REIN: BRIDGING THE GAP BETWEEN REASONING AND RELIABILITY VIA REFLECTION AND ABSTENTION ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Large reasoning models (LRMs) are prone to hallucination, which undermines their reliability and poses challenges for safe deployment. Hallucinations in LRMs arise from two distinct failure sources: reasoning hallucination, where flawed inference steps propagate to an incorrect conclusion, and knowledge hallucination, where the model lacks the requisite factual knowledge to answer the query. To address reasoning hallucination, we propose REIN, an alignment framework that trains LRMs to produce a structured reasoning sequence, `<think>` $\rightarrow$ `<reflection>` $\rightarrow$ `<answer>`, enabling explicit self-reflection before committing to a final answer. To address knowledge hallucination, REIN introduces a reward mechanism that encourages explicit abstention (e.g., "I don't know") when none of the sampled reasoning chains yields a correct answer, allowing the model to refrain from unsupported predictions. Extensive evaluations on mathematical and commonsense reasoning benchmarks show that REIN consistently improves selective accuracy, reduces incorrect-but-self-endorsed responses, and maintains high coverage compared with competitive baselines. Notably, REIN achieves these gains within a single forward pass, without requiring process supervision, inference-time controllers, external search, or multi-round critiques. Experiments on multiple backbones show that REIN reduces the hallucination proxy by $58 \sim 72\%$ relative to the base models while maintaining $86 \sim 91\%$ average coverage, and improves selective accuracy on attempted questions by $6.6 \sim 14.2\%$.

1 INTRODUCTION

Large reasoning models (LRMs) such as DeepSeek-R1 (DeepSeek-AI, 2025) and Qwen3 (Qwen Team, 2025) have made substantial progress on mathematics, scientific reasoning, and programming (Hendrycks et al., 2021a; Zheng et al., 2025; Chen et al., 2025) by eliciting explicit chains of thought (Wei et al., 2022). Yet explicit reasoning does not eliminate hallucination—it often increases the apparent plausibility of unsupported conclusions, producing outputs that seem coherent and persuasive but still contain factual errors (Min et al., 2023) or internally inconsistent inference (Cheng et al., 2025).

We identify two primary failure modes of hallucination: reasoning-level hallucination and knowledge-level hallucination. In *reasoning-level hallucination*, the model possesses the requisite knowledge, but a particular sampled trajectory contains a flawed inference step that propagates to a wrong answer. This failure mode is self-recoverable, since a different sample or a verified self-check could still yield the correct answer. In *knowledge-level hallucination*, by contrast, the model lacks the requisite knowledge under the current policy and sampling budget, and any additional reasoning merely rationalizes an unsupported prediction. Thus, additional reasoning or reflection cannot rescue the answer, and the only safe action is to abstain. These two modes demand fundamentally different interventions, and conflating them is a central source of mis-calibration in current alignment recipes.

Existing approaches address at most one of the two failure modes. Outcome-level preference alignment such as RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2023) optimizes the final response but provides no explicit supervision over the reasoning trajectory, so reasoning-level errors that propagate to the final answer remain undetected. Process-supervision and self-correction

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

methods, including step-level process reward models (Uesato et al., 2022; Lightman et al., 2024), Self-Refine (Madaan et al., 2023), and Reflexion (Shinn et al., 2023), focus on reliability assessment but presume that the errors are self-recoverable—they have no mechanism to distinguish a fixable inference slip from genuinely missing knowledge, and recent evidence shows that without external feedback, models often fail to detect their own errors or even overturn correct answers (Huang et al., 2023; Kamoi et al., 2024). Conversely, abstention and uncertainty-calibration methods such as IDK Prompting, R-Tuning (Zhang et al., 2024), BARREL (Yang et al., 2026b), and TruthRL (Wei et al., 2026) encourage refusal under uncertainty but do not explicitly target reasoning repair, potentially abstaining when an error remains recoverable. S^2R (Ma et al., 2025) trains self-verification and correction, but does not jointly address when the model should stop repairing and abstain. It remains an open question how to establish a unified, reliability-oriented alignment framework that can decide, within a single completion, whether the current reasoning supports a reliable answer or whether the model should abstain.

We propose **REIN**, an alignment framework that teaches an LRM, in a single forward pass, to reflect on its own reasoning before committing to an answer and to abstain when the answer is not adequately supported by the model's internal knowledge. REIN trains the model to emit a structured completion `<think>` → `<reflection>` → `<answer>`, where the `<reflection>` span is a trainable reliability judgment whose stance is aligned, via a centered reflection-veracity reward, with the verified correctness of the answer. To handle knowledge-level failures, REIN further derives a group-level boundary indicator from the model's own sampled completions and uses a boundary-aware IDK reward that rewards abstention only when no sample in the group yields a (nearly) correct answer—thereby separating recoverable reasoning failures from genuine knowledge gaps without process supervision, external retrieval, or test-time controllers.

Our contributions are threefold:

1. We propose REIN, a unified single-pass alignment framework that structures generation as `<think>` → `<reflection>` → `<answer>`, enabling LRMs to decide within one forward pass whether the current reasoning should be trusted for finalization or whether the model should abstain, without external controllers or iterative loops.
2. We design a centered reflection-veracity reward to align the reflection span with the correctness of the pre-reflection draft conclusion, along with a boundary-aware IDK reward to incentivize abstention when the model lacks sufficient knowledge.
3. Extensive experiments across multiple backbones and reasoning benchmarks demonstrate that REIN reduces the hallucination proxy by 58–72% while maintaining 86–91% average coverage, and improves accuracy on attempted questions by 6.6–14.2%.

2 Related Work

Self-correction and reflection. Self-correction methods push the LLM to self-critique and revise its own output. Self-Refine is an inference-time framework that iteratively alternates between model-generated feedback and output revision (Madaan et al., 2023). SCoRe instead learns multi-turn self-correction from on-policy traces through reinforcement learning (Kumar et al., 2025). Evidence on intrinsic self-correction nevertheless shows that self-generated feedback does not consistently correct errors and may overturn correct answers when no reliable external signal is available (Huang et al., 2023; Kamoi et al., 2024; Li et al., 2024). REIN assigns reflection a different role: the `<reflection>` span assesses whether the preceding reasoning trajectory supports a reliable draft conclusion. Its judgment is aligned with the verified correctness of that pre-reflection conclusion and guides finalization within the same structured completion, without requiring a separate critique–revision loop.

Outcome and process alignment. Preference-based RL such as RLHF (Ouyang et al., 2022), Safe RLHF (Dai et al., 2023), and DPO (Rafailov et al., 2023) aligns model behavior to outcome-level preferences but provides no supervision on the reasoning trajectory itself. Rejection-sampling fine-tuning and verifier-driven RL extend this to verifiable-answer domains via automatic answer checking (Yuan et al., 2023a; Shao et al., 2024; DeepSeek-AI, 2025), while process-reward models supply step-level signals at the cost of dense human or LLM annotations (Uesato et al., 2022;

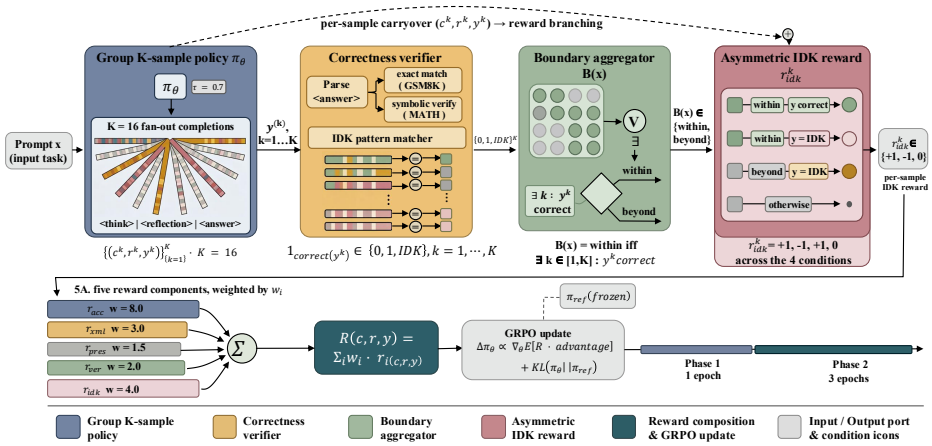

Figure 1: Overview of the REIN training pipeline: reflection-augmented completions feed into GRPO with reflection and boundary-aware abstention rewards.

Lightman et al., 2024). Unlike outcome-only preference RL, which is silent about the reasoning trajectory and process-supervised reward models, which require fine-grained step labels and still cannot decide when to abstain. REIN keeps the GRPO outer loop but redesigns the reward family around *reliability*: a reflection-veracity reward aligns the model's self-assessment with answer correctness, and a boundary-aware IDK reward injects an abstention signal, all without step-level supervision.

Uncertainty and abstention. R-Tuning and IDK-token methods train models to abstain under uncertainty (Zhang et al., 2024; Cohen et al., 2024; Cheng et al., 2024). TruthRL instead uses a ternary RL objective that distinguishes correct answers, hallucinations, and abstentions (Wei et al., 2026). REIN differs by separately aligning a pre-finalization reflection with the correctness of the draft conclusion implied by the reasoning trajectory, and the final answer-or-abstain decision with a sampled knowledge boundary. KnowRL introduces fine-grained factual supervision over atomic claims in the reasoning trace (Ren et al., 2026), whereas REIN requires only task-answer verification. BAPO studies boundary-aware rewards for agentic search (Liu et al., 2026), while BARREL promotes concise, boundary-aware reasoning to mitigate overthinking (Yang et al., 2026a). In contrast, REIN jointly aligns reflection and abstention within a single structured completion.

3 REIN Framework

3.1 Framework Setup

Given a prompt $x \in \mathcal{X}$, the policy generates a structured completion $z \sim \pi_\theta(\cdot \mid x)$ that follows the order

$$\texttt{<think>} \rightarrow \texttt{<reflection>} \rightarrow \texttt{<answer>}, \tag{1}$$

from which a deterministic parser extracts the three spans $(c, r, y_1) := \mathcal{P}(z)$. The reasoning trajectory may imply a preliminary conclusion before reflection; we denote the draft conclusion extracted from c as $y_0 := \mathcal{D}_x(c)$, where $\mathcal{D}_x$ is a task-specific deterministic answer extractor.

Here, c is the pre-reflection reasoning trajectory, y_0 is the draft conclusion implied by that trajectory, r is the model's reliability assessment conditioned on the preceding trajectory, and y_1 is the final answer generated after reflection. The reflection assesses whether the draft answer y_0 implied by the preceding reasoning is reliable. The final answer y_1 is either a substantive task answer in $\mathcal{Y}$ or the

canonical abstention response `IDK`. Because r is generated after the reasoning trajectory but before y_1,it influences the subsequent finalization decision.

For any candidate answer y, we define

$$T(x, y) := \begin{cases} 1, & \text{if } y \neq \text{IDK and } \mathcal{V}_x(y, y^\star(x)) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The draft answer y_0 and the final answer y_1 are verified for different purposes. The reflection-veracity reward evaluates whether the reflection judgment agrees with $T(x, y_0)$, whereas the answer-correctness and boundary-aware abstention rewards evaluate $T(x, y_1)$. Answers mentioned elsewhere in the reflection span are not treated as task answers. A final answer $y_1 = \texttt{IDK}$ is not counted as a correct task answer and is handled separately by the abstention reward. Malformed structured outputs are handled by the format rewards described in Section 3.4.

3.2 Reflection Alignment

Before generating the final answer, REIN uses the `<reflection>` span to assess whether the preceding reasoning trajectory supports a reliable draft conclusion y_0. After the completion is generated, REIN verifies the draft conclusion y_0 and rewards the reflection when its reliability judgment matches $T(x, y_0)$. The post-reflection final answer y_1 is not used as the target of the reflection-veracity reward. This distinction prevents a correct assessment of an unreliable draft trajectory from being penalized when the model subsequently revises its final answer.

A fixed parser maps the reflection r to

$$s(r) \in \{\texttt{correct}, \texttt{wrong}, \texttt{uncertain}, \bot\},$$

where $\bot$ denotes a missing, invalid, or ambiguous stance. For a valid stance, we define the binary reflection judgment as

$$J(r) := \begin{cases} 1, & s(r) = \texttt{correct}, \\ 0, & s(r) \in \{\texttt{wrong}, \texttt{uncertain}\}. \end{cases} \tag{3}$$

Here, $J(r) = 1$ indicates that the model considers the conclusion implied by the preceding reasoning trajectory reliable, while $J(r) = 0$ indicates that it considers the trajectory unreliable.

The main failure targeted by reflection alignment is false endorsement:

$$\mathrm{FE}(x, r, y_0) := \mathbb{1}[y_0 \in \mathcal{Y}]\big(1 - T(x, y_0)\big)J(r). \tag{4}$$

False endorsement occurs when the draft conclusion implied by the reasoning trajectory is incorrect but the reflection nevertheless assigns it a reliable stance. Such over-confidence may cause the model to preserve an unreliable trajectory during finalization.

The opposite mismatch is false rejection:

$$\mathrm{FR}(x, r, y_0) := T(x, y_0)\big(1 - J(r)\big). \tag{5}$$

False rejection occurs when a reliable draft conclusion is judged unreliable. Although the final answer y_1 may still be correct, this mismatch can lead to unnecessary revision or abstention.

The reflection-veracity reward is defined as

$$r_{\mathrm{vrcty}}(x, r, y_0) := \begin{cases} +1, & s(r) \neq \bot \ \wedge \ J(r) = T(x, y_0), \\ -1, & s(r) \neq \bot \ \wedge \ J(r) \neq T(x, y_0), \\ 0, & s(r) = \bot. \end{cases} \tag{6}$$

Invalid or missing reflection stances receive no veracity reward and are handled by the format rewards described in Section 3.4.

The reflection reward checks whether the parsed stance agrees with the correctness of y_0. It does not score individual reasoning steps. It does not perform step-level verification or identify which reasoning step causes an error. The correctness of the final answer y_1 is supervised separately by the answer-correctness reward, while the final answer-or-abstain decision is further guided by the group-level signal introduced next.

3.3 BOUNDARY-AWARE ABSTENTION ALIGNMENT

Reflection alignment assesses whether a single reasoning trajectory supports a reliable draft conclusion y_0, but it cannot determine whether the current policy can produce a verified final answer y_1 for the prompt. REIN therefore uses a group of on-policy completions to obtain an empirical estimate of the current policy's knowledge boundary.

For each prompt x, the current policy samples K completions $z^{(k)} \sim \pi_\theta(\cdot \mid x)$, $k = 1, \ldots, K$. Applying $\mathcal{P}$ and $\mathcal{D}_x$ to each sampled completion yields $\mathcal{G}_K(x) := \left((c^{(k)}, y_0^{(k)}, r^{(k)}, y_1^{(k)})\right)_{k=1}^{K}$, where $(c^{(k)}, r^{(k)}, y_1^{(k)}) = \mathcal{P}(z^{(k)})$ and $y_0^{(k)} = \mathcal{D}_x(c^{(k)})$.

Here, $\mathcal{G}_K(x)$ denotes the policy-gradient group, sampled under the prompt used in the current training phase. To estimate the boundary, REIN uses completions sampled under an IDK-free prompt. In Phase 1, the policy-gradient group already uses this prompt and can be reused. In Phase 2, we independently sample K_b boundary-probe completions from the same current policy under the IDK-free prompt. Let $\tilde{y}_1^{(j)}$ denote the final answer of the j-th boundary-probe completion.

The forced-answer boundary is defined as

$$\widehat{B}_{K_b}^{\text{forced}}(x) := \begin{cases} \texttt{within}, & \exists j \in \{1, \ldots, K_b\} : T(x, \tilde{y}_1^{(j)}) = 1, \\ \texttt{beyond}, & \text{otherwise.} \end{cases} \tag{7}$$

The estimated boundary depends on the current policy, the boundary-probe group size K_b, and the decoding configuration. It is therefore an empirical, policy-relative estimate rather than an oracle label of intrinsic knowledge absence. Larger probe groups reduce finite-sampling uncertainty at additional rollout cost. We use $K_b = 16$ in the main experiments; Appendix F reports the sensitivity to this choice, and Appendix H gives its finite-sample interpretation.

REIN uses this boundary to supervise the final answer-or-abstain decision in the policy-gradient completions. When $\widehat{B}_{K_b}^{\text{forced}}(x) = \texttt{within}$, the boundary-probe group contains at least one verified-correct final answer $\tilde{y}_1^{(j)}$. REIN therefore rewards verified-correct final answers and penalizes unnecessary abstention in the policy-gradient group. When $\widehat{B}_{K_b}^{\text{forced}}(x) = \texttt{beyond}$, REIN rewards $y_1^{(k)} = \texttt{IDK}$ instead of an unverified substantive final answer.

For completion k, the boundary-aware abstention reward is

$$r_{\text{idk}}^{(k)} := \begin{cases} +1, & \widehat{B}_{K_b}^{\text{forced}}(x) = \texttt{within} \wedge T(x, y_1^{(k)}) = 1, \\ -1, & \widehat{B}_{K_b}^{\text{forced}}(x) = \texttt{within} \wedge y_1^{(k)} = \texttt{IDK}, \\ +1, & \widehat{B}_{K_b}^{\text{forced}}(x) = \texttt{beyond} \wedge y_1^{(k)} = \texttt{IDK}, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

We detect abstention from the parsed <answer> span y_1 using a fixed set of IDK patterns. The same boundary label is used for all policy completions sampled for prompt x. A single failed completion does not by itself justify abstention when another rollout for the same prompt has already produced a verified-correct final answer. Incorrect substantive final answers receive no positive reward from r_{idk} and are evaluated separately by r_{acc}. The reflection-veracity reward separately supervises whether the reflection r correctly assesses the reliability of the corresponding reasoning trajectory and its implied draft conclusion y_0.

3.4 JOINT REWARD AND GRPO OPTIMIZATION

REIN first applies a short supervised fine-tuning stage to initialize the structured output format and the reflection labels used to assess the reliability of the pre-reflection reasoning trajectory and its implied draft conclusion y_0. The resulting policy is then optimized with GRPO over the on-policy completion groups defined in Section 3.3.

or each policy completion k, we sum five reward terms: answer correctness, XML structure, reflection presence, reflection veracity, and boundary-aware abstention:

$$R^{(k)}(x) = w_{\text{acc}} r_{\text{acc}}^{(k)} + w_{\text{xml}} r_{\text{xml}}^{(k)} + w_{\text{pres}} r_{\text{pres}}^{(k)} + w_{\text{vrcty}} r_{\text{vrcty}}^{(k)} + w_{\text{idk}} r_{\text{idk}}^{(k)}. \tag{9}$$

For completion k, the answer-correctness reward $r_{\text{acc}}^{(k)}$ evaluates the post-reflection final answer $y_1^{(k)}$ and rewards it when it is verified correct. The format rewards $r_{\text{xml}}^{(k)}$ and $r_{\text{pres}}^{(k)}$ enforce the required tag structure and the presence of a non-empty reflection span, respectively. The reflection-veracity reward $r_{\text{vrcty}}^{(k)}$ aligns the reflection judgment $r^{(k)}$ with the verified correctness of the corresponding draft conclusion $y_0^{(k)}$ implied by the preceding reasoning trajectory. The boundary-aware reward $r_{\text{idk}}^{(k)}$ aligns the final answer-or-abstain action expressed in $y_1^{(k)}$ with the group-level sampled knowledge boundary.

Within GRPO, relative advantages are computed across the K policy-gradient completions for the same prompt. In Phase 2, the separately sampled boundary-probe completions are used to construct the boundary label, rather than being included as additional members of the policy-gradient group:

$$\widehat{A}^{(k)} = \frac{R^{(k)} - \text{mean}_j\, R^{(j)}}{\text{std}_j\, R^{(j)} + \epsilon}. \tag{10}$$

REIN then applies the standard GRPO objective with KL regularization relative to the reference policy. While $r_{\text{vrcty}}^{(k)}$ supervises whether the reflection correctly assesses the reliability of the pre-reflection reasoning trajectory, $r_{\text{acc}}^{(k)}$ and $r_{\text{idk}}^{(k)}$ supervise final-answer correctness and abstention in $y_1^{(k)}$, respectively. All reward components contribute to the same scalar reward and update the same policy. Therefore, trajectory reliability assessment and final answer-or-abstain decisions are optimized jointly.

REIN uses a two-phase GRPO curriculum. In Phase 1, the policy is trained under an IDK-free prompt with $w_{\text{idk}} = 0$. The policy-gradient group therefore already satisfies the forced-answer condition and can be reused to estimate the boundary without additional rollouts.

In Phase 2, the policy prompt enables abstention and w_{idk} is set to 4.0. The policy-gradient group contains $K = 16$ IDK-enabled completions, while an independent group of $K_b = 16$ completions is sampled under the IDK-free prompt to estimate $\widehat{B}_{K_b}^{\text{forced}}(x)$. Answer correctness, format, and reflection-veracity rewards remain active in both phases.

At inference time, REIN generates a single structured completion without reference answers, task verifiers, group sampling, or multi-round critique–revision loops. Within this single completion, the trained policy forms a draft conclusion y_0 from the reasoning trajectory, assesses its reliability through r, and produces a final answer y_1 that is either substantive or `IDK`.

4 Experimental Setup

Datasets and Evaluation Metrics. We evaluate all methods on four benchmarks: GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021b) for mathematical reasoning, and StrategyQA (Geva et al., 2021) and ARC-Challenge (Clark et al., 2018) for commonsense reasoning. GRPO optimization uses only the GSM8K training set; results on the other three benchmarks therefore measure the cross-task transfer of the learned reliability behavior.

We report four primary metrics: Selective Accuracy (**Sel.Acc.**), Hallucination Proxy (**H-Proxy**), Coverage (**Cov.**), and Effective Accuracy (**Eff.Acc.**).

Sel.Acc. measures correctness among substantive final answers. H-Proxy measures the proportion of incorrect pre-reflection draft conclusions y_0 that are nevertheless positively endorsed by the corresponding reflection r. Specifically, it estimates $\Pr[J(r) = 1 \mid T(x, y_0) = 0]$, thereby measuring draft-level false endorsement rather than the overall incidence of incorrect final answers. Cov. is the proportion of evaluation examples for which the model produces a substantive final answer, while Eff.Acc. is the proportion of all examples that are answered correctly. Higher Sel.Acc., Cov., and Eff.Acc. are better, whereas lower H-Proxy is better. Detailed definitions of these metrics are provided in Appendix A.

Models and Baselines. We evaluate REIN on Qwen2.5-7B (Qwen Team, 2024) and Llama-3.1-8B (Dubey et al., 2024) to test generalization across model families. Training-free baselines are Base, Self-Refine (Madaan et al., 2023), and IDK Prompting. Fine-tuning baselines are RFT (Yuan

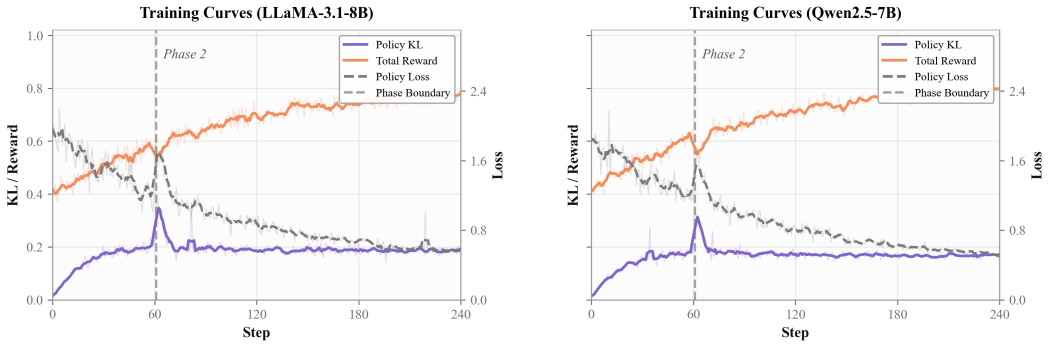

Figure 2: Training curves of KL divergence, reward, and loss. KL remains low and stable, while reward steadily increases and loss decreases throughout training.

et al., 2023b) and R-Tuning (Zhang et al., 2024). RL-based baselines are S^2R (Ma et al., 2025), BARREL (Yang et al., 2026b), and TruthRL (Wei et al., 2026). We also include Reflection-only to isolate reflection alignment. Training-based baselines are adapted from their original objectives, and all methods use the same test splits and evaluation protocol. For H-Proxy, each method generates `<think>` $\rightarrow$ `<reflection>` $\rightarrow$ `<answer>` in a single completion, using its own reflection to judge the draft y_0 without a post-hoc judging call.

We additionally evaluate REIN on Mistral-7B-v0.3 (Mistral AI, 2024) and DeepSeek-R1-Distill-8B (DeepSeek-AI, 2025), and observe consistent improvements. The corresponding results and further implementation details are provided in Appendix C.

Training Details. Unless otherwise stated, all backbones use the same training configuration. We first perform a short SFT stage to initialize the structured output format and reflection behavior, followed by GRPO training on GSM8K. For each prompt, the policy-gradient group contains $K = 16$ completions in both phases. Sampling uses a maximum length of 2048 and temperature 0.7. We use LoRA with $r = 16$ and $\alpha = 32$, and set the KL coefficient to 0.05.

The IDK-alignment term is inactive in Phase 1 ($w_{\text{idk}} = 0$) and activated in Phase 2 ($w_{\text{idk}} = 4.0$). Training follows the two-phase IDK curriculum described in Section 3.4. Its 16+16 rollouts include an IDK-free boundary probe sampled only in Phase 2, whose marginal cost is three quarters of a full group. At evaluation, all models use single-pass greedy decoding without self-consistency or multi-round revision.

5 Main Results

Overall results. Table 1 reports the main comparison on Qwen2.5-7B and Llama-3.1-8B across four benchmarks. Macro-averaged over the eight backbone–benchmark pairs, REIN obtains 78.7% selective accuracy, 12.0% H-Proxy, 90.5% coverage, and 71.7% effective accuracy. REIN achieves the highest selective accuracy and the lowest H-Proxy in every reported cell. Relative to the strongest external baseline in each cell, the selective-accuracy margin ranges from 0.4 to 4.8 percentage points, while H-Proxy is lower by 1.8–12.0 points. The pattern is also stable across backbones: the four-task macro selective accuracy is 78.6% on Qwen2.5-7B and 78.7% on Llama-3.1-8B, with corresponding H-Proxy values of 11.3% and 12.8%.

Relative to the Base checkpoints, REIN improves selective accuracy by 14.0 percentage points and effective accuracy by 7.0 points, while reducing H-Proxy by 28.5 points on average. The 9.6-point reduction in coverage therefore does not translate into lower overall correctness relative to the unaligned models.

Comparison with different methods. Compared with IDK Prompting, REIN reduces H-Proxy by 12.6 percentage points while improving coverage by 6.4 points and effective accuracy by 8.0 points. Compared with R-Tuning, REIN improves selective accuracy by 6.3 points and effective

Table 1: Comparison of reliability alignment methods. Sel.Acc. is accuracy among answered examples, H-Proxy is the fraction of incorrect pre-reflection drafts endorsed as correct by the corresponding reflection., and Eff.Acc. accuracy over all examples. Coverage below 100% is given as a subscript of Eff.Acc.; omitted entries indicate full coverage. Higher Sel.Acc./Eff.Acc. and lower H-Proxy are better.

Method	Mathematical Reasoning						Commonsense Reasoning					
	GSM8K			MATH			StrategyQA			ARC-Challenge		
	Sel.Acc.↑	H-Proxy↓	Eff.Acc.↑	Sel.Acc.↑	H-Proxy↓	Eff.Acc.↑	Sel.Acc.↑	H-Proxy↓	Eff.Acc.↑	Sel.Acc.↑	H-Proxy↓	Eff.Acc.↑
Backbone: Qwen2.5-7B												
Training-free												
Base	79.1	41.0	79.1	41.2	45.0	41.2	64.3	38.0	64.3	74.8	36.0	74.8
Self-Refine	78.4	39.0	78.4	40.6	44.0	40.6	65.1	36.0	65.1	74.9	35.0	74.9
IDK Prompting	86.8	22.5	$78.9_{(90.9)}$	54.1	31.1	$38.7_{(71.5)}$	73.4	22.0	$60.2_{(82.0)}$	84.2	20.8	$76.7_{(91.1)}$
Fine-tuning												
RFT	84.7	34.0	84.7	47.9	39.0	47.9	68.9	32.0	68.9	78.6	30.0	78.6
R-Tuning	84.2	13.5	$68.8_{(81.7)}$	51.2	27.5	$40.7_{(79.4)}$	64.3	12.4	$57.5_{(89.4)}$	86.1	17.2	$82.0_{(95.2)}$
RL-based												
S^2R	88.5	31.0	88.5	53.8	37.0	**53.8**	74.2	28.0	**74.2**	79.8	27.0	79.8
BARREL	87.4	21.0	$79.7_{(91.2)}$	56.2	27.0	$44.7_{(79.6)}$	73.5	20.0	$64.8_{(88.1)}$	84.1	19.0	$76.0_{(90.4)}$
TruthRL	86.9	17.9	$80.6_{(92.7)}$	57.3	42.7	$48.7_{(85.0)}$	67.7	26.9	$62.7_{(92.6)}$	83.7	10.8	$77.5_{(92.6)}$
Ours												
Reflection-only	85.2	18.9	**85.2**	51.2	22.1	51.2	70.3	17.5	**70.3**	82.1	16.2	**82.1**
REIN	**90.6**	**11.0**	$85.0_{(93.8)}$	**61.0**	**15.0**	$50.9_{(83.5)}$	**76.2**	**10.0**	$69.2_{(90.8)}$	**86.5**	**9.0**	$80.9_{(93.5)}$
Backbone: LLaMA-3.1-8B												
Training-free												
Base	76.8	44.0	76.8	38.5	48.0	38.5	66.7	37.0	66.7	76.2	35.0	76.2
Self-Refine	76.1	42.0	76.1	37.8	47.0	37.8	67.4	35.0	67.4	76.8	34.0	76.8
IDK Prompting	85.0	25.2	$74.2_{(87.3)}$	51.8	33.9	$36.8_{(71.0)}$	74.8	21.1	$67.4_{(90.1)}$	86.2	20.5	$76.2_{(88.4)}$
Fine-tuning												
RFT	83.2	36.0	83.2	45.1	42.0	45.1	70.6	31.0	70.6	79.7	29.0	79.7
R-Tuning	86.5	21.4	$77.9_{(90.0)}$	52.9	28.9	$44.8_{(84.6)}$	73.8	23.1	$69.0_{(93.5)}$	79.6	18.4	$77.8_{(97.7)}$
RL-based												
S^2R	86.8	33.0	86.8	53.4	39.0	**53.4**	75.6	30.0	**75.6**	81.0	28.0	81.0
BARREL	86.1	22.0	$77.9_{(90.5)}$	54.7	28.0	$42.8_{(78.2)}$	75.1	21.0	$65.6_{(87.4)}$	85.8	20.0	$78.2_{(91.1)}$
TruthRL	77.8	22.1	$77.3_{(99.3)}$	53.2	50.6	$47.3_{(89.0)}$	73.7	19.2	$68.5_{(92.9)}$	82.5	17.2	$81.3_{(98.6)}$
Ours												
Reflection-only	83.5	21.2	83.5	49.2	26.0	49.2	72.9	17.7	72.9	86.2	15.9	**86.2**
REIN	**89.2**	**12.0**	$83.0_{(93.0)}$	**59.5**	**18.0**	$48.5_{(81.5)}$	**78.2**	**11.0**	$72.3_{(92.5)}$	**88.0**	**10.0**	$83.6_{(95.0)}$

accuracy by 6.9 points, with both higher coverage and lower H-Proxy. Relative to TruthRL, REIN reduces H-Proxy by 13.9 points and improves effective accuracy by 3.8 points, at the cost of a 2.4-point reduction in coverage. These comparisons describe each method at its native operating point; Section 5.1 separately compares methods at a common coverage.

Behavior across task difficulty. On MATH, REIN improves selective accuracy over Base by 19.8 percentage points for Qwen2.5-7B and 21.0 points for Llama-3.1-8B, while reducing coverage by 16.5 and 18.5 points, respectively. The coverage reductions on GSM8K and ARC-Challenge are smaller. This pattern is consistent with more selective answering on a harder benchmark, although benchmark difficulty is only a proxy for policy-relative answerability. For both backbones, effective accuracy on MATH also exceeds the Base result despite lower coverage.

5.1 Mechanism Analysis

Comparison at matched coverage. To control for differences in native answer rates, we threshold each method's own continuous confidence score to obtain 90% coverage on GSM8K with Qwen2.5-7B. Table 2 reports forced-answer accuracy with abstention disabled (Raw), selective accuracy at 90% coverage, and the gain over Raw. This thresholding is an evaluation-only analysis, separate from REIN's native answer-or-abstain generation.

This comparison shows higher retained-answer accuracy, but does not by itself isolate better abstention ranking from differences in underlying answer quality.

Confidence scores on fixed answers. We compare seven selective scores on the same fixed REIN answers. The reflection-stance score achieves an area under the risk–coverage curve (AURC) of 0.038 on GSM8K, versus 0.040 for sixteen-sample self-consistency agreement; on MATH, the

Table 2: Matched-coverage evaluation on GSM8K with Qwen2.5-7B. Raw is accuracy with abstention disabled. Sel.Acc.@90 is selective accuracy at 90% coverage, obtained by thresholding each method's own continuous confidence score. Δ is Sel.Acc.@90 minus Raw, in percentage points (pp).

Method	Raw (%)	Sel.Acc.@90 (%)	Δ (pp)
R-Tuning	79.4	82.0	+2.6
BARREL	85.0	87.7	+2.7
TruthRL	84.0	88.0	+4.0
REIN	88.1	**92.2**	+4.1

corresponding values are 0.257 and 0.255. Combining the scores reduces AURC to 0.031 and 0.246, respectively. Reflection is therefore competitive rather than uniformly superior to sample agreement. Full selective-prediction results are reported in the appendix.

Two-level finalization. Reflection assesses the current draft y_0, while the IDK-free boundary-probe group supplies a prompt-level empirical answerability signal. A no-success probe group is a finite-budget supervision signal, not proof of intrinsic knowledge absence. Across 3,220 Qwen2.5-7B completions, finalization repairs 82 of 747 incorrect drafts (11.0%) and corrupts 22 of 2,454 correct drafts (0.9%). The complete transition counts are reported in Appendix G; non-substantive outputs include both explicit abstentions and invalid outputs.

Reflection interventions. Holding initial drafts fixed, we intervene before final-answer decoding (Table 3). Removing reflection reduces repair to 0.3%. Transplanting a reflection from another question while preserving its stance reduces repair to 4.1%, whereas a meaning- and stance-preserving paraphrase retains 10.4%. Forcing a `wrong` stance increases repair but also corruption and reduces effective accuracy. These interventions support a role for question-specific reflection content in finalization, without establishing step-level error localization.

Table 3: Reflection interventions on Qwen2.5-7B, using the same initial drafts ($N = 3220$). Metrics are micro-aggregated percentages. Repair and Corrupt are conditional on initially incorrect and correct drafts, respectively; corruption means an incorrect substantive final answer.

ID	Intervention	Sel.Acc.↑	Cov.↑	Eff.Acc.↑	Repair↑	Corrupt↓
A0	Intact	83.9	91.9	77.0	11.0	0.9
A1	Reflection excised	77.8	97.8	76.1	0.3	0.1
A2	Force `correct`	77.1	98.6	76.0	0.9	0.4
A3	Force `wrong`	88.2	71.5	63.0	14.2	6.1
A4	Stance flipped	74.8	90.2	67.5	5.8	5.4
A5	Reflection transplanted	79.5	92.1	73.2	4.1	3.6
A6	Reflection paraphrased	83.5	92.0	76.8	10.4	1.0

6 Conclusion

We presented REIN, a framework that improves the reliability of large reasoning models by enabling them to assess whether a generated solution should be trusted or rejected before finalizing an answer. REIN addresses two complementary failure modes of hallucination. Reflection alignment helps the model identify unreliable reasoning outcomes, while boundary-aware abstention encourages the model to avoid unsupported answers when the current capability is insufficient. Together, these signals provide a unified reliability-oriented training approach without requiring step-level supervision or inference-time intervention. Experiments across multiple backbones and reasoning benchmarks show that REIN reduces confident but incorrect responses while maintaining strong accuracy and coverage. These results highlight that reliable reasoning requires not only generating better solutions, but also knowing when a solution should not be trusted. Future work will investigate more fine-grained reliability assessment and broader uncertainty modeling.

AI USE STATEMENT

We used AI assistants solely for manuscript polishing, including grammar correction, wording refinement, and readability improvement. All research ideas, technical methods, experimental design, result analysis, and conclusions were independently developed and validated by the authors.

ETHICS STATEMENT

The goal of this research is to mitigate inaccurate yet confident argumentation through better self-verification and correction. More reliable text is a way to decrease the negative impact of false or misleading facts. However, increased user trust in model outputs due to increased reliability also makes it necessary to provide calibrated uncertainty and transparent communication about model outputs. To accomplish this, we use automated reward metrics, and a reflection style of training data; if the training data has any form of bias or unsafe content, the final model will inherit them since they are likely to be aligned. We recommend the following best practices for data hygiene, safety filtering, and red teaming prior to deploying the aligned model. In addition, we recommend the inclusion of both accuracy and hallucination-aware metrics when reporting results, as there is a risk of prioritizing perceived reliability over accurate results.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed REIN framework, including the reflection alignment objective, boundary-aware abstention mechanism, and optimization procedure in Section 3. The experimental settings, datasets, evaluation protocols, model configurations, and training hyperparameters are described in Section 4, with additional implementation details and analyses provided in the Appendix. We will release the source code, training configurations, and evaluation scripts in the supplementary materials to facilitate independent reproduction of our results.

REFERENCES

Zhenyu Chen et al. R1-Code-Interpreter: LLMs reason with code via supervised and multi-stage reinforcement learning. *arXiv preprint arXiv:2505.21668*, 2025.

Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert van Rooij, Kun Zhang, and Zhouchen Lin. Empowering llms with logical reasoning: A comprehensive survey, 2025. URL https://arxiv.org/abs/2502.15652.

Qinyuan Cheng, Tianxiang Sun, Xiangyang Liu, Wenwei Zhang, Zhangyue Yin, Shimin Li, Linyang Li, Zhengfu He, Kai Chen, and Xipeng Qiu. Can AI assistants know what they don't know? In *Proceedings of the 41st International Conference on Machine Learning*, pp. 8184–8202. PMLR, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Roi Cohen, Eden Biran, Ori Yoran, Mor Geva, and Amir Globerson. I don't know: Explicit modeling of uncertainty with an IDK token. *Advances in Neural Information Processing Systems*, 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/14c018d2e72c521605b0567029ef0efb-Abstract-Conference.html.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle Use a Laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021b.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023. URL https://arxiv.org/abs/2310.01798.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024. URL https://aclanthology.org/2024.tacl-1.78/.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang Ma, Tom Le Paine, Julian Schrittwieser, Yao Zhou, Clement Farabet, Bobak Shahriari, John Wieting, Quan Vuong, Pankaj Kumar, Noah Fiedel, Noam Shazeer, Kevin Swersky, and Sergey Levine. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CjwERcAU7w.

Kenneth Li, Oam Patel, Fernanda Vi
'egas, Hanspeter Pfister, and Martin Wattenberg. When hindsight is not 20/20: Testing limits on reflective thinking in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3151–3162, 2024. URL https://aclanthology.org/2024.findings-naacl.237/.

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

Shiyu Liu, Yongjing Yin, Jianhao Yan, Yunbo Tang, Qinggang Zhang, Bei Li, Xin Chen, Jingang Wang, Xunliang Cai, and Jinsong Su. Bapo: Boundary-aware policy optimization for reliable agentic search, 2026. URL https://arxiv.org/abs/2601.11037.

Ruotian Ma et al. S^2R: Teaching LLMs to self-verify and self-correct via reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2303.17651.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

Mistral AI. Mistral-7B-v0.3, 2024. URL https://huggingface.co/mistralai/Mistral-7B-v0.3. Model card.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290.

Baochang Ren, Shuofei Qiao, Da Zheng, Huajun Chen, and Ningyu Zhang. Knowrl: Exploring knowledgeable reinforcement learning for factuality, 2026. URL https://arxiv.org/abs/2506.19807.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2303.11366.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Zhepei Wei, Xiao Yang, Kai Sun, Jiaqi Wang, Rulin Shao, Jingxiang Chen, Mohammad Kachuee, Teja Gollapudi, Yiwei Liao, Nicolas Scheffer, Rakesh Wanga, Anuj Kumar, Yu Meng, Wen tau Yih, and Xin Luna Dong. Truthrl: Incentivizing truthful llms via reinforcement learning, 2026. URL https://arxiv.org/abs/2509.25760.

Junxiao Yang, Jinzhe Tu, Haoran Liu, Xiaoce Wang, Chujie Zheng, Zhexin Zhang, Shiyao Cui, Caishun Chen, Tiantian He, Hongning Wang, Yew-Soon Ong, and Minlie Huang. Barrel: Boundary-aware reasoning for factual and reliable lrms, 2026a. URL https://arxiv.org/abs/2505.13529.

Junxiao Yang et al. BARREL: Boundary-aware reasoning for factual and reliable LRMs. In *International Conference on Learning Representations*, 2026b.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023a. URL https://arxiv.org/abs/2308.01825.

Zheng Yuan et al. Scaling relationship on learning mathematical reasoning with large language models, 2023b.

Hanning Zhang, Shizhe Chen, Yaowei Wang, Chao Chen, Li Wang, Zhiyuan Liu, and Maosong Sun. R-Tuning: Instructing large language models to say "i don't know". In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 7113–7130, 2024. URL https://aclanthology.org/2024.naacl-long.394/.

Shenghe Zheng, Qianjia Cheng, Junchi Yao, Mengsong Wu, Haonan He, Ning Ding, Yu Cheng, Shuyue Hu, Lei Bai, Dongzhan Zhou, Ganqu Cui, and Peng Ye. Scaling physical reasoning with the PHYSICS dataset. *arXiv preprint arXiv:2506.00022*, 2025.

A METRIC DEFINITIONS

A.1 OUTCOME PARSING AND NOTATION

For an evaluation set containing $N \geq 1$ instances, each method first generates one response using its native inference procedure. A deterministic output adapter extracts the final-answer span and maps it to exactly one of three mutually exclusive outcomes: a substantive answer, an explicit `IDK`, or an invalid output. We define

$$\begin{aligned} A_i &= \mathbb{1}[y_i \text{ is a substantive answer}], \\ D_i &= \mathbb{1}[y_i \text{ is an explicit } \texttt{IDK}], \\ U_i &= \mathbb{1}[y_i \text{ is invalid}], \end{aligned} \qquad A_i + D_i + U_i = 1. \tag{11}$$

The abstention parser examines only the extracted final-answer span; uncertainty expressed in the reasoning or reflection does not constitute an abstention. An empty or missing answer, malformed required delimiters, multiple conflicting answers, a mixture of an abstention and a substantive answer, or an answer that cannot be parsed for the target task is considered invalid. Method-specific output adapters may be used to accommodate different native output formats, but they are fixed before evaluation and cannot access reference answers or verifier feedback.

Let

$$T_i = \mathbb{1}\left[A_i = 1 \ \wedge \ y_i \text{ passes the task-specific verifier}\right]. \tag{12}$$

Thus, $T_i \leq A_i$, and $T_i = 0$ whenever $A_i = 0$. GSM8K uses normalized numerical equivalence, MATH-500 uses mathematical-equivalence checking, StrategyQA uses normalized exact matching over `yes`/`no`, and ARC-Challenge uses normalized exact matching of the selected option label.

We define

$$N_a = \sum_{i=1}^{N} A_i, \qquad N_c = \sum_{i=1}^{N} T_i, \qquad N_w = \sum_{i=1}^{N} (A_i - T_i), \tag{13}$$

where N_a, N_c, and N_w denote the numbers of substantive, correct substantive, and incorrect substantive answers, respectively. Consequently, $N_a = N_c + N_w$.

A.2 NATIVE REFLECTION INDICATOR

For H-Proxy evaluation, every method generates a structured`<think>` $\rightarrow$ `<reflection>` $\rightarrow$ `<answer>` completion in a single autoregressive pass. We use *native* to mean that the reflection is emitted before the final answer within this same completion, regardless of whether the method was explicitly trained with a reflection objective. No additional post-hoc judging call is used to compute the primary H-Proxy. Each native completion is parsed into a pre-reflection draft $y_i^{(0)}$, a reflection r_i, and a final answer $y_i^{(1)}$. The draft $y_i^{(0)}$ is extracted only from the content preceding the reflection.

Let

$$P_i^{(0)} = \mathbb{1}\left[y_i^{(0)} \text{ is a parseable substantive draft}\right] \tag{14}$$

denote whether a valid draft answer can be extracted. Draft extraction failures are excluded from draft-correctness metrics and are reported separately through the draft extraction rate.

Draft correctness is defined as

$$T_i^{(0)} = \mathbb{1}\left[P_i^{(0)} = 1 \ \wedge \ y_i^{(0)} \text{ passes the task-specific verifier}\right]. \tag{15}$$

The native reflection stance is parsed as

$$s_i \in \{\texttt{correct}, \texttt{wrong}, \texttt{uncertain}, \perp\}, \tag{16}$$

where $\perp$ denotes an unparsable reflection. We define the binary native reliability indicator as

$$J_i^{\text{nat}} = \mathbb{1}\left[s_i = \texttt{correct}\right]. \tag{17}$$

Thus, `wrong`, `uncertain`, and unparsable reflections are not treated as endorsements. Uncertain and unparsable cases remain separately visible in the native-reflection diagnostics.

Let

$$N_w^{(0)} = \sum_{i=1}^{N} P_i^{(0)}(1 - T_i^{(0)}) \tag{18}$$

denote the number of incorrect parseable drafts, and let

$$N_e^{\mathrm{nat}} = \sum_{i=1}^{N} P_i^{(0)}(1 - T_i^{(0)}) J_i^{\mathrm{nat}} \tag{19}$$

denote the number of incorrect drafts that are endorsed as `correct` by their native reflections.

A.3 Primary Evaluation Metrics

Since REIN allows the model to abstain, standard accuracy alone is not sufficient for evaluation. A model can obtain higher accuracy on attempted questions simply by refusing many difficult questions. Therefore, we evaluate reliability under a selective answering setting using four metrics: Selective Accuracy, Hallucination Proxy (H-Proxy), Coverage, and Effective Accuracy.

$$\begin{aligned}
\mathrm{Sel.Acc.} &= \frac{N_c}{N_a}, \\
\text{H-Proxy} &= \frac{N_e^{\mathrm{nat}}}{N_w^{(0)}}, \\
\mathrm{Cov.} &= \frac{N_a}{N}, \\
\mathrm{Eff.Acc.} &= \mathrm{Sel.Acc.} \times \mathrm{Cov.} = \frac{N_c}{N}.
\end{aligned}$$

Selective Accuracy measures the correctness of answers that the model actually attempts. It reflects whether the model gives more accurate answers when it chooses to answer.

Hallucination Proxy measures the fraction of incorrect pre-reflection drafts that are nevertheless endorsed as `correct` by the native reflection generated within the same completion. It captures the failure mode in which the model incorrectly endorses an unreliable draft before producing its final answer.

Coverage measures the fraction of questions for which the model provides a substantive answer instead of abstaining or producing an invalid output.

Effective Accuracy measures the overall fraction of questions answered correctly after accounting for abstention. It combines answer correctness and coverage, and therefore reflects the actual utility of the model under selective answering.

Higher Selective Accuracy, Coverage, and Effective Accuracy are better, while lower Hallucination Proxy is better. Together, these metrics show whether REIN improves reliability without relying on excessive abstention.

A.4 Edge Cases and Reporting

If $N_a = 0$, Coverage and Effective Accuracy are defined as zero, whereas Selective Accuracy is undefined because its denominator is zero. Because H-Proxy is computed from pre-reflection drafts, its definition does not depend on N_a.

If $N_w^{(0)} = 0$, H-Proxy is undefined and is reported as "–". Other conditional reliability metrics are likewise reported as "–" whenever their corresponding denominators are zero.

All metrics are computed directly from sample-level events and are reported as percentages. Algebraic identities are used only for interpretation; final values are not reconstructed by multiplying rounded table entries. Metrics are computed separately for each checkpoint–benchmark pair, and every cross-benchmark aggregate explicitly states whether macro- or micro-averaging is used.

Table 4: Evaluator statistics reported as mean and standard deviation. $\mathrm{ACC}_{\mathrm{step}}$ measures the accuracy of locating the first error in a response, while $\mathrm{ACC}_{\mathrm{reason}}$ measures the accuracy of identifying the error type. The evaluation is based on GSM8K and MATH examples. A total of 200 incorrect responses for each method were manually annotated.

Method	$\mathrm{ACC}_{\mathrm{step}} \uparrow$	$\mathrm{ACC}_{\mathrm{reason}} \uparrow$
Cold-start SFT	$46.2_{\pm 2.1}$	$38.8_{\pm 2.4}$
REIN w/o IDK	$57.3_{\pm 1.8}$	$48.5_{\pm 2.1}$
REIN (Ours)	$\mathbf{62.0}_{\pm 1.5}$	$\mathbf{53.5}_{\pm 1.8}$

B Native-Reflection Evaluation Protocol

All reliability judgments are obtained directly from the `<reflection>` field generated within the same native completion as the draft and final answer. No additional post-hoc reliability-judgment call is used.

Reflection parsing. The deterministic parser maps each native reflection to `correct`, `wrong`, `uncertain`, or $\perp$, according to the rules defined in the Native Reflection Indicator section. The parser examines only the reflection field and cannot access reference answers, verifier feedback, or training-derived boundary labels.

Reporting. We report H-Proxy together with the draft extraction rate, reflection parse rate, uncertain rate, false-rejection rate, balanced accuracy, and Matthews correlation coefficient. We also report the complete reflection-stance confusion matrix so that uncertain and unparsable outputs remain visible.

All native-reflection metrics are computed from the same completion used to produce the final answer. Consequently, these metrics evaluate deployment-faithful self-assessment rather than the checkpoint's ability to verify a frozen answer through an additional evaluation call.

B.1 Error-Localization Diagnostics

The repository has two metrics to evaluate reflection quality, which are calculated from incorrect answers that were manually annoted. **$\mathrm{ACC}_{\mathrm{step}}$** Tests the accuracy of identifying the first step in reasoning, representing how well the model localizes the source of that failure. **$\mathrm{ACC}_{\mathrm{reason}}$** Tests the accuracy of identifying the underlying reason for that failure. This means that the model has been able to learn about the reasoning failure beyond simply identifying it. The diagnostic does require an annotation file that includes the ground truth for both step and reason for each failure to generate a result will be reflected in these metrics.

B.2 Auxiliary Post-hoc Reliability Probe

The primary H-Proxy reported in this paper is computed from the native pre-finalization reflection generated within the same completion as the draft and final answer. We additionally introduce a standardized post-hoc reliability probe as an auxiliary validation of this native judgment. The probe is used only for the analysis in Table 5; it is not used as a training signal, does not modify the frozen completion, and does not replace the native H-Proxy reported in the primary results.

Probe input. For each parseable substantive pre-reflection draft $y_i^{(0)}$, the evaluated checkpoint is shown only the original question x_i and its own frozen draft $y_i^{(0)}$. The native reasoning trace, native reflection r_i, delivered final answer $y_i^{(1)}$, reference answer, verifier feedback, and training-derived boundary labels are withheld. Thus, the native and post-hoc judgments evaluate the same draft, while differing only in whether the judgment is generated within the original completion or through an additional evaluation call.

The fixed evaluation prompt is:

> You are assessing the reliability of a frozen draft answer. Do not revise, replace, or extend the draft. If you are uncertain whether it is correct, label it unreliable.
> Question: {question}
> Draft answer: {draft}
> Output exactly one label: reliable or unreliable.

Decoding and parsing. We use greedy decoding with at most eight new tokens. After lowercasing and removing surrounding whitespace, the deterministic parser accepts only the exact labels `reliable` and `unreliable`. Additional text, multiple labels, or the absence of a valid label are mapped to $\perp$. We write

$$s_i^{\text{post}} \in \{\texttt{reliable}, \texttt{unreliable}, \perp\}. \tag{20}$$

The corresponding binary post-hoc reliability indicator is

$$J_i^{\text{post}} = \mathbb{1}\left[s_i^{\text{post}} = \texttt{reliable}\right]. \tag{21}$$

Thus, only an exact `reliable` label constitutes an endorsement; `unreliable` and unparsable judgments do not.

Post-hoc H-Proxy and agreement. To isolate the effect of the judgment procedure, the native and post-hoc quantities are computed over exactly the same population of incorrect parseable drafts. Using $N_w^{(0)}$ from Appendix A, we define

$$N_e^{\text{post}} = \sum_{i=1}^{N} P_i^{(0)} \left(1 - T_i^{(0)}\right) J_i^{\text{post}}, \tag{22}$$

and

$$\text{H-Proxy}^{\text{post}} = \frac{N_e^{\text{post}}}{N_w^{(0)}}. \tag{23}$$

We compare the two judgment procedures using

$$\Delta = \text{H-Proxy}^{\text{post}} - \text{H-Proxy}, \tag{24}$$

and Cohen's κ between J_i^{nat} and J_i^{post}, restricted to

$$\left\{i : P_i^{(0)} = 1 \ \wedge \ T_i^{(0)} = 0\right\},$$

the incorrect-draft population that defines H-Proxy.

Table 5 shows that the two judgment procedures exhibit substantial but imperfect agreement. Across the REIN and Reflection-only benchmark cells, the post-hoc probe endorses 2.1–3.5 percentage points fewer incorrect drafts than the native reflection, with an average gap of approximately 2.6 points. Cohen's κ ranges from .694 to .778.

The difference is a stable level shift rather than a change in method ordering: REIN has a lower false-endorsement rate than Reflection-only on every benchmark under both judgment procedures. These results indicate that the lower native H-Proxy of REIN is not specific to the same-pass reflection parser. At the same time, the nonzero gaps show that native and post-hoc judgments are not interchangeable; the post-hoc probe is consistently more conservative.

C Supplementary Experimental Results

C.1 Comparison with Existing Methods

We compare REIN with existing methods for mitigating reasoning hallucinations in figure 3, including In-Context Learning (ICL), knowledge distillation, and BARREL. To ensure a fair comparison, we adopt three same key evaluation metrics with BARREL: Accuracy (Acc.), Truthfulness (Truth.), and Reliability (Rel.), which are defined as follows: Acc. $= \frac{N_c}{N}$, Truth. $= \frac{N_c+N_r}{N}$, and Rel. $=$ ans. $\cdot$ Truth. $+ (1 -$ ans.$) \cdot$ Acc., where ans. $= 1 - \frac{N_r}{N}$. As illustrated in Figure 3, REIN achieves the highest Acc., Truth., and Rel. among all compared baselines. Furthermore, we also compare our method against token-level calibration approaches such as R-Tuning and IDK-token training. We observe that these methods significantly interfere with the model's original output distribution, thereby degrading its inherent reasoning capabilities.

Table 5: Native and post-hoc reliability judgments on the same pre-reflection drafts for Qwen2.5-7B. Δ is defined as $\text{H-Proxy}^{\text{post}} - \text{H-Proxy}$ and is negative throughout, indicating that the post-hoc probe is the stricter judge. Cohen's κ measures agreement between the two binary judgments on the incorrect parseable drafts that define H-Proxy. For Self-Refine, a best-effort reliability stance is recovered from its last critique; because this critique is not a native same-pass reflection attached to the evaluated draft, the row is included only as an auxiliary validation point.

Method / benchmark	**H-Proxy**	**H-Proxy^{post}**	**Δ**	**κ**
REIN				
GSM8K	11.0	8.6	-2.4	.751
MATH	15.0	11.8	-3.2	.712
StrategyQA	10.0	7.7	-2.3	.766
ARC-Challenge	9.0	6.9	-2.1	.778
macro	**11.3**	**8.8**	$\mathbf{-2.5}$	**.752**
Reflection-only				
GSM8K	18.9	16.2	-2.7	.728
MATH	22.1	18.6	-3.5	.694
StrategyQA	17.5	15.1	-2.4	.741
ARC-Challenge	16.2	14.0	-2.2	.753
macro	**18.7**	**16.0**	$\mathbf{-2.7}$	**.729**
Self-Refine (last critique verdict; not scored natively)				
macro	41.4	38.5	-2.9	.688

Table 6: Training cost on Qwen2.5-7B, using two RTX PRO 6000 Blackwell GPUs, LoRA $r = 16$, 240 GRPO steps at 32 prompts per step. Roll./pr. is rollouts per prompt per step; * R-Tuning is teacher-forced and generates nothing during training; ‡ REIN's second group is the IDK-free boundary probe and is drawn only in Phase 2, so its marginal cost is three quarters of a second full group rather than a full one. All methods share the SFT warm start, whose cost is listed once and excluded from the ratio column.

Method	Roll./pr.	Gen. tok.	GPU-h	Wall (h)	×GRPO
SFT warm start (shared)	–	–	3.1	1.6	–
R-Tuning	1*	0.0 M	5.8	2.9	0.27×
DPO	4	9.6 M	8.4	4.2	0.39×
RFT	8	19.4 M	13.1	6.6	0.61×
Vanilla GRPO	16	38.1 M	21.4	10.7	1.00×
TruthRL	16	38.4 M	21.6	10.8	1.01×
BARREL	16	39.6 M	22.3	11.2	1.04×
Reflection-only (ours)	16	40.2 M	22.8	11.4	1.07×
S^2R	$16 + 8^{\ddagger}$	55.1 M	30.4	15.2	1.42×
REIN (ours)	$16 + 16^{\ddagger}$	**67.2 M**	**37.9**	**18.9**	**1.77×**

C.2 Extended Baseline Comparison under a Unified Protocol

Table 1 places REIN alongside nine published methods spanning training-free self-correction, refusal-aware fine-tuning, and RL-based reliability alignment, together with the Base checkpoint and the IDK-prompting control of Table 1 of the main paper. The results separate two quantities that the literature usually reports together. Reasoning-oriented optimization reliably buys accuracy: vanilla GRPO lifts average effective accuracy from 64.9 to 70.7 on Qwen2.5-7B, and S^2R, which explicitly rewards self-verification and self-correction, reaches the highest effective accuracy of any method in the table at 74.1. Neither buys reliability: their H-Proxy stays at 34.8 and 30.8, barely below the 40.0 of the untrained backbone, so roughly one wrong answer in three is still delivered with an explicit assertion that it is sound. Test-time sampling is not an alternative route to it either—sixteen-sample self-consistency adds 6.8 points of effective accuracy and simultaneously raises H-Proxy to 42.0, because majority voting removes exactly the errors the model was least sure of and leaves behind the self-consistent ones that reflection is most willing to endorse. Abstention supervision

Table 7: Full results

Method	Mathematical Reasoning						Commonsense Reasoning					
	GSM8K			MATH			StrategyQA			ARC-Challenge		
	Sel.Acc.↑	H-Proxy↓	Eff.Acc.↑	Sel.Acc.↑	H-Proxy↓	Eff.Acc.↑	Sel.Acc.↑	H-Proxy↓	Eff.Acc.↑	Sel.Acc.↑	H-Proxy↓	Eff.Acc.↑
Backbone: Qwen2.5-7B												
Training-free												
Base	79.1	41.0	79.1	41.2	45.0	41.2	64.3	38.0	64.3	74.8	36.0	74.8
Self-Refine	78.4	39.0	78.4	40.6	44.0	40.6	65.1	36.0	65.1	74.9	35.0	74.9
Self-Consist@16	87.3	43.0	87.3	50.4	47.0	50.4	69.8	40.0	69.8	79.1	38.0	79.1
IDK Prompting	86.8	22.5	$78.9_{(90.9)}$	54.1	31.1	$38.7_{(71.5)}$	73.4	22.0	$60.2_{(82.0)}$	84.2	20.8	$76.7_{(91.1)}$
Fine-tuning												
RFT	84.7	34.0	84.7	47.9	39.0	47.9	68.9	32.0	68.9	78.6	30.0	78.6
DPO	82.9	36.0	82.9	46.1	41.0	46.1	67.5	33.0	67.5	77.4	31.0	77.4
R-Tuning	84.2	13.5	$68.8_{(81.7)}$	51.2	27.5	$40.7_{(79.4)}$	64.3	12.4	$57.5_{(89.4)}$	86.1	17.2	$82.0_{(95.2)}$
RL-based												
Vanilla GRPO	86.4	35.0	86.4	49.7	40.0	49.7	68.4	33.0	68.4	78.2	31.0	78.2
S^2R	88.5	31.0	88.5	53.8	37.0	**53.8**	74.2	28.0	**74.2**	79.8	27.0	79.8
BARREL	87.4	21.0	$79.7_{(91.2)}$	56.2	27.0	$44.7_{(79.6)}$	73.5	20.0	$64.8_{(88.1)}$	84.1	19.0	$76.0_{(90.4)}$
TruthRL	86.9	17.9	$80.6_{(92.7)}$	57.3	42.7	$48.7_{(85.0)}$	67.7	26.9	$62.7_{(92.6)}$	83.7	10.8	$77.5_{(92.6)}$
Ours												
Reflection-only	85.2	18.9	**85.2**	51.2	22.1	51.2	70.3	17.5	**70.3**	82.1	16.2	**82.1**
REIN	**90.6**	**11.0**	$85.0_{(93.8)}$	**61.0**	**15.0**	$50.9_{(83.5)}$	**76.2**	**10.0**	$69.2_{(90.8)}$	**86.5**	**9.0**	$80.9_{(93.5)}$
Backbone: LLaMA-3.1-8B												
Training-free												
Base	76.8	44.0	76.8	38.5	48.0	38.5	66.7	37.0	66.7	76.2	35.0	76.2
Self-Refine	76.1	42.0	76.1	37.8	47.0	37.8	67.4	35.0	67.4	76.8	34.0	76.8
Self-Consist@16	85.6	46.0	85.6	47.9	50.0	47.9	71.5	39.0	71.5	80.3	37.0	80.3
IDK Prompting	85.0	25.2	$74.2_{(87.3)}$	51.8	33.9	$36.8_{(71.0)}$	74.8	21.1	$67.4_{(90.1)}$	86.2	20.5	$76.2_{(88.4)}$
Fine-tuning												
RFT	83.2	36.0	83.2	45.1	42.0	45.1	70.6	31.0	70.6	79.7	29.0	79.7
DPO	81.4	38.0	81.4	43.6	44.0	43.6	69.2	33.0	69.2	78.5	31.0	78.5
R-Tuning	86.5	21.4	$77.9_{(90.0)}$	52.9	28.9	$44.8_{(84.6)}$	73.8	23.1	$69.0_{(93.5)}$	79.6	18.4	$77.8_{(97.7)}$
RL-based												
Vanilla GRPO	84.9	37.0	84.9	47.2	43.0	47.2	70.1	32.0	70.1	79.4	30.0	79.4
S^2R	86.8	33.0	86.8	53.4	39.0	**53.4**	75.6	30.0	**75.6**	81.0	28.0	81.0
BARREL	86.1	22.0	$77.9_{(90.5)}$	54.7	28.0	$42.8_{(78.2)}$	75.1	21.0	$65.6_{(87.4)}$	85.8	20.0	$78.2_{(91.1)}$
TruthRL	77.8	22.1	$77.3_{(99.3)}$	53.2	50.6	$47.3_{(89.0)}$	73.7	19.2	$68.5_{(92.9)}$	82.5	17.2	$81.3_{(98.6)}$
Ours												
Reflection-only	83.5	21.2	83.5	49.2	26.0	49.2	72.9	17.7	72.9	86.2	15.9	**86.2**
REIN	**89.2**	**12.0**	$83.0_{(93.0)}$	**59.5**	**18.0**	$48.5_{(81.5)}$	**78.2**	**11.0**	$72.3_{(92.5)}$	**88.0**	**10.0**	$83.6_{(95.0)}$
Backbone: Mistral-7B-v0.3												
Base	51.4	51.0	51.4	12.8	55.0	12.8	58.5	43.1	58.5	65.4	42.2	65.4
IDK Prompting	54.8	34.0	$45.2_{(82.5)}$	15.8	43.1	$9.2_{(58.4)}$	62.4	32.3	$49.3_{(79.0)}$	69.5	30.1	$59.0_{(84.9)}$
Reflection-only	54.5	32.1	**54.5**	15.2	36.1	**15.2**	62.0	28.7	**62.0**	69.0	27.1	**69.0**
REIN (Ours)	**58.5**	**17.9**	$52.7_{(90.1)}$	**18.5**	**25.1**	$13.4_{(72.4)}$	**65.7**	**19.7**	$59.4_{(90.4)}$	**72.2**	**18.1**	$66.0_{(91.4)}$
Backbone: DeepSeek-R1-Distill-8B												
Base	86.5	28.0	86.5	72.8	22.0	72.8	69.8	30.0	69.8	79.6	31.0	79.6
IDK Prompting	89.0	19.0	$81.4_{(91.5)}$	76.5	17.0	$62.7_{(82.0)}$	73.8	23.0	$61.6_{(83.5)}$	83.2	24.0	$71.1_{(85.5)}$
Reflection-only	87.8	15.0	**87.8**	74.3	13.0	**74.3**	70.8	20.0	**70.8**	81.2	18.0	81.2
REIN (Ours)	**91.8**	**8.0**	$87.7_{(95.5)}$	**80.5**	**9.0**	$74.1_{(92.0)}$	**76.8**	**14.0**	$70.7_{(92.0)}$	**86.0**	**10.0**	**81.3**$_{(94.5)}$

moves the second quantity but at a cost on the first: R-Tuning reaches the lowest H-Proxy of any external method (17.7) yet abstains hard enough that its effective accuracy of 62.3 falls below the untrained baseline, and BARREL, whose IDK recipe was calibrated on long-CoT models that guess at the last moment, transfers only partially to short-CoT instruction-tuned backbones and settles at a conservative operating point of 87.3 coverage and 66.3 effective accuracy.

The most informative comparison is with TruthRL, the strongest RL-based abstention baseline, and here the two quantities come apart cleanly. On Qwen2.5-7B the two methods sit at almost the same coverage (90.7 against our 90.4), and at that shared operating point REIN adds 4.7 points of selective accuracy and 4.1 points of effective accuracy while cutting H-Proxy from 24.6 to 11.3. On LLaMA-3.1-8B TruthRL retains 4.5 more coverage points (95.0 against 90.5) and still converts them into 3.3 fewer correct answers over the full evaluation set (68.6 against 71.9). The advantage is not carried by one benchmark: REIN attains higher effective accuracy in all eight backbone–benchmark cells, and the largest single margin, 6.5 points, appears on StrategyQA with Qwen2.5-7B, where TruthRL answers 92.6% of the questions and endorses 26.9% of the wrong answers it delivers.

That contrast also localizes the mechanism. TruthRL's ternary reward leaves H-Proxy above our reflection-only ablation on both backbones (24.6 against 18.7 on Qwen2.5-7B, 27.3 against 20.2

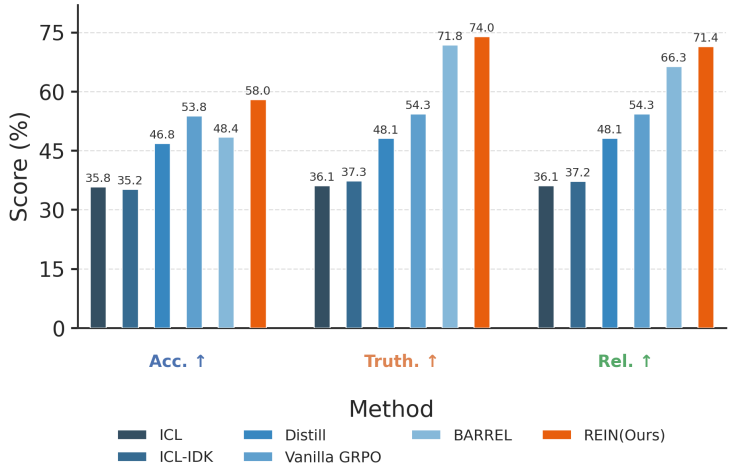

Figure 3: Comparison of REIN with existing methods. REIN achieves the highest Accuracy, Truthfulness, and Reliability. The model is Deepseek-Distill-8B and the benchmark is TriviaQA.

on LLaMA-3.1-8B), which is what one expects of any signal that prices whether to answer without constraining how the model assesses the answer it does give; only the full objective, in which the reflection-veracity term aligns the stance of an explicit reflection span with verified correctness while the boundary-aware term redirects the answer distribution itself, brings the rate below 20.0 on every benchmark and every backbone, and to 12.0 or below outside MATH. What REIN does not claim is uniform superiority on utility alone: S^2R answers every question and still reaches 74.1 average effective accuracy against our 71.5, so a deployment that never pays for a wrong answer is better served by refusing nothing. The case for REIN is that at a comparable answer rate it is right more often and, when it is wrong, says so.

C.3 Supplementary MCC Results

Supplementary reflection-aware metric We also present the Matthews Correlation Coefficient (MCC) as a secondary measure of reflection-aware reliability because it measures how much agreement there is between whether a model's output is correct or the output is reliably classified according to its reflection judgement (reliable vs.

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

unreliable). Here solution correctness will be used as the ground-truth label and the reflection judgment will be used as a binary prediction. In terms of definitions: TP, TN, FP, and FN are defined as true positives, true negatives, false positives, and false negatives respectively. Therefore, the MCC can be calculated as In our scenario, a positive label means that a solution is correct and the associated reflection judgement labels the solution as reliable, whereas a negative label indicates that the solution is incorrect and the associated reflection judgement labels the solution as unreliable. The MCC does not depend on whether the two classes of examples are equally distributed (i.e., has class imbalance) and therefore will give penalties to both false-positive (incorrect answers that are classified by the reflection judgement as being reliable) and false-negative (correct answers that are classified by the reflection judgement as being unreliable) classifications. Where indicated, we map correct to reliable and incorrect/uncertain to unreliable.

C.4 Evolution of Reward Components

Figure 5 illustrates the evolution of the overall reward and its five components during REIN training. The training exhibits two distinct phases. In the first phase, the model rapidly acquires the required response format and reasoning behavior, as reflected by the fast convergence of the XML structure

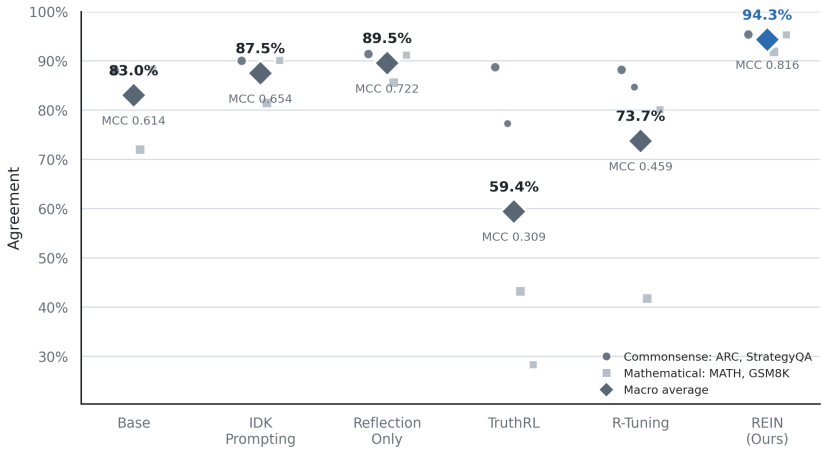

Figure 4: Reliability judgment calibration across methods and backbones: reflection–answer agreement and Matthews correlation coefficient (MCC).

and reflection presence rewards. Meanwhile, the accuracy reward steadily improves, indicating that the enforced reflection mechanism does not compromise task-solving ability.

After the phase transition, the reflection veracity and IDK alignment rewards continue to increase, suggesting that the model gradually learns to calibrate its self-evaluation and align its abstention behavior with its actual capability boundary. The consistent improvement of total reward demonstrates that REIN jointly optimizes answer correctness, reliable self-reflection, and selective abstention rather than optimizing these objectives independently.

Selective score construction. For REIN, we use the normalized probability assigned to the first stance token in the `<reflection>` span as a continuous selective score. Higher scores correspond to greater confidence in the current draft. For comparisons among selective scores, all scores rank the same fixed set of REIN answers, so the underlying candidate answers are held constant. Risk–coverage curves are obtained by thresholding each score over its empirical coverage grid, and AURC is the mean selective risk over that grid.

D Ablation Study

Additional analyses on group size, reward components, and IDK behavior are reported in Appendix E. Table 10 isolates the contribution of each reward component.

D.1 Effect of Reflection Alignment.

To investigate how the amount of reflection affects reliability, we vary the reflection depth $d \in \{1, 2, 3\}$ during training, where $d = 1$ uses a single `<reflection>` block and $d > 1$ uses nested reflection-correction cycles. Table 11 reports results on GSM8K.

A single reflection layer ($d = 1$) yields the best trade-off between hallucination reduction and reasoning fidelity. Deeper reflection ($d = 2, 3$) further suppresses hallucinations ($0.11 \rightarrow 0.08 \rightarrow 0.06$) but reduces coverage ($93.8\% \rightarrow 90.3\% \rightarrow 88.4\%$) and effective accuracy ($85.0 \rightarrow 80.4 \rightarrow 77.9$), suggesting an *overcorrection* phenomenon where the model becomes overly cautious in self-judgment. This result supports our design choice that reflection alignment should remain lightweight.

We compare Base with Reflection-only to examine the overall effect of reflection-structured training. Reflection-only uses the same structured output format and reflection rewards as REIN, but removes the boundary-aware IDK reward. Averaged over the four benchmarks, Reflection-only improves selective accuracy over Base for all two backbones and substantially reduces the hallucination proxy. However, Reflection-only keeps 100% coverage, showing that reflection-structured training improves reliability assessment but does not teach the model when to abstain.

Table 8: Selective prediction with different abstention scores. The native reflection stance provides a competitive single-pass selective signal, outperforming conventional likelihood-based confidence measures and matching or exceeding sixteen-sample self-consistency on GSM8K. Its combination with self-consistency yields the strongest selective performance, indicating that reflection-based diagnosis and sample agreement provide complementary reliability information.

(a) One model, one fixed answer set, seven abstention scores

Selective score	AUROC↑	AURC↓	Sel.@90↑	Sel.@80↑	#fwd
GSM8K (raw=88.1, risk@100=11.9)					
Sequence log-prob	.712	.063	89.6	91.2	1
Length-norm. log-prob	.741	.058	90.1	92.0	1
P(True)	.779	.051	90.8	93.1	2
Semantic entropy @16	.836	.043	91.6	94.5	16
Self-consistency agr. @16	.849	.040	91.9	94.9	16
Reflection stance (ours)	.857	.038	92.2	95.3	1
+ self-consistency @16	**.884**	**.031**	**93.1**	**96.4**	17
MATH (raw=55.0, risk@100=45.0)					
Sequence log-prob	.688	.288	56.4	58.2	1
Length-norm. log-prob	.703	.281	57.0	59.1	1
P(True)	.724	.272	57.6	60.2	2
Semantic entropy @16	.781	.259	58.4	61.8	16
Self-consistency agr. @16	.792	.255	58.7	62.4	16
Reflection stance (ours)	.786	.257	58.5	62.0	1
+ self-consistency @16	**.818**	**.246**	**59.3**	**63.5**	17

Table 9: Selective prediction comparison under a fixed coverage of 90%.

(b) Each method calibrated to Cov.=90 (GSM8K)

Method	Raw	Sel.@90↑	Δ over raw↑	H-Proxy@90↓
R-Tuning	79.4	82.0	+2.6	16.5
BARREL	85.0	87.7	+2.7	20.0
TruthRL	84.0	88.0	+4.0	17.0
REIN	88.1	**92.2**	**+4.1**	**10.0**

Table 10: Component ablation on GSM8K with Qwen2.5-7B (Refl.: reflection reward). All variants share the same SFT initialization, GRPO configuration, accuracy reward, and format rewards.

Configuration	Sel.Acc.↑	H-Proxy↓	Cov.↑	Eff.Acc.↑
Core GRPO	84.8	28.0	99.8	84.6
+ Refl.	85.2	19.0	**100.0**	**85.2**
+ IDK	90.2	20.0	93.5	84.3
+ Refl. + IDK	**90.6**	**11.0**	93.8	85.0

D.2 Selective boundary behavior.

The difficulty-stratified results in Appendix E show that abstention becomes more frequent as the tasks become harder. The IDK rate increases from 2.1% on two-step GSM8K problems to 14.8% on problems requiring at least five steps, and from 3.2% on MATH Level 1 to 31.8% on Level 5. ARC-Challenge shows a similar pattern across grade levels. Although benchmark difficulty is only an approximate indicator of policy-relative answerability, this monotonic trend argues against a uniform refusal strategy. The IDK precision of 78.4–83.2% and recall of 65.7–72.8% in Table 12 further indicate that refusals are concentrated on low-support prompts rather than applied indiscriminately.

Depth d	Sel.Acc↑	H-proxy↓	Cov.↑	Eff.Acc ↑
1	**90.6**	0.11	**93.8**	**85.0**
2	89.1	0.08	90.3	80.4
3	88.1	**0.06**	88.4	77.9

Table 11: Effect of reflection depth d on GSM8K (7B). Deeper reflection further suppresses hallucination but reduces accuracy and coverage due to overcorrection.

Table 12: Abstention quality for REIN on Qwen2.5-7B. The $K = 64$ reference is a higher-budget empirical boundary and is not an oracle label of intrinsic knowledge absence. Non-substantive Rate is the complement of coverage and includes both explicit abstentions and invalid outputs.

Dataset	IDK Prec.↑	IDK Recall↑	Non-substantive Rate
GSM8K	82.6	71.3	6.2%
MATH	78.4	65.7	16.5%
StrategyQA	80.1	68.5	9.2%
ARC-C	83.2	72.8	6.5%

D.3 Effect of Boundary-Aware Abstention.

Table 12 reports IDK precision and recall using the higher-budget $K = 64$ empirical boundary as a reference. A prompt is labelled `beyond` under this reference when none of the 64 IDK-free probes produces a verified-correct answer. As noted above, this reference is not an oracle label of intrinsic knowledge absence.

Both methods use the same reflection format, but REIN additionally uses the group-level boundary indicator and IDK reward. Across the evaluated backbones, REIN further improves selective accuracy and reduces the hallucination proxy compared with Reflection-only. This comes with a controlled reduction in coverage, because REIN learns to abstain on low-support prompts. Effective accuracy remains close to Reflection-only, indicating that the reliability gain is not obtained by excessive refusal.

E Knowledge-Boundary Definition Audit

The sampled knowledge boundary admits several plausible operational definitions, depending on which stage of the completion is inspected and whether abstention is available during sampling. We therefore compare three boundary estimators: a draft-based estimator, a forced-answer estimator, and an IDK-enabled final-answer estimator.

Our analysis has two parts. First, we apply the three estimators to the same prompts and checkpoint to compare their label distributions and pairwise agreement. Second, we train three otherwise identical variants that differ only in the boundary estimator and compare their selective behavior and downstream performance. This analysis measures the sensitivity of boundary-aware abstention to the operational definition of the sampled boundary.

E.1 Three Boundary Estimators

Fix a prompt x. All three estimators classify x as `within` when at least one sampled completion reaches the verified gold answer, but they differ in the sampled prompt and in whether the pre-reflection draft y_0 or delivered answer y_1 is inspected.

Draft-based boundary. We sample K completions under the IDK-enabled prompt and inspect the pre-reflection draft:

$$\widehat{B}_K^{\text{draft}}(x) = \begin{cases} \texttt{within}, & \exists k,\ T(x, y_0^{(k)}) = 1, \\ \texttt{beyond}, & \text{otherwise.} \end{cases} \tag{25}$$

The estimator returns `beyond` otherwise. Because the draft is extracted before the final answer-or-abstain decision, parseable drafts do not contain the final IDK action. This estimator can reuse the policy-gradient completion group and requires no additional rollouts.

Forced-answer boundary. We independently sample K_b completions under an IDK-free prompt, in which abstention is not offered as a final action, and inspect the delivered answer:

$$\widehat{B}^{\text{forced}}_{K_b}(x) = \begin{cases} \texttt{within}, & \exists k,\ T(x, y_1^{(k)}) = 1, \\ \texttt{beyond}, & \text{otherwise.} \end{cases} \tag{26}$$

The estimator returns `beyond` otherwise. This variant separates the capability estimate from the model's current tendency to abstain, but requires an additional group of K_b rollouts.

IDK-enabled final-answer boundary. We reuse the K completions sampled under the IDK-enabled prompt and inspect their delivered final answers:

$$\widehat{B}^{\text{idk}}_{K}(x) = \begin{cases} \texttt{within}, & \exists k,\ T(x, y_1^{(k)}) = 1, \\ \texttt{beyond}, & \text{otherwise.} \end{cases} \tag{27}$$

This estimator also returns `beyond` otherwise and requires no additional sampling. Unlike the other two estimators, however, a completion can fail to satisfy $T = 1$ either because it answers incorrectly or because it abstains.

E.2 Dependence on the Abstention Probability

Let

$$q_\theta(x) = \Pr_{\pi_\theta}[y_1 = \text{IDK} \mid x] \tag{28}$$

denote the probability of abstention, and let

$$p^{\text{ans}}_\theta(x) = \Pr_{\pi_\theta}[T(x, y_1) = 1 \mid y_1 \neq \text{IDK}, x] \tag{29}$$

denote correctness conditional on producing a substantive answer. For the IDK-enabled estimator,

$$\Pr\left(\widehat{B}^{\text{idk}}_{K}(x) = \texttt{beyond}\right) = \left(1 - (1 - q_\theta(x))p^{\text{ans}}_\theta(x)\right)^K. \tag{30}$$

Consequently, the probability of a `beyond` label increases with $q_\theta(x)$ even when the model's correctness conditional on answering remains unchanged. The estimator therefore measures the joint event of answering and being correct, rather than answer capability independently of abstention.

Proposition 1 (Sign reversal under alternative boundary estimators)**.** *Consider a prompt for which an abstention-free estimator, $\widehat{B}^{\text{draft}}_K$ or $\widehat{B}^{\text{forced}}_{K_b}$, returns `within`. Suppose that an IDK-enabled policy-gradient group contains m abstentions and $K - m$ incorrect substantive answers, with $0 < m < K$. Let $A^{(k)}_{\text{idk}}$ denote the centered contribution of the boundary-aware abstention reward to the group-relative advantage of completion k, before the common positive normalization factor. Then*

$$A^{(k)}_{\text{idk}} = \begin{cases} +\left(1 - \frac{m}{K}\right), & y_1^{(k)} = \text{IDK}, \quad \widehat{B}^{\text{idk}}_K = \texttt{beyond}, \\ -\left(1 - \frac{m}{K}\right), & y_1^{(k)} = \text{IDK}, \quad \widehat{B}^{\text{draft}}_K \text{ or } \widehat{B}^{\text{forced}}_{K_b} = \texttt{within}, \end{cases} \tag{31}$$

while the substantive-answer completions receive the complementary centered values $-m/K$ and $+m/K$, respectively.

If $m = K$, the abstention reward is constant across the group under each estimator and therefore contributes zero to the centered advantage. Moreover, the probability that an IDK-enabled group contains no correct substantive answer is given by Equation (30) *and is increasing in $q_\theta(x)$.*

Proof. Under $\widehat{B}^{\text{idk}}_K$, the group contains no verified-correct delivered answer, so the estimator returns `beyond`. The boundary-aware reward assigns $+1$ to each abstention and 0 to each incorrect substantive answer. Its group mean is therefore m/K, yielding centered values $1 - m/K$ and $-m/K$.

Table 13: Boundary-estimator audit on the 7,473 prompts of the GSM8K training split, evaluated at the same Qwen2.5-7B checkpoint with $K = K_b = 16$. In panel (b), $w \to b$ counts prompts labelled `within` by the reference estimator and `beyond` by the compared estimator, while $b \to w$ denotes the converse. The forced-answer estimator is the reference for the first two rows, and the IDK-enabled estimator is the reference for the third row.

(a) Label distribution

Estimator	within	beyond	Extra rollouts
$\widehat{B}_{16}^{\text{draft}}$	6142 (82.2%)	1331 (17.8%)	0
$\widehat{B}_{16}^{\text{forced}}$	6091 (81.5%)	1382 (18.5%)	K_b
$\widehat{B}_{16}^{\text{idk}}$	5704 (76.3%)	1769 (23.7%)	0

(b) Pairwise agreement

Pair	Agreement	κ	w→b / b→w
draft vs. forced	97.8%	.926	57 / 108
idk vs. forced	93.9%	.818	421 / 34
draft vs. idk	93.1%	.790	40 / 478

Under an abstention-free estimator that returns `within`, the same policy-gradient completions are evaluated against a `within` boundary label. Abstentions receive -1 and incorrect substantive answers receive 0. The group mean is $-m/K$, yielding centered values $-(1 - m/K)$ and $+m/K$. When $m = K$, the raw abstention reward is constant across all completions, so centering removes its contribution. The final claim follows directly from Equation (30). □

Proposition 1 shows that the estimators differ not only in sampling variance. On a prompt for which an abstention-free sample demonstrates a correct solution, an IDK-enabled group containing only abstentions and incorrect answers can assign the opposite centered reward signal to abstention.

E.3 Agreement among Boundary Estimators

We first apply all three estimators to the same 7,473 prompts from the GSM8K training split using the same final Qwen2.5-7B checkpoint and $K = K_b = 16$.

Table 13 shows that the draft-based and forced-answer estimators are closely aligned. They agree on 97.8% of prompts with $\kappa = .926$, and their disagreements occur in both directions.

The IDK-enabled estimator exhibits a more asymmetric difference. Relative to the forced-answer estimator, 421 prompts move from `within` to `beyond`, while only 34 move in the opposite direction. Among these 421 prompts, 388 (92.2%) contain at least 15 abstentions among the 16 IDK-enabled completions. Thus, most of the one-directional disagreements are associated with the policy's abstention behavior rather than with a different verified solution being found.

Across all 1,769 prompts labelled `beyond` by $\widehat{B}_{16}^{\text{idk}}$, 21.9% are prompts for which an abstention-free estimator finds a verified-correct answer. This directly measures the abstention-dependent component of the IDK-enabled boundary labels.

E.4 Controlled Training Comparison

We next train three otherwise identical Qwen2.5-7B variants on GSM8K, changing only the boundary estimator used by the boundary-aware abstention reward.

The draft-based and forced-answer variants converge to similar selective behavior. Their coverage decreases during the early training stage and then stabilizes near 94%, while their final Effective Accuracy differs by only 0.1 points.

The IDK-enabled variant follows a different trajectory. Its coverage decreases monotonically from 96.8% at step 60 to 76.4% at step 240. Over the same checkpoints, the proportion of training prompts labelled `beyond` rises from 18.6% to 23.7%, 31.4%, and 39.2%. This behavior is consistent with the

Table 14: Controlled comparison of three boundary estimators on GSM8K with Qwen2.5-7B. The three training runs use the same initialization, optimization configuration, reward weights, and policy-gradient group size; only the operational boundary estimator is changed. Coverage is evaluated on the held-out split at four training checkpoints. H-Proxy and Effective Accuracy are reported at step 240.

Estimator	**Coverage at training step**				**H-Proxy↓**	**Eff.Acc.↑**
	60	120	180	240		
$\widehat{B}^{\text{forced}}_{K_b}$	97.1	94.6	93.9	93.8	11.0	85.0
$\widehat{B}^{\text{draft}}_{K}$	97.3	94.9	94.3	94.2	11.4	85.1
$\widehat{B}^{\text{idk}}_{K}$	96.8	91.2	84.7	76.4	9.8	70.2

Table 15: Ablation study of REIN's reliability alignment components on GSM8K (Qwen2.5-7B). Each row removes one objective component. Removing Reflection Veracity increases hallucination errors, while removing IDK alignment affects answer-abstain calibration.

Configuration	Sel.Acc↑	H-Proxy ↓	Cov.↑	Eff.Acc
Full REIN	**90.6**	**0.11**	93.8	**85.0**
w/o Accuracy	78.5	0.16	92.0	72.2
w/o XML Structure	88.4	0.14	93.0	82.2
w/o Refl. Presence	89.8	0.13	93.4	83.9
w/o Refl. Veracity	90.2	0.20	93.5	84.3
w/o IDK alignment	85.2	0.19	100	85.2

dependence on $q_\theta(x)$ in Equation (30): as abstention becomes more frequent, an IDK-enabled group is more likely to contain no correct substantive answer.

At step 240, the IDK-enabled variant attains the lowest H-Proxy, 9.8, but answers only 76.4% of the evaluation questions. Its Effective Accuracy is consequently 70.2, compared with 85.0 and 85.1 for the forced-answer and draft-based variants. This comparison illustrates why H-Proxy and Selective Accuracy must be interpreted jointly with Coverage and Effective Accuracy: a lower conditional error measure can coincide with substantially lower overall utility when it is achieved through extensive abstention.

Taken together, the label audit and controlled training comparison show that the operational definition of the sampled boundary affects both the estimated boundary labels and the learned answer–abstain trade-off. The draft-based and forced-answer estimators behave similarly in these experiments, whereas the IDK-enabled estimator is more strongly coupled to the policy's current abstention rate.

F Ablation and Diagnostic Analysis

Key findings (1) **Accuracy reward is essential**: removing it causes a large drop in selective accuracy (-12.1), confirming that correctness signal is the primary learning driver. (2) **Reflection Veracity controls hallucination**: removing it has minimal effect on selective accuracy (-0.4) but nearly doubles the hallucination rate ($0.11 \rightarrow 0.20$), showing this reward is specifically responsible for aligning the reflection's self-assessment with actual correctness. (3) **IDK alignment enables abstention**: without it, the model never abstains (Cov.=100%) and achieves identical results to "REIN w/o IDK" in the main table, confirming this reward is the sole driver of IDK behavior. (4) **XML Structure supports reflection quality**: removing it reduces effective accuracy by 2.8 points and increases hallucination ($0.11 \rightarrow 0.14$), indicating that consistent formatting is a prerequisite for effective self-checking.

F.1 Sensitivity to Boundary-Probe Size K_b

The forced-answer boundary in Eq. (7) is estimated from an IDK-free probe group of size K_b. We vary K_b while keeping the policy-gradient group fixed at $K = 16$. Table 16 reports the resulting boundary estimates and downstream performance on GSM8K and MATH.

Larger K_b yields more accurate boundary detection on both benchmarks ($+11.3$ on GSM8K, $+13.2$ on MATH from K_b=4 to 16), which translates to better selective accuracy and lower hallucination rate. Boundary detection is inherently harder on MATH (85.3% vs. 89.7% at K_b=16) due to its broader difficulty distribution, where more questions lie near the model's capability frontier. We use $K_b = 16$ as the default, which achieves a good balance between detection accuracy and computational efficiency.

Table 16: Effect of group size K_b on boundary estimation and downstream metrics. Larger K_b improves boundary detection but increases sampling cost.

Dataset	K_b	Sel.Acc↑	Halluc.↓	Cov.↑	Bnd.Acc
GSM8K	4	87.3	0.15	95.8	78.4
	8	89.1	0.13	94.5	84.2
	16	**90.6**	**0.11**	93.8	**89.7**
MATH	4	56.5	0.21	85.4	72.1
	8	59.1	0.18	84.0	79.5
	16	**61.**	**0.15**	83.5	**85.3**

F.2 Difficulty-Stratified Abstention

Table 17: IDK rate stratified by difficulty across benchmarks (7B). The model's abstention rate increases monotonically with question difficulty, confirming calibrated knowledge-boundary awareness.

Dataset	Difficulty Bin	IDK Rate (%)
GSM8K (by steps)	2-step	2.1
	3–4 step	5.4
	5+ step	14.8
MATH (by level)	Level 1	3.2
	Level 2	8.5
	Level 3	15.1
	Level 4	22.4
	Level 5	31.8
ARC-C (by grade)	Elementary	2.8
	Middle school	6.5
	High school	13.4

G Reflection Causality and Draft-to-Final Transition

The main analysis attributes part of REIN's reliability improvement to the `<reflection>` span. This section investigates whether reflection causally affects the downstream finalization process. Rather than treating reflection as an auxiliary textual annotation, we perform inference-time interventions on the reflection span and trace how different reflection states influence the transition from the initial draft y_0 to the final delivered answer y_1.

G.1 Intervening on the Reflection

We take the trained REIN checkpoint and, at inference time only, replace the reflection span with controlled substitutes before the final answer is decoded. Six interventions are compared against the intact model.

A1 removes the reflection span and emits empty reflection tags. A2 and A3 force the stance token to `correct` and `wrong`, respectively, while allowing the remaining reflection content to be generated normally. A4 flips the stance produced by the model while preserving the remaining reflection text. A5 transplants the entire reflection generated for a different question. The transplanted reflection is selected such that its parsed stance matches the original stance of the target example, preserving the

Table 18: Inference-time interventions on the reflection span, evaluated on Qwen2.5-7B. Each intervention is applied after generating the initial draft y_0, isolating the causal effect of reflection on final decision making. Repair denotes the fraction of incorrect drafts that become correct after finalization, while $\mathrm{Corrupt}$ denotes the fraction of correct drafts that become incorrect.

ID	Intervention	Sel.Acc.↑	Cov.↑	Eff.Acc.↑	Repair↑	Corrupt↓
A0	Intact	83.9	91.9	77.0	11.0	0.9
A1	Reflection excised	77.8	97.8	76.1	0.3	0.1
A2	Stance forced `correct`	77.1	98.6	76.0	0.9	0.4
A3	Stance forced `wrong`	88.2	71.5	63.0	14.2	6.1
A4	Stance flipped	74.8	90.2	67.5	5.8	5.4
A5	Reflection transplanted	79.5	92.1	73.2	4.1	3.6
A6	Reflection paraphrased	83.5	92.0	76.8	10.4	1.0

stance while scrambling the question-specific content. A6 paraphrases the model's own reflection using a separate rewriter, preserving the original meaning and stance.

The interventions are designed to separate three possible interpretations of reflection. If reflection is merely a formatting artifact, removing or replacing it should have little effect on final decisions. If only the discrete stance matters, stance-matched transplantation should preserve the original behavior. If the reflection contains question-specific information that guides finalization, removing or scrambling its content should reduce the quality of downstream decisions.

Table 3 shows that reflection is not merely a decorative intermediate text span. Removing the reflection (A1) substantially reduces the model's ability to revise incorrect drafts: the repair rate decreases from 11.0% to 0.3%. At the same time, the model becomes less likely to change its initial draft, indicating that reflection provides a mechanism for deciding whether the current reasoning should be preserved, revised, or rejected.

Forcing the stance also changes downstream behavior in the expected direction. Forcing `correct` (A2) suppresses revisions and abstention, whereas forcing `wrong` (A3) substantially increases revision and abstention. However, excessive rejection can harm overall utility: although A3 achieves higher selective accuracy, its effective accuracy decreases because many correct drafts are unnecessarily discarded. This demonstrates that the reflection signal affects the finalization policy rather than simply annotating an already determined answer.

The contrast between A5 and A6 provides evidence that reflection content, rather than only the discrete stance, contributes to final decision making. Paraphrasing the model's own reflection (A6) produces behavior close to the intact model, while transplanting a stance-matched reflection from another question (A5) substantially reduces repair performance. The model continues to revise at a similar rate but loses the ability to make reliable corrections, suggesting that question-specific reflection content provides actionable information for finalization.

G.2 Draft-to-Final Transition Analysis

To further analyze how reflection changes the final output, we trace each instance from the initial draft answer y_0 to the final delivered answer y_1.

Among 747 initially incorrect drafts, 82 become correct after reflection, yielding a repair rate of

$$\frac{82}{747} = 11.0\%.$$

In contrast, only 22 out of 2454 initially correct drafts become incorrect, corresponding to a corruption rate of

$$\frac{22}{2454} = 0.9\%.$$

Table 19: Draft-to-final transition matrix of REIN on Qwen2.5-7B pooled over the four benchmarks. Rows indicate the correctness status of the initial draft y_0, while columns indicate the final delivered outcome y_1. The matrix characterizes how reflection-guided finalization transforms initial reasoning states into final decisions. The column includes 236 explicit abstentions and 26 invalid outputs.

Draft y_0	y_1 correct	y_1 wrong	y_1 IDK	**Total**
correct	2399	22	33	2454
wrong	82	455	210	747
unparsable	0	0	19	19
Total	2481	477	262	3220

Table 20: Reflection-conditioned finalization transitions of REIN on Qwen2.5-7B. The table shows how different reflection stances influence whether the model preserves, revises, or abstains from the initial draft y_0.

Stance	**Draft** y_0	y_1 correct	y_1 wrong	y_1 IDK	**Total**
`correct`	correct	2374	6	7	2387
	wrong	8	49	1	58
`wrong`	correct	16	14	14	44
	wrong	60	263	56	379
`uncertain`	correct	9	2	12	23
	wrong	14	143	153	310
$\perp$	unparsable	0	0	19	19

Therefore, reflection improves finalization quality by correcting incorrect reasoning outcomes substantially more often than it damages correct ones.

The transition analysis also shows that abstention is targeted rather than uniform. Among the 262 non-substantive final outputs, most originate from incorrect or unreliable initial drafts. This indicates that the abstention mechanism operates jointly with reflection: unreliable reasoning trajectories are more likely to be removed rather than forced into unsupported final answers.

G.3 Reflection-Conditioned Finalization Transition

We further condition the transition process on the predicted reflection stance to understand how different reliability judgments influence subsequent decisions.

The conditioned transition analysis reveals that reflection acts as a control signal for downstream decision making. When the model assigns a `correct` stance, it tends to preserve the original draft. Conversely, a `wrong` or `uncertain` stance increases the probability of revision or abstention, especially for incorrect drafts.

However, reflection primarily functions as an error detector rather than a complete solver. Among the cases where the model identifies an incorrect draft through a `wrong` stance, many remain incorrect after finalization. This indicates that recognizing unreliable reasoning is easier than repairing it completely.

Overall, these results show that reflection and abstention play complementary roles: reflection provides an internal reliability assessment, while abstention converts uncertain or unsupported reasoning states into safer final decisions.

H Finite-Sample Interpretation and Rollout-Budget Sensitivity

Notation. We use the same notation as in Section 3. For a prompt x, $y^\star(x)$ denotes the gold answer, $T(x, y) = \mathbb{1}[y = y^\star(x)]$ denotes answer correctness, and $p(x) = \mathbb{E}[T(x, y) \mid x]$ denotes

the per-prompt correctness probability. The parsed reflection judgment is denoted by $J(r) \in \{0, 1\}$, where 1 means reliable and 0 means unreliable.

H.1 Finite-Sample Interpretation and Rollout-Budget Sensitivity

Let

$$p_\theta(x) = \Pr_{z \sim \pi_\theta(\cdot|x)} [T(x, y_1) = 1] \tag{32}$$

denote the probability that the current policy produces a verified-correct final answer for prompt x. Under conditionally independent sampling, the probability that none of the K completions succeeds is

$$\Pr\left(\widehat{B}_K(x) = \texttt{beyond} \mid x\right) = (1 - p_\theta(x))^K. \tag{33}$$

Therefore, a no-success group does not imply that $p_\theta(x) = 0$. Instead, it provides increasingly strong evidence of a low policy-level success probability as the rollout budget increases.

When zero successes are observed among K samples, a one-sided $(1 - \alpha)$ upper confidence bound on the success probability is

$$p_\theta(x) \leq 1 - \alpha^{1/K}. \tag{34}$$

At the 95% confidence level, the corresponding upper bounds are 31.23%, 17.07%, 8.94%, and 4.57% for $K = 8, 16, 32$, and 64, respectively. Larger rollout budgets therefore reduce finite-sampling uncertainty, but increase rollout generation and verification costs approximately linearly. We use $K = 16$ during GRPO training as a practical compromise between boundary-estimation stability and computational cost.

We further evaluate the sensitivity of the empirical boundary using $K \in 1, 2, 4, 8, 16, 32, 64$. For each budget, we report the proportion of prompts for which no verified solution is observed. We also compare the training budget $K = 16$ with the higher-budget $K = 64$ reference by reporting the proportion of $K = 16$ beyond-boundary prompts that remain beyond at $K = 64$, together with the proportion that recover to within-boundary status. The $K = 64$ result is used only as a higher-budget empirical reference and should not be interpreted as an oracle label of intrinsic knowledge absence.

H.2 Confidence Bound of the Group-level Boundary Indicator

The group-level boundary indicator $B(x)$ estimates whether a prompt lies within the model's effective capability under the current policy and sampling budget. Although $B(x)$ does not represent the true epistemic boundary of the model, we show that observing no correct samples provides a probabilistic upper bound on the probability of producing a correct answer.

Lemma 1 (Confidence bound of $B(x)$)**.** *Assume that K completions are independently sampled from the policy*

$$\{(c^{(k)}, r^{(k)}, y^{(k)})\}_{k=1}^K \sim \pi_\theta(\cdot|x),$$

where each completion has success probability

$$p(x) = \Pr[y = y^\star(x) \mid x, \pi_\theta].$$

If the group-level boundary indicator satisfies

$$B(x) = \text{beyond},$$

meaning that none of the K sampled completions produces the correct answer, then, with confidence level $1 - \alpha$,

$$p(x) \leq 1 - \alpha^{1/K}.$$

Proof. For a single completion sampled from $\pi_\theta(\cdot|x)$, the probability of obtaining an incorrect answer is

$$1 - p(x).$$

Since the K samples are independent, the probability that all sampled completions fail to produce the correct answer is

$$\Pr[B(x) = \text{beyond}] = (1 - p(x))^K.$$

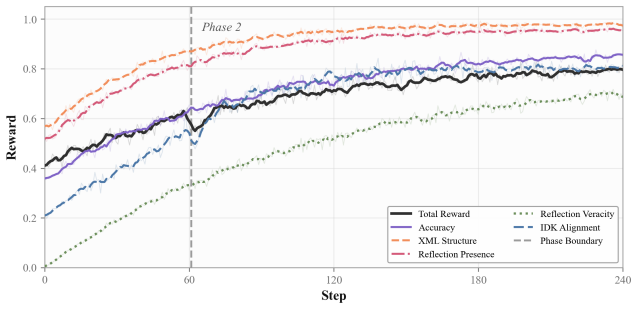

Figure 5: Reward Composition Dynamics During REIN Training (Qwen2.5-7B)

Observing $B(x) = \text{beyond}$ means that zero successful samples are obtained in K independent trials. For a confidence level $1 - \alpha$, the failure probability satisfies

$$(1 - p(x))^K \geq \alpha.$$

Taking the K-th root on both sides gives

$$1 - p(x) \geq \alpha^{1/K}.$$

Therefore,

$$p(x) \leq 1 - \alpha^{1/K}.$$

Hence, when no correct answer is observed among K independent samples, the probability that the model can correctly answer the prompt is bounded above by $1 - \alpha^{1/K}$ with confidence level $1 - \alpha$. □

Example. For $K = 16$ samples and a 90% confidence level ($\alpha = 0.10$), the upper bound becomes

$$p(x) \leq 1 - 0.1^{1/16} \approx 0.134.$$

Therefore, if none of the 16 sampled completions produces the correct answer, we can infer with 90% confidence that the probability of the current policy solving the prompt is at most 13.4%.

I Implementation Details and Computational Resources

I.1 Implemented Variants and Scope

The repository contains several implemented variants beyond the minimal baseline described above. These include 7B and 8B training configs, veracity-augmented reward variants, and additional prompt templates that increase the amount of reflective structure. We treat these as implementation variants rather than as fully reported analyses unless accompanied by committed experimental outputs.

I.1.1 Reward Configuration

The final REIN objective contains five reward terms: answer correctness, XML structure, reflection presence, reflection veracity, and boundary-aware abstention, as defined in Eq. (9). The first four terms are active throughout GRPO training. The abstention term is disabled in Phase 1 ($w_{\text{idk}} = 0$) and activated in Phase 2 ($w_{\text{idk}} = 4.0$). Earlier three- and four-reward configurations are implementation variants and are not used for the headline REIN results.

I.1.2 Training and Evaluation Details

To avoid test leakage, all reported results are evaluated on the official test splits.For cold-start formatting, we sample 630 examples from the training split of each benchmark, resulting in 2,520 examples in total. For reward-based alignment, we use the remaining GSM8K training examples.Therefore, the

Table 21: We have a shared configuration across all the backbones unless we say otherwise. Other implementation details will adhere to the released configurations unless otherwise indicated.

Item	**Setting**
Policy-gradient samples (K)	16
Boundary-probe samples (K_b)	16 (Phase 2 only)
Max length	2048
Temperature	0.7
LoRA (r, α)	$(16, 32)$
GRPO β	0.05
r_{acc} weight	8.0
r_{xml} weight	3.0
r_{pres} weight	1.5
r_{vrcty} weight	2.0
r_{idk} weight	0 (Phase 1), 4.0 (Phase 2)

Table 22: Effect of the Phase 2 boundary-probe budget on GSM8K with Qwen2.5-7B. The policy-gradient group is fixed at $K = 16$. $K_b = 0$ denotes the variant that reuses the IDK-enabled policy group instead of sampling an independent forced-answer probe; it is therefore a different boundary estimator rather than a zero-budget version of the forced-answer estimator.

K_b	Sel.Acc.↑	Cov.↑	Eff.Acc.↑	GPU-h
0	91.9	76.4	70.2	21.4
4	87.3	95.8	83.6	25.5
8	89.1	94.5	84.2	29.6
16	90.6	93.8	85.0	37.9

results on MATH, StrategyQA, and ARC-Challenge mainly evaluate whether the reward-optimized reliability behavior can transfer beyond the GRPO training domain. The training config and hyperparameter are shown in table 21.

As per the instructions provided in the main evaluation script, we use the single-pass generation decoding protocol with `do_sample=False`. The main evaluation script supports the modes of reflection, instruction, and no-CoT. It produces the final answers using `boxed{}` for MATH/AIME-style evaluation and delimited by `####` for GSM8K. There is no use of test-time majority voting, self-consistency, or multi-round critique loops.

I.1.3 Two-Phase Curriculum

GRPO training proceeds in two phases with a fixed transition after 60 optimization steps. In Phase 1, the policy is sampled under the IDK-free prompt and the abstention term is disabled ($w_{\text{idk}} = 0$). Because the policy-gradient group is already sampled without the abstention option, it can also be used to evaluate the forced-answer boundary without additional rollouts.

In Phase 2, the policy prompt enables abstention and w_{idk} is set to 4.0. The policy-gradient group contains $K = 16$ completions, while a separate group of $K_b = 16$ completions is sampled under the IDK-free prompt to estimate $\widehat{B}^{\text{forced}}_{K_b}(x)$. The boundary-probe completions are used only to construct the boundary label and are not included in the GRPO normalization group.

I.1.4 Inference Cost

We additionally measure single-stream inference latency under vLLM with batch size 1 and no KV-cache reuse across independently sampled responses. Latency is normalized to greedy decoding. The boundary probe used by REIN is a training-time construction and is not executed at inference.

Table 23: Inference cost on Qwen2.5-7B, averaged over the four benchmarks. Latency is single-stream wall-clock time under vLLM at batch size 1, normalized to greedy decoding. The self-consistency latency is a serial measurement rather than a FLOP ratio.

Method	#fwd	Lat.	Sel.Acc.↑	Cov.↑	Eff.Acc.↑
Base	1	1.00×	64.9	100.0	64.9
+ self-verification prompt	2	1.93×	65.8	100.0	65.8
+ Self-Refine	∼7	6.81×	64.8	100.0	64.8
+ Self-Consistency @16	16	15.24×	71.7	100.0	71.7
Reflection-only	1	1.19×	72.2	100.0	72.2
REIN	1	1.21×	78.6	90.4	71.5
REIN + Self-Consistency @16	17	15.41×	83.1	89.8	75.0

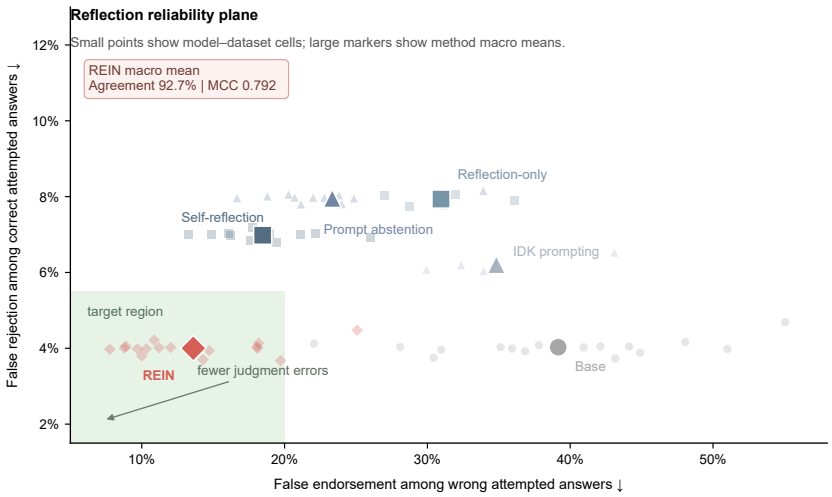

Figure 6: Reflection judgments align with answer outcomes across backbones and benchmarks. Each small point denotes one observed model–dataset cell, while large markers report method-level macro averages. False endorsement measures the fraction of wrong attempted answers judged reliable; false rejection measures the fraction of correct attempted answers judged unreliable. Lower values are better. Comparisons are descriptive because the available numbers of model–dataset cells differ across methods.

I.1.5 Diagnostic Scope

In addition to task-level metrics, the repository includes a diagnostic script for evaluating whether reflections identify the erroneous step and error type on a manually annotated subset of incorrect generations. We view this as a useful analysis tool for future experiments, but not as a replacement for the main benchmark metrics of accuracy, MCC, and hallucination rate.

I.2 Hardware and Duration

We conduct our experiments on a computing system with NVIDIA RTX PRO™ 6000 Blackwell GPUs, each equipped with 96GB of VRAM. We trained the models on two GPUs and the average duration of one full training of a model with final learning rates and exhaustive evaluation around 30h.

J Robustness of the Measurement

Every number in this paper passes through a prompt, a decoding pass and a parser before it becomes a metric. Each of those is a degree of freedom that could be tuned, after the fact, in a direction that favours the proposed method. This appendix fixes all three and measures how much the headline

Table 24: Abstention surface forms, REIN on Qwen2.5-7B pooled over the four benchmarks. Detect. is the parser's recall on held-out completions manually labelled as abstentions; Emitted is the share of REIN's 236 explicit abstentions taking that form under the canonical prompt, so the column is 229/4/2/1/0/0/0/0 expressed as percentages. Both columns are percentages.

Surface form	**Detect.↑**	**Emitted**
`I don't know` (canonical)	100.0	97.0
`I do not know`	100.0	1.7
`\boxed{\text{I don't know}}`	99.8	0.8
`I'm not sure`	99.6	0.4
`Unknown`	99.1	0.0
`Cannot be determined`	98.4	0.0
`N/A`	97.2	0.0
`IDK`	96.5	0.0

results move when they are varied: across abstention surface forms, across the output conflicts the parser must adjudicate, across paraphrases of the evaluation prompt, and across four parser implementations.

J.1 Abstention Surface Forms

The abstention parser recognizes a fixed lexicon rather than a single string, because a model that abstains in words the parser does not know would be scored as producing an invalid output and would lose coverage it has in fact earned.

Table 24 shows the question is close to moot for a trained model: 97.0% of REIN's abstentions use the canonical string the reward was written against, and 99.6% fall in the top three forms. Restricting the parser to the canonical string alone reclassifies the seven non-canonical abstentions as invalid outputs. It cannot move coverage, because Eq. (1) makes abstentions and invalid outputs alike non-substantive, and it cannot move selective accuracy or H-Proxy, because neither outcome enters their numerators or denominators; the lexicon governs only the abstention–invalid split reported in Appendix G. What the lexicon does affect is the prompted baselines, whose abstentions are far less uniform, and for which a canonical-only parser would misclassify genuine refusals as malformed output. Going the other way and instructing REIN to abstain in each of the eight forms in turn leaves macro coverage in $[89.9, 90.9]$ and H-Proxy in $[10.9, 11.8]$, so no reported conclusion depends on which phrase the abstention is required to take. The four forms the trained model never produces are in the lexicon for the baselines alone.

J.2 Conflicting Outputs

A completion can contradict itself, and how those cases are resolved is a choice that must be declared rather than buried in an implementation.

Self-contradiction is rare, 1.65% of completions, and the three adjudication rules of Table 25 separate the headline metrics by at most 0.4 points. The conservative default is not the one that flatters REIN: the permissive rule would report a lower H-Proxy (10.9) and higher coverage (91.0). We keep the conservative rule because it is the only one of the three that cannot be accused of resolving ambiguity in the proposed method's favour, and we apply it identically to every baseline.

Class C5 contains eight completions with an endorsing reflection and an explicit abstention in the final answer. These cases are counted as abstentions for outcome-level metrics. Their contribution to H-Proxy depends only on the pre-reflection draft: an instance belongs to the H-Proxy population whenever y_0 is a parseable incorrect draft, regardless of whether the final output y_1 is substantive or `IDK`.

(a) Taxonomy and incidence

Table 25: Self-contradictory outputs from the REIN model on Qwen2.5-7B, pooled over the four benchmarks ($N = 3220$). The conservative rule treats C1–C3 and C6 as invalid and C4 as a non-endorsement, which is the convention used everywhere else in this paper; the permissive rule resolves each conflict in the model's favour by taking the first well-formed answer and reading C4 as an endorsement; the third rule removes the 53 affected instances from all denominators.

Conflict	Description	Count	%
C1	abstention marker inside a substantive answer	11	0.34
C2	multiple distinct answers in `<answer>`	7	0.22
C3	answer only in `<reflection>`	5	0.16
C4	stance absent or unparsable	19	0.59
C5	endorsing stance with explicit IDK	8	0.25
C6	duplicated or nested tag structure	3	0.09
	total	**53**	**1.65**

(b) Sensitivity to the adjudication rule

Rule	Sel.Acc.↑	H-Proxy↓	Cov.↑	Eff.Acc.↑
Conservative (default)	78.6	11.3	90.4	71.5
Permissive	78.9	10.9	91.0	72.0
Drop conflicts	78.8	11.2	90.1	71.3

Table 26: Stability across five evaluation templates on Qwen2.5-7B, macro-averaged over the four benchmarks. Each entry is the mean over templates ± half of the observed range.

Method	Sel.Acc.↑	Cov.↑	Eff.Acc.↑
IDK Prompting	74.6 ± 3.4	83.9 ± 7.6	63.6 ± 5.2
R-Tuning	71.5 ± 1.4	86.4 ± 2.8	62.3 ± 2.1
TruthRL	73.9 ± 1.1	90.7 ± 2.0	67.4 ± 1.6
Reflection-only	72.2 ± 0.8	100.0 ± 0.0	72.2 ± 0.8
REIN	$\mathbf{78.6 \pm 0.7}$	90.4 ± 1.2	71.5 ± 1.0

J.3 Prompt Paraphrase

We test five evaluation templates that preserve the task while rephrasing the instruction, reordering the formatting requirements, and varying the wording of the abstention clause. Table 26 reports the mean over the five templates together with half of the observed range.

IDK Prompting has a 15.2-point coverage range across the five templates, compared with 2.4 points for REIN. R-Tuning and TruthRL fall between these two cases. The smaller variation for the trained methods indicates that their abstention behavior is less sensitive to the exact wording of the evaluation instruction than prompt-only abstention.

J.4 Parser Implementation

The four parsers of Table 27 span the range from a literal matcher to a model adjudicator and separate the headline metrics by at most 0.6 points of selective accuracy, 0.7 of H-Proxy and 1.7 of coverage. The outlier is V2, which by removing normalization discards well-formed answers that differ from the reference only in spacing or formatting, and it moves every method in the same direction.

Applying all four parsers to all thirteen methods of Table 11 leaves the H-Proxy ordering unchanged, with Kendall $\tau = 1.00$ between every pair of variants. The strict parser is the default because it is the only one that makes no judgement call at evaluation time; V3 is reported to show that the residual

Table 27: Four parser implementations applied to one fixed set of REIN completions on Qwen2.5-7B, macro-averaged over the four benchmarks. V1 additionally accepts case and punctuation variants and the full abstention lexicon of Table 24; V2 removes all normalization and matches the answer delimiters literally; V3 refers the 53 conflicts of Table 25 and every unparsable span to a separate model call that sees no reference answer. Unparsable is a percentage of all completions.

	Parser	Unparsable	Sel.Acc.↑	H-Proxy↓	Cov.↑	Eff.Acc.↑
V0	strict (default)	0.59	78.6	11.3	90.4	71.5
V1	lenient	0.21	78.5	11.4	90.9	71.9
V2	regex only	2.14	78.1	11.9	89.2	70.4
V3	model-adjudicated	0.06	78.7	11.2	90.8	71.9

unparsable mass, which the strict parser charges against coverage, is small enough that resolving it perfectly would gain REIN 0.4 points of coverage and cost it 0.1 of H-Proxy.

K Qualitative Examples

A representative case of correcting reasoning errors within the knowledge boundary

System: You are a careful reasoner. For every problem, reply with EXACTLY this structure: `<think>...</think> <reflection>...</reflection> <answer>...</answer>`.

User: A store has 120 apples. They sell $\frac{1}{3}$ of the apples in the morning and $\frac{1}{4}$ of the remaining apples in the afternoon. How many apples are left?

Assistant:
`<think>`
Morning: $120 \times \frac{1}{3} = 40$ sold, leaving 80 apples. Afternoon: $120 \times \frac{1}{4} = 30$ sold, leaving $80 - 30 = 50$ apples.
`</think> <reflection>`

Wait. The afternoon sale should be $\frac{1}{4}$ of the remaining 80 apples, not the original 120. Recalculating: $80 \times \frac{1}{4} = 20$ sold in the afternoon. Remaining apples: $80 - 20 = 60$. The initial calculation contains an error in step 2.
`</reflection> <answer>`
`60`
`</answer>`

A representative case of the REIN framework triggering selective abstention

System: You are a careful reasoner. For every problem, reply with EXACTLY this structure: `<think>...</think> <reflection>...</reflection> <answer>...</answer>`.

User: Find the unique 3-digit integer N such that N is twice the sum of its digits cubed.

Assistant:
`<think>`
Such Armstrong-like number is uncommon. I would need to search digit-by-digit, but this is computationally heavy without a calculator.
`</think> <reflection>` **Unsure.** The search space requires careful enumeration that I cannot do reliably by hand. `</reflection> <answer>`
`I don't know`
`</answer>`

An empirical failure mode of the self-correction mechanism within the REIN framework

System: You are a careful reasoner. For every problem, reply with EXACTLY this structure: `<think>...</think> <reflection>...</reflection> <answer>...</answer>`.

User: Solve: $x^2 - 4 = 0$.

Assistant:
`<think>`
$x = \pm 2$. Recorded: 2.
`</think> <reflection>`
Wrong. There are two roots ± 2; I only listed one.
`</reflection> <answer>`
`\boxed{2}`
`</answer>`

L Implementation Details for Baselines.

To ensure a fair comparison and reproducibility, all trainable baselines are implemented with the same backbone models, training data, and parameter-efficient fine-tuning strategy as REIN. Specifically, we use LoRA-based fine-tuning and keep the optimization configurations consistent whenever applicable. Each baseline is trained using its original objective, while the inference-time decoding and evaluation protocol are unified across all methods. The implementation details of each baseline are described as follows:

- **IDK Prompting.** IDK Prompting is an inference-time baseline that introduces explicit abstention instructions into the prompt. The model is encouraged to respond with "I don't know" when it is uncertain or when the problem is beyond its capability. No additional training is performed, and the original model parameters remain unchanged.
- **Reflection-only.** Reflection-only preserves the reflection format and reflection generation process used in REIN, allowing the model to assess the reliability of its own reasoning. However, it removes the boundary-aware IDK component and does not include the knowledge-boundary-based abstention reward. Therefore, this baseline evaluates the contribution of reflection alignment alone without selective abstention learning.
- **R-Tuning.** We reproduce R-Tuning following its original uncertainty-aware instruction tuning objective. The model is trained to explicitly produce "I don't know" responses on questions beyond its capability while preserving answer generation on solvable questions.
- **TruthRL.** We adopt the same GRPO optimization framework and training configuration as REIN to ensure a controlled comparison. The only difference is the reward formulation. Following the original TruthRL design, the reward assigns positive feedback to correct answers, neutral reward to explicit abstention, and negative reward to incorrect answers.

M Prompts

M.1 IDK-Free Training and Boundary-Probe Prompt

The following prompt is used for the Phase 1 policy group and for the IDK-free boundary-probe group in Phase 2:

> Answer the question using the demonstrated style. Reply with exactly one `<think>...</think><reflection>...</reflection><answer>...</answer>` sequence and nothing else.
> In `<reflection>`, briefly audit the reasoning and begin with `Correct` if reliable.
> In `<answer>`, write the final numerical answer after `####`.

M.2 IDK-Enabled Phase 2 Prompt

The Phase 2 policy-gradient group uses the same structured format but allows explicit abstention:

Answer the question using the demonstrated style. Reply with exactly one `<think>...</think><reflection>...</reflection><answer>...</answer>` sequence and nothing else.

In `<reflection>`, briefly audit the reasoning and begin with `Correct` if reliable.

In `<answer>`, write the final numerical answer after `####`. If the problem lacks enough information or you cannot solve it confidently, write exactly `I don't know` inside `<answer>`.

M.3 REIN EVALUATION PROMPT

You are a careful reasoner. For every problem, reply with EXACTLY this structure:

```
<think>...</think>
<reflection>...</reflection>
<answer>...</answer>.
```

M.4 REFLECTION-ONLY PROMPT

You are a helpful AI Assistant. Respond in the following format:

```
<think>
...
</think>
<reflection>
...
</reflection>
<answer>
...
</answer>
```