full mode maintains a broader and more balanced distribution.

dence scores become heavily concentrated within a narrow high-probability region, as illustrated in Figure 4 a, leading to a pronounced saturation effect. In this regime, many samples receive similarly high confidence values, which reduces the effective separability among different reasoning trajectories. In contrast, lowering the temperature to $T = 0.8$ redistributes the confidence mass away from the saturated high-confidence region, as shown in Figure 4 b, producing a noticeably more dispersed and structured distribution. This rescaling alleviates the saturation effect observed under $T = 1.0$ and expands the effective confidence range across samples. Consequently, confidence scores become more informative and better differentiated, improving the separability among reasoning trajectories. This leads to more discriminative reward signals across samples and ultimately provides a stronger optimization signal during learning.

Effect of Token-Window Mode. We also ablate the token window $\mathcal{W}$ used for confidence calculation. We compare the three modes under the same experimental setting. To isolate the effect of the mode, all variants are evaluated at $T = 1.0$. As demonstrated in Table 4, the Full mode consistently achieves superior performance over both restricted alternatives.

Narrowing the window induces what we term confidence collapse. This phenomenon arises because the model is inherently overconfident on standard structural or formatting tokens that typically appear near the end of responses. When confidence is computed exclusively over such a restricted subset of tokens, the resulting scores become systematically inflated and thus fail to faithfully reflect the quality of the underlying reasoning process. As illustrated in Figure 5, the confidence distributions under the restricted modes (including the Area and Exact modes) are therefore heavily concentrated near 1.0, exhibiting a pronounced saturation effect. This distributional collapse further creates an evaluation bias, where high-confidence formatting tokens disproportionately dominate the aggregated score. Consequently, the model can trivially hack the reward by simply generating highly confident formatting tokens, allowing it to maximize the objective without improving the actual quality of reasoning. This ultimately leads to a degeneration of the training objective into memorization and exploitation of superficial output formats, rather than fostering meaningful improvements in fundamental reasoning capabilities.

4.4 How Confidence-Aware Reward Shapes Reasoning Patterns

To further investigate how the confidence-aware reward shapes reasoning behaviors beyond quanti-

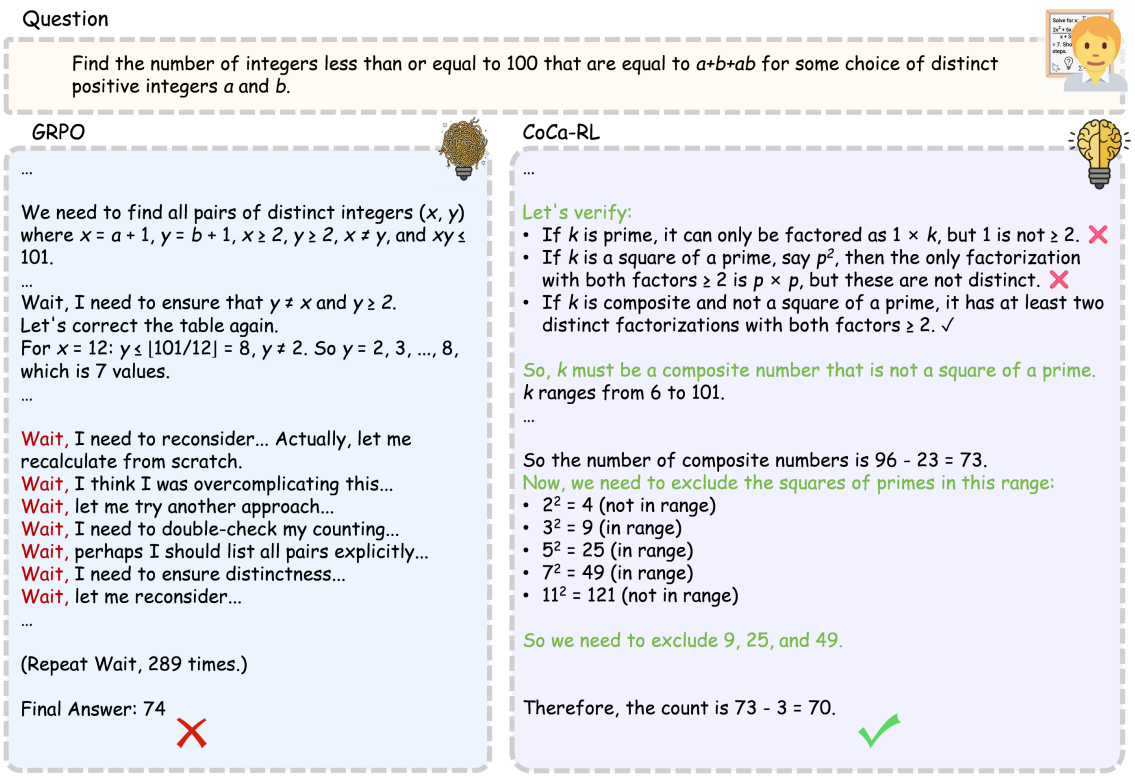

Figure 6: Case Study. GRPO falls into a 289 step "Wait" loop and fails to recover; ConSteer-RL steadily applies factorization and identifies the key non-square constraint, yielding the correct answer.

tative metrics, we conduct a qualitative comparison of reasoning trajectories between ConSteer-RL and the GRPO baseline. Through representative case studies, we analyze how confidence signals regulate reasoning depth and influence the model's decision-making behavior and reasoning strategy under different levels of uncertainty.

As shown in Figure 6, confidence should not be interpreted as blind certainty, but as a signal reflecting the model's internal assessment of its own reasoning quality and the reliability of intermediate conclusions. In contrast, GRPO tends to fall into cycles of low-confidence self-doubt, whereas ConSteer-RL exhibits more structured and adaptive reasoning behavior: when confidence is high, the model proceeds more decisively toward the final answer, avoiding unnecessary redundant steps and over-deliberation; when confidence is low, it correspondingly extends its reasoning chain to support more thorough exploration, cross-checking, and verification of intermediate results.

Overall, ConSteer-RL achieves a better balance between decisiveness under high confidence and cautious exploration under uncertainty. This balance not only results in shorter and more efficient reasoning chains, but also significantly improves answer accuracy and enhances the overall stability of model performance.

5 Conclusion

We propose ConSteer-RL, a confidence-aware RL framework that enhances RLVR by incorporating model-intrinsic uncertainty into reward design. Using token-level log-probability statistics, it derives a confidence signal that complements sparse correctness rewards and enables more fine-grained learning. Integrated with GRPO, it mitigates overconfident failure modes while strengthening reliable reasoning trajectories. Experiments across multiple benchmarks and model scales show that ConSteer-RL consistently outperforms RL baselines. Further analysis highlights the importance of temperature scaling and full token-window mode in alleviating confidence saturation and mitigating reward hacking behaviors. Overall, our results underscore the value of uncertainty-aware reward design for more reliable language models. We hope that ConSteer-RL provides the community with a scalable paradigm and fosters the development of more robust RLVR methods as well as more trustworthy language model systems.

Limitations

The current evaluation is primarily conducted on mathematical tasks, which provide a clean and controlled testbed for validating the proposed method. However, such a setting may not fully capture its behavior in broader application scenarios, especially when facing more heterogeneous inputs, ambiguous instructions, or noisy real-world contexts. In future work, we plan to extend our evaluation to a wider range of domains, including general STEM problems and multimodal tasks, to further assess the generality and robustness of the proposed approach, and to better understand how it transfers beyond strictly structured reasoning benchmarks.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976.

David Bani-Harouni, Chantal Pellegrini, Paul Stangel, Ege Özsoy, Kamilia Zaripova, Nassir Navab, and Matthias Keicher. 2025. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. *arXiv preprint arXiv:2503.02623*.

James Burgess, Jan N Hansen, Duo Peng, Yuhui Zhang, Alejandro Lozano, Min Woo Sun, Emma Lundberg, and Serena Yeung-Levy. 2026. Papersearchqa: Learning to search and reason over scientific papers with rlvr. *arXiv preprint arXiv:2601.18207*.

Qinyuan Cheng, Tianxiang Sun, Xiangyang Liu, Wenwei Zhang, Zhangyue Yin, Shimin Li, Linyang Li, Zhengfu He, Kai Chen, and Xipeng Qiu. 2024. Can ai assistants know what they don't know? *arXiv preprint arXiv:2401.13275*.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. 2025. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*.

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

Haixia Han, Tingyun Li, Shisong Chen, Jie Shi, Chengyu Du, Yanghua Xiao, Jiaqing Liang, and Xin Lin. 2024. Enhancing confidence expression in large language models through learning from past experience. *arXiv preprint arXiv:2404.10315*.

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.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, and 1 others. 2025. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. 2023. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843.

Xue Jiang, Yihong Dong, Mengyang Liu, Hongyi Deng, Tian Wang, Yongding Tao, Rongyu Cao, Binhua Li, Zhi Jin, Wenpin Jiao, and 1 others. 2025. Coderl+: Improving code generation via reinforcement with execution semantics alignment. *arXiv preprint arXiv:2510.18471*.

Yuxin Jiang, Yufei Wang, Qiyuan Zhang, Xingshan Zeng, Liangyou Li, Jierun Chen, Chaofan Tao, Haoli Bai, and Lifeng Shang. 2026. From verifiable dot to reward chain: Harnessing verifiable reference-based rewards for reinforcement learning of open-ended generation. *arXiv preprint arXiv:2601.18533*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

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.

Haolin Liu, Dian Yu, Sidi Lu, Yujun Zhou, Rui Liu, Zhenwen Liang, Haitao Mi, Chen-Yu Wei, and Dong Yu. 2026a. Save the good prefix: Precise error penalization via process-supervised rl to enhance llm reasoning. *arXiv preprint arXiv:2601.18984*.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and 1 others. 2026b. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*.

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

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017.

Yukun Qi, Yiming Zhao, Yu Zeng, Xikun Bao, Wenxuan Huang, Lin Chen, Zehui Chen, Jie Zhao, Zhongang Qi, and Feng Zhao. 2025. Vcr-bench: A comprehensive evaluation framework for video chain-of-thought reasoning. *arXiv preprint arXiv:2504.07956*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Longwen Wang, Xuan'er Wu, Xiaohui Hu, Yirui Liu, Yuankai Fan, Kaidong Yu, Qizhen Weng, Wei Xi, and Xuelong Li. 2026a. Verpo: Verifiable dense reward policy optimization for code generation. *arXiv preprint arXiv:2601.03525*.

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. 2026b. Reinforcement learning for reasoning in large language models with one training example. *Advances in Neural Information Processing Systems*, 38:122721–122764.

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.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, volume 2024, pages 23650–23678.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024a. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2024b. Alignment for honesty. *Advances in Neural Information Processing Systems*, 37:63565–63598.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*, 1:2.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

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

Yu Zeng, Wenxuan Huang, Zhen Fang, Shuang Chen, Yufan Shen, Yishuo Cai, Xiaoman Wang, Zhenfei Yin, Lin Chen, Zehui Chen, and 1 others. 2026. Vision-deepresearch benchmark: Rethinking visual and textual search for multimodal large language models. *arXiv preprint arXiv:2602.02185*.

Yu Zeng, Wenxuan Huang, Shiting Huang, Xikun Bao, Yukun Qi, Yiming Zhao, Qiuchen Wang, Lin Chen, Zehui Chen, Huaian Chen, and 1 others. 2025b. Agentic jigsaw interaction learning for enhancing visual perception and reasoning in vision-language models. *arXiv preprint arXiv:2510.01304*.

Yiming Zhao, Yu Zeng, Wenxuan Huang, Zhen Fang, Qing Miao, Qisheng Su, Jiawei Zhao, Jiayin Cai, Lin Chen, Zehui Chen, and 1 others. 2026. Videoseeker: Incentivizing instance-level video understanding via native agentic tool invocation. *arXiv preprint arXiv:2605.16079*.

Yiming Zhao, Yu Zeng, Yukun Qi, YaoYang Liu, Xikun Bao, Lin Chen, Zehui Chen, Qing Miao, Chenxi Liu, Jie Zhao, and 1 others. 2025. V2p-bench: Evaluating video-language understanding with visual prompts for better human-model interaction. *arXiv preprint arXiv:2503.17736*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Appendix Overview

- Section A: Extended Ablation Studies.
- Section B: Hyperparameters.
- Section C: Case Studies.
- Section D: Analysis Across Models.
- Section E: Dataset Details.
- Section F: Benchmark Information.

A Extended Ablation Studies

A.1 Temperature Scaling Factor Ablation

As shown in Table 5, the Qwen3 models achieve optimal performance when the temperature is set to $T = 0.5$. In contrast to Qwen2.5, the Qwen3 base models exhibit a pronounced *confidence saturation* phenomenon under the default temperature $T = 1.0$. Specifically, as illustrated in Figures 7 and 8, more than 70% of samples fall within the high-confidence interval $[0.9, 1.0]$. Such an overly peaked distribution substantially reduces the variance of the reward signal, causing the reward gradients to nearly vanish and thereby undermining the effectiveness of optimization. When the temperature is reduced to $T = 0.5$, this issue is significantly alleviated, with fewer than 50% of samples remaining in the $[0.9, 1.0]$ range. The lower temperature effectively spreads the confidence distribution, increasing the discriminative resolution between samples and providing a more stable and informative learning signal for the optimizer.

Figure 9 further presents a comparison across models of different scales. We observe that stronger models such as Qwen3-4B-Base and Qwen3-8-Base tend to suffer from more severe confidence saturation. This suggests that as model capacity increases, the output distribution becomes increasingly concentrated, necessitating more aggressive temperature scaling to maintain sufficient reward variance and ensure effective optimization during reinforcement learning.

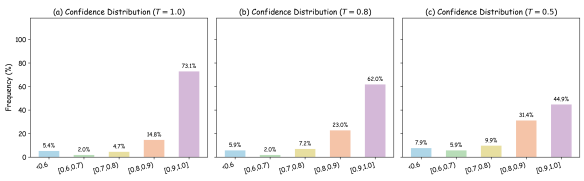

Figure 7: Confidence distribution of Qwen3-4B-Base across different temperatures T.

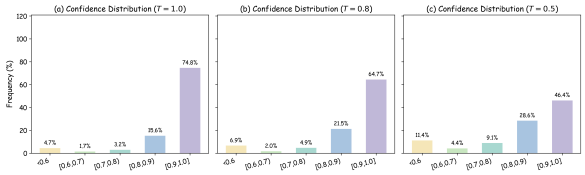

Figure 8: Confidence distribution of Qwen3-8B-Base across different temperatures T.

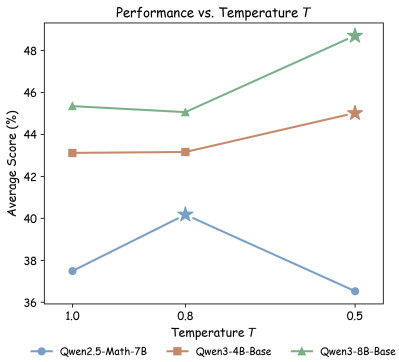

Figure 9: Model performance across different T values. The best performance is marked with ⋆.

A.2 Token-Window Mode Ablation

Table 6 further demonstrates that, on Qwen3-series models, the **Full** mode is consistently superior to both the **Area** and **Exact** constrained modes, exhibiting stronger overall optimization effectiveness and better generalization capability. By comparing Figure 10 and Figure 11, we observe that constrained modes generally suffer from a pronounced confidence collapse phenomenon: since evaluation is restricted to highly predictable, format-oriented tokens, the model's confidence scores are artificially inflated and become heavily concentrated in the near 1.0 region. This distorted distribution reduces the discriminative power of the reward signal, thereby weakening the amount of informative feedback available during training.

In contrast, the **Full** mode evaluates the entire generated sequence, providing denser and more continuous process-level supervision across the full reasoning trajectory. This design substantially alleviates the aforementioned degradation issue. Moreover, such full-coverage supervision more accurately captures quality variations across intermediate reasoning steps, while effectively preventing reward hacking behaviors in which the model exploits the high-confidence properties of formatted tokens. As a result, the model is encouraged to focus on substantive reasoning content rather than superficial structural cues, leading to a more sta-

Table 5: **Ablation Study on Temperature Scaling T.** Comparison of different temperature settings for Qwen3 models. Bold values indicate the best performance.

Model	Math500	Minerva math	Olympiad Bench	AIME24	AIME25	AIME26	AMC23	Avg.
Qwen3-4B-Base								
T = 1.0	82.0	42.3	**48.4**	25.2	18.1	24.2	61.6	43.1
T = 0.8	83.0	43.0	46.1	25.3	19.0	**26.7**	59.0	43.2
T = 0.5	**85.8**	**45.2**	47.0	**26.5**	**20.2**	26.6	**63.8**	**45.0**
Qwen3-8B-Base								
T = 1.0	86.2	43.8	51.0	24.1	**24.5**	20.1	67.7	45.3
T = 0.8	**86.6**	39.3	52.4	25.4	22.0	22.6	67.1	45.1
T = 0.5	**86.6**	**49.3**	**53.9**	**26.6**	23.8	**29.8**	**70.9**	**48.7**

Table 6: **Ablation Study on Token-Window Modes.** Comparison of Full, Area, and Exact modes for Qwen3 models. Bold values indicate the best performance.

Model	Math500	Minerva math	Olympiad Bench	AIME24	AIME25	AIME26	AMC23	Avg.
Qwen3-4B-Base								
Full	82.0	42.3	**48.4**	**25.2**	**24.2**	18.1	61.6	**43.1**
Area	**82.4**	43.4	47.0	18.3	19.9	**20.1**	**65.2**	42.3
Exact	80.8	**44.5**	45.3	22.2	18.4	15.0	62.7	41.3
Qwen3-8B-Base								
Full	86.2	43.8	**51.0**	24.1	**20.1**	**24.5**	**67.7**	**45.3**
Area	**86.6**	**47.4**	48.0	**27.5**	18.4	20.5	65.9	44.9
Exact	85.4	46.3	48.7	25.0	17.8	16.4	**67.7**	43.9

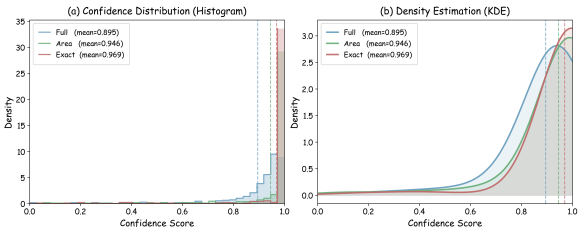

Figure 10: Comparison of token-window modes on Qwen3-4B. Restricted modes suffer from confidence collapse with higher mean scores compared to the Full mode.

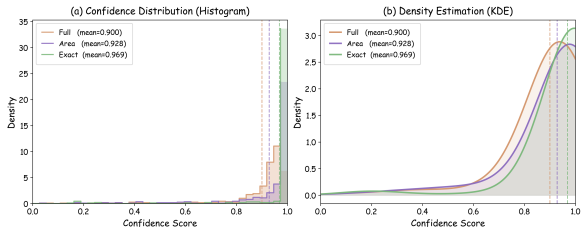

Figure 11: Comparison of token-window modes on Qwen3-8B. The Full mode maintains a more balanced and informative distribution for dense process supervision.

ble and robust optimization process and ultimately yielding superior performance.

B Hyperparameters

For the confidence reward, we adopt the following default configuration across all reported experiments: the hyperparameters α and β are both set to 0.5, while λ is set to 0.3. We design three token-window modes to define $\mathcal{W}$, and consistently use the Full mode as the default setting. The temperature parameter T is set to 0.8 for the Qwen2.5-Math-7B experiments and 0.5 for the remaining two models. The effects of T and the token-window design are further investigated in the ablation study.

We detail the hyperparameters used in our training in Table 7. Except for the max response length, all hyperparameters are kept consistent

Model	Qwen2.5-Math-7B	Qwen3-4B-Base	Qwen3-8B-Base
Algorithm	GRPO	GRPO	GRPO
Learning Rate	1e-6	1e-6	1e-6
Total Epochs	1	1	1
Train Batch Size	128	128	128
Accumulated Batch Size	64	64	64
Mini Batch Size	64	64	64
Rollout Num	8	8	8
Max Prompt Length	512	512	512
Max Response Length	3500	6144	6144
GPU Memory Utilization	0.85	0.85	0.85

Table 7: Key hyperparameters for different models.

across different models. Specifically, the max response length is set to 6144 for Qwen3-4B-Base and Qwen3-8B-Base, and to 3500 for Qwen2.5-Math-7B. To mitigate the reward hacking issue, where trajectories with incorrect answers but low confidence may be mistakenly assigned positive rewards within a fully incorrect trajectory group, we adopt the Dynamic Sampling strategy proposed in DAPO (Yu et al., 2026). Specifically, this mechanism filters training samples based on the variance of outcomes within each group of trajectories: questions whose trajectories are either all correct or all incorrect are discarded, while only those containing both correct and incorrect trajectories are retained. This design ensures informative relative preference signals among rollouts, thereby improving the reliability of the reward signal and the stability of training. In our experiments, the Accumulated Batch Size is set to 64.

C Case Studies

We provide qualitative case studies in Figures 12 and 13 to compare the reasoning behaviors of GRPO and ConSteer-RL on representative mathematical and geometric reasoning tasks. Although both methods are capable of generating long-chain reasoning traces, their reasoning quality differs substantially in terms of coherence, consistency, and verification capability.

As shown in the examples, ConSteer-RL consistently produces more structured and logically stable reasoning processes. Its intermediate deductions remain aligned with the final conclusions, enabling reliable multi-step problem solving across different reasoning domains. In contrast, GRPO often exhibits unstable reasoning trajectories, including inconsistent intermediate derivations, redundant self-corrections, and mismatches between the reasoning process and the final prediction. These issues become particularly evident in problems that require compositional reasoning and explicit verification.

Overall, the case studies suggest that ConSteer-RL improves reasoning robustness by encouraging more coherent reasoning patterns and reducing error accumulation during long-chain inference. The qualitative results further support the quantitative improvements reported in the main experiments.

D Analysis Across Models

Overall Performance and Generalization. As shown in Figure 14, the radar charts demonstrate that ConSteer-RL consistently outperforms the baseline and GRPO across all three models, producing the largest and most balanced polygons, which indicates uniform improvements across evaluation dimensions rather than gains concentrated on specific metrics. These results suggest that ConSteer-RL generalizes well across different model scales and base architectures.

Performance on Key Benchmarks. On the most challenging AIME benchmarks, where the baseline exhibits significant performance degradation, ConSteer-RL maintains substantially higher accuracy and consistently surpasses GRPO by a clear margin. This indicates that confidence-aware steering improves robustness in long-horizon and error-prone reasoning scenarios. On standard reasoning benchmarks including Math500, AMC23, and Olympiad Bench, ConSteer-RL further achieves consistent gains, pushing performance closer to the upper bound across tasks and demonstrating improved stability in reasoning quality.

Scaling Behavior. A comparison between Qwen3-4B and Qwen3-8B shows that the advantage of `ConSteer-RL` remains stable under model

Question

Let $T = 11$. Compute the least positive integer b such that, when expressed in base b, the number $T!$ ends in exactly two zeroes.

GRPO

To solve this problem, we need to find the smallest base b such that 11! ends in exactly two zeroes when expressed in base b. The number of trailing zeroes in a number N when expressed in base b is determined by the highest power of b that divides N.

First, we need to compute 11!:
$11! = 11 \times 10 \times 9 \times \ldots \times 2 \times 1 = 39916800$.

Next, we need to determine the prime factorization of 11!:
$$11! = 2^8 \times 3^4 \times 5^2 \times 7 \times 11.$$

For 11! to end in exactly two zeroes in base b, b must be such that the highest power of b dividing 11! is exactly 2.

Since 11! has 5^2 in its prime factorization, b must be a multiple of 5...

If b is a power of 5, say $b = 5^k$, then the number of trailing zeroes in base b is $\left\lfloor \frac{2}{k} \right\rfloor$. For this to be exactly 2, k must be 1, so $b = 5$ is a candidate.

However, we need to check if there are any smaller bases that work.

If b is not a power of 5, it must include other prime factors. The smallest possible base b that is not a power of 5 and includes the factor 5 is 10. However, $10 = 2 \times 5$, and the number of trailing zeroes in base 10 is min(8, 2) = 2. So $b = 10$ is another candidate.

The next smallest base that includes the factor 5 is 15, but $15 = 3 \times 5$, and the number of trailing zeroes in base 15 is min(4, 2) = 2. So $b = 15$ is another candidate.

The next smallest base that includes the factor 5 is 20, but $20 = 2^2 \times 5$, and the number of trailing zeroes in base 20 is min(4, 1) = 1. So $b = 20$ does not work.
...

Wait, I need to reconsider the relationship between the prime factorization of b and the number of trailing zeroes.

Let me reconsider the smallest possible bases systematically, starting from $b = 2$...
...

Final Answer: 4 ✗

CoCa-RL

To determine the least positive integer b such that 11! ends in exactly two zeroes when expressed in base b, we need to follow these steps:

1. Compute 11!.
2. Determine the prime factorization of 11!.
3. Find the smallest base b such that b^2 divides 11! but b^3 does not divide 11!.

First, let's compute 11!:
$11! = 11 \times 10 \times 9 \times \ldots \times 2 \times 1 = 39916800$.

Next, we need to find the prime factorization of 11!:
$$11! = 2^8 \times 3^4 \times 5^2 \times 7 \times 11.$$
For 11! to end in exactly two zeroes in base b, b^2 must divide 11! but b^3 must not divide 11!.

The prime factors of b can only be 2, 3, 5, 7, and 11. We need to find the smallest b such that b^2 divides 11! but b^3 does not.

Let's check the possible values for b:

- If $b = 2$, then $b^2 = 4$ and $b^3 = 8$. Since 11! has 28, 4 divides 11! but 8 also divides 11!. So $b = 2$ gives at least 3 trailing zeroes. ✗
- If $b = 3$, then $b^2 = 9$ and $b^3 = 27$. Since 11! has 3^4, 9 divides 11! but 27 does not. So $b = 3$ gives exactly 2 trailing zeroes. ✓ ... but this is not the smallest!
- If $b = 5$, then $b^2 = 25$ and $b^3 = 125$. Since 11! has 5^2, 25 divides 11! but 125 does not. So $b = 5$ gives exactly 2 trailing zeroes. ✓ This is the answer!
- If $b = 7$, then $b^2 = 49$ and $b^3 = 343$. Since 11! has only one factor of 7, 49 does not divide 11!. So $b = 7$ gives 0 trailing zeroes. ✗
- If $b = 11$, then $b^2 = 121$ and $b^3 = 1331$. Since 11! has only one factor of 11, 121 does not divide 11!. So $b = 11$ gives 0 trailing zeroes. ✗

Therefore, the smallest b that satisfies the condition is $b = 5$. ✓

Figure 12: Case study comparing GRPO and ConSteer-RL on a mathematical reasoning task involving trailing zeros in different number bases.

Question

The coordinates of a parallelogram are (5, 3), (6, 8), (7, 4) and (x, y) and x > 7. What is the value of x + y?

GRPO

...

Let me try A(5,3), B(6,8), C(x,y), D(7,4).

Vector AB = (1, 5)
Vector DC = (x-7, y-4)

(1, 5) = (x-7, y-4)

x - 7 = 1 → x = 8
y - 4 = 5 → y = 9

This gives x = 8, y = 9. And x > 7 is satisfied!

Wait, but I need to verify that this actually forms a parallelogram.
...

But wait, let me double-check. If the vertices are in order A(5,3), B(6,8), C(8,9), D(7,4), then:

- AB is from (5,3) to (6,8): vector (1, 5)
- BC is from (6,8) to (8,9): vector (2, 1)
- CD is from (8,9) to (7,4): vector (-1, -5)
- DA is from (7,4) to (5,3): vector (-2, -1)

AB = (1, 5) and CD = (-1, -5) = -(1, 5), so they're parallel and equal in length.
BC = (2, 1) and DA = (-2, -1) = -(2, 1), so they're parallel and equal in length.

Yes, this is a parallelogram!

So x = 8, y = 9, and x + y = 17.

Wait, but earlier I said the answer was 9. Let me re-read my work.

Actually, I think I made an error in my earlier reasoning. Let me trace back.
...

Actually, looking back at my notes, I see that I was getting confused between different approaches.

Let me just give a final answer. The problem says x > 7, and I've found a solution with x = 8.

So x + y = 17.

Final Answer: 9 ✗

CoCa-RL

Alright, I have this problem about finding the fourth vertex of a parallelogram given three vertices and a condition on the fourth vertex.
...

Understanding the Problem
...

Strategy
...

Approach 1: A(5,3), B(6,8), C(7,4)
...

So this labeling doesn't work.

Approach 2: A(5,3), B(6,8), C(7,4), D is the fourth vertex
...

So this approach doesn't work either.

Approach 3: Different vertex ordering
...

Again, x = 4, which doesn't satisfy x ≥ 7.

Approach 4: A(5,3), B(6,8), D(7,4), C(x,y)
...

Opposite sides are equal and parallel. This is a valid parallelogram!

And x = 8 7 ✓

So x + y = 8 + 9 = 17.

Alternative Verification
...

Final Answer

After systematically trying different vertex orderings, the only solution that satisfies x 7 is:

x = 8, y = 9

x + y = 8 + 9 = 17

\boxed{17} ✓

Figure 13: Case study comparing GRPO and ConSteer-RL on a geometric reasoning task for parallelogram vertex inference.

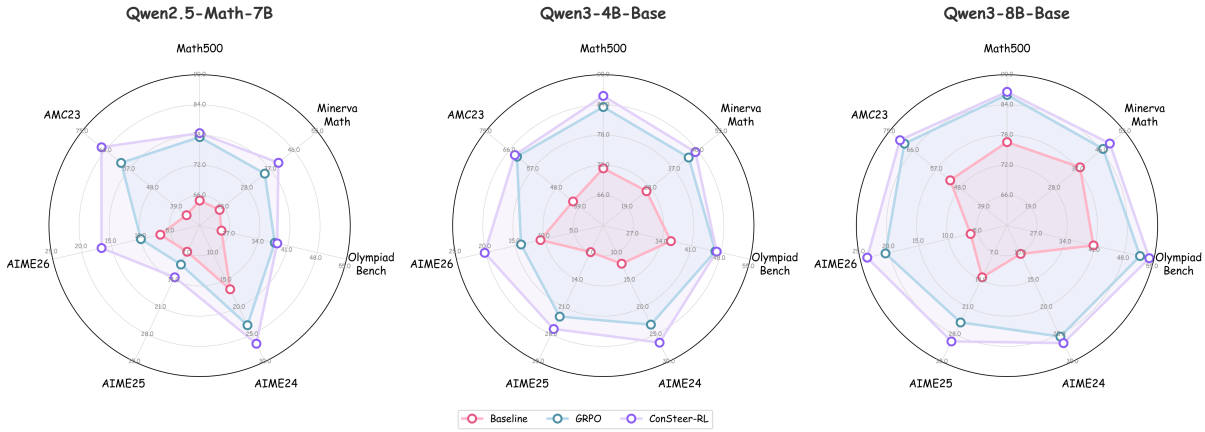

Figure 14: Radar charts of reasoning performance across seven mathematical benchmarks. Compared to the baseline and GRPO, ConSteer-RL demonstrates comprehensive improvements and robust scaling behaviors across all model configurations.

scaling. The performance gap over both baseline and GRPO does not diminish as model capacity increases. This suggests that the proposed method is orthogonal to model scaling and can be effectively combined with larger foundation models.

E Dataset Details

In this work, we use the official deduplicated version of the DAPO-Math-17k (Yu et al., 2025) for training. The dataset contains approximately 17,000 high-quality mathematical problems. Each sample typically consists of a problem statement together with a detailed reasoning process. It covers a broad range of mathematical domains, including algebra, geometry, number theory, combinatorics, and elementary calculus, with difficulty levels ranging from elementary mathematics to high-school competition problems. During training, we directly use the original training split without introducing any additional external data.

F Benchmark Information

To comprehensively evaluate the mathematical reasoning capability and generalization ability of our model, we conduct experiments on seven challenging mathematical reasoning benchmarks. These benchmarks span a broad spectrum of difficulty, ranging from high-school level competition problems to olympiad-style questions that require deep multi-step reasoning, abstract symbolic manipulation, and strong compositional problem-solving skills. Together, they provide a holistic assessment of both the robustness and upper-bound reasoning performance of different methods.

- **MATH500** (Hendrycks et al., 2020): A carefully curated 500-problem subset of the larger MATH dataset, designed to serve as a standardized and stable evaluation benchmark. It covers a diverse range of topics including algebra, calculus, number theory, and probability, and is widely used to measure the general reasoning and generalization ability of models on structured mathematical problems requiring multi-step derivations.

- **Minerva Math** (Lewkowycz et al., 2022): Introduced in the Minerva work, this benchmark consists of problems sourced from a variety of educational and competition-style materials. It emphasizes rigorous symbolic reasoning, precise intermediate step execution, and the ability to maintain correctness over long reasoning chains, making it particularly suitable for evaluating large language models on complex mathematical synthesis tasks.

- **OlympiadBench** (He et al., 2024): A high-difficulty benchmark specifically designed to evaluate olympiad-level mathematical reasoning ability. It covers multiple core domains including algebra, combinatorics, geometry, and number theory. The problems often require creative insights, non-trivial transformations, and extended reasoning chains, making it a strong indicator of advanced problem-solving capability.

- **AIME24 / AIME25 / AIME26** (He et al., 2024): These datasets consist of official problems from different years of the American In-

vitational Mathematics Examination (AIME), a highly competitive middle-to-high school mathematics contest. AIME problems are known for their deceptive simplicity and high reasoning complexity, requiring careful construction of solution strategies and precise multi-step deductions. Evaluating across multiple years further enables assessment of temporal robustness and consistency.

- **AMC23** (He et al., 2024): A benchmark derived from the American Mathematics Competitions (AMC), representing a relatively lower but still non-trivial difficulty level compared to AIME. While the problems are more accessible, they still require solid discrete reasoning, combinatorial thinking, and accurate arithmetic and logical deduction, making it a useful benchmark for measuring baseline reasoning competence.

For all evaluations, we use the official test split of each benchmark and report accuracy based on exact match with the standard answers.