

# SCOPE: Stepwise Calibration of Reasoning Confidence with Outcome-Grounded Prefix Evidence

Huaqin Zhao<sup>1</sup> Yifan Zhou<sup>1</sup> Bolun Sun<sup>2</sup> Chao Cao<sup>3</sup> Yiwei Li<sup>1</sup>  
Zhaojun Ding<sup>1</sup> Lu Zhang<sup>4</sup> Tianming Liu<sup>1</sup> Jin Lu<sup>1</sup>

<sup>1</sup>University of Georgia

<sup>2</sup>Northwestern University

<sup>3</sup>The University of Texas at Arlington

<sup>4</sup>Indiana University Indianapolis

## Abstract

Reinforcement learning with verifiable rewards (RLVR) improves language-model reasoning using outcome-level supervision, but this supervision is temporally coarse: it indicates whether a trajectory ends with a correct answer, without revealing when the reasoning becomes reliable or silently fails. Existing verbalized-confidence methods typically calibrate a single confidence score after the final answer, leaving intermediate uncertainty unobserved. We propose SCOPE (Stepwise Calibration with Outcome-grounded Prefix Evidence), which trains reasoning models to emit confidence after each reasoning step. SCOPE interprets step confidence as the model’s prefix-conditioned belief in the gold answer: how likely the gold answer is under the reasoning prefix generated so far. During training, SCOPE constructs a soft target by scoring the gold answer conditioned on each reasoning prefix under the current policy. It then distills this signal with a two-stage masked RL objective: the first stage stabilizes the output format and final-answer accuracy, while the second stage performs calibration-only masked updates by applying stepwise and final-confidence rewards only to the numerical tokens inside the emitted confidence spans. Experiments on mathematical reasoning benchmarks show that SCOPE improves stepwise and final-confidence calibration while preserving final-answer reasoning performance. We further show that the resulting confidence signals support downstream inference-time control, enabling more reliable Learn-Then-Test risk control and stronger confidence-weighted aggregation.

## 1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a central paradigm for improving the reasoning ability of large language models (LLMs). By optimizing outcome-level signals such as final-answer correctness, GRPO-style training

can substantially improve mathematical and logical reasoning performance (Guo et al., 2025; OpenAI, 2024; Shao et al., 2024). However, outcome-level supervision is temporally coarse: it provides only a terminal correctness signal, without indicating when a reasoning prefix becomes reliable, when it begins to support the correct answer, or when an intermediate error causes the trajectory to deviate from a valid solution path. A model can therefore produce a fluent multi-step derivation, make a damaging intermediate mistake, and still report high confidence at the end.

Recent work on verbalized confidence calibration trains models to output a scalar confidence after the final answer (Damani et al., 2025; Bani-Harouni et al., 2025; Li et al., 2026; Ma et al., 2026). These methods establish a response-level calibration paradigm: the model solves the problem, emits one final confidence score, and that score is trained to match final correctness. But reasoning is not a single-shot prediction. A long chain of thought may contain useful derivations, irrelevant detours, brittle assumptions, and self-corrections before the final answer is produced. A single post-answer confidence score cannot reveal where uncertainty arises inside the reasoning process.

This motivates *stepwise reasoning confidence*: the model should expose how its confidence evolves as the chain of thought unfolds. This raises a central modeling question: what should stepwise confidence represent? A natural answer is local step correctness, but local correctness is not always the right target for test-time control. A step can be mathematically valid but irrelevant to the final solution, or locally incomplete while still moving the model toward the correct answer. We instead interpret step confidence as the model’s *prefix-conditioned belief in the gold answer*: the likelihood assigned to the gold answer given the reasoning prefix generated so far.

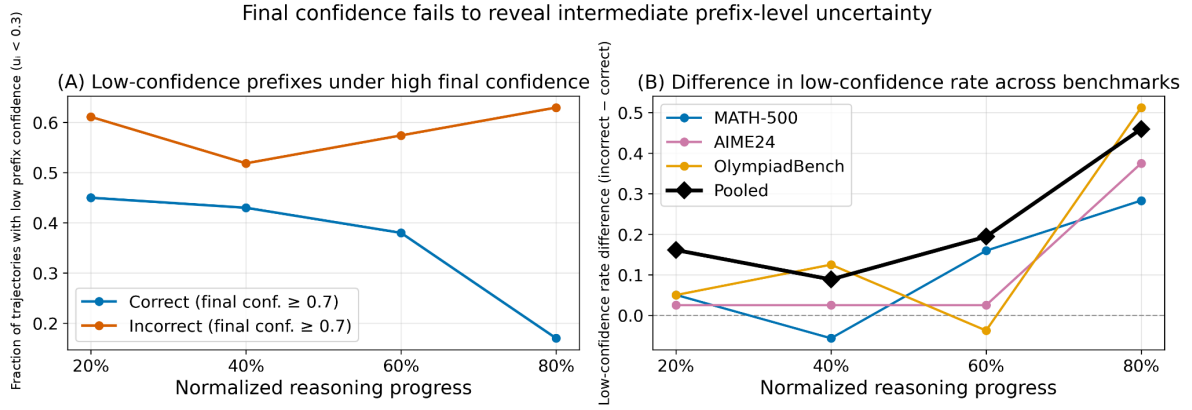


Figure 1: Motivation analysis on high-final-confidence trajectories from an RLCR-style calibrated model. We condition on final confidence  $\geq 0.7$  and track prefix confidence  $u_i$ , the model’s likelihood of the gold answer given the current reasoning prefix. Incorrect trajectories are more likely than correct trajectories to enter low-prefix-confidence states ( $u_i < 0.3$ ), and this gap increases toward later reasoning steps. This indicates that final confidence can mask intermediate reliability failures.

We propose SCOPE, a framework for training reasoning models to emit inline confidence after each reasoning step. For each reasoning prefix  $h_i = (x, z_{\leq i})$ , SCOPE constructs a soft supervision target by scoring the gold answer conditioned on that prefix under the current policy. This outcome-grounded prefix evidence captures whether the reasoning accumulated so far supports the correct final answer, rather than whether the latest step is locally correct.

To distill this signal without disrupting reasoning, SCOPE uses a two-stage masked RL recipe. Stage 1 stabilizes the required stepwise format and final-answer accuracy. Stage 2 starts from this policy and performs calibration-only masked updates on the numerical tokens inside the stepwise and final-confidence spans. The resulting model produces a confidence trace during ordinary decoding, making the signal directly usable for confidence-based test-time control.

Our contributions are threefold:

- To the best of our knowledge, we introduce the first framework to train the language model itself to emit calibrated stepwise verbalized confidence using outcome-grounded prefix evidence, with region-masked RL to avoid disturbing reasoning tokens.
- We define stepwise verbalized confidence as the model’s prefix-conditioned belief in the gold answer and instantiate this target with outcome-grounded prefix evidence, computed from gold-answer likelihood under the current policy. We train this signal using a two-stage

masked GRPO objective that separates format-and-accuracy learning from calibration-only updates on confidence-number tokens.

- Experiments on mathematical reasoning benchmarks show that SCOPE improves stepwise and final-confidence calibration while preserving final-answer accuracy. We further show that the learned confidence traces support test-time scaling through Learn-Then-Test risk control and confidence-weighted aggregation.

## 2 Method

SCOPE trains a reasoning policy to emit a numerical confidence after each reasoning step. The confidence is supervised by outcome-grounded prefix evidence, a soft target for the model’s prefix-conditioned belief in the gold answer. Training follows a two-stage masked RL procedure: Stage 1 stabilizes the stepwise format and final-answer accuracy, and Stage 2 applies calibration-only updates to the emitted confidence numbers.

### 2.1 Problem Setup

Given a question  $x$ , we write a generated trajectory as

$$\tau = (z_1, v_1, \dots, z_K, v_K, y, v_F),$$

where  $z_i$  is the  $i$ -th reasoning step,  $v_i$  is the confidence statement emitted after that step,  $y$  is the final answer, and  $v_F$  is the final-answer confidence statement. Let  $y^*$  denote the gold answer.

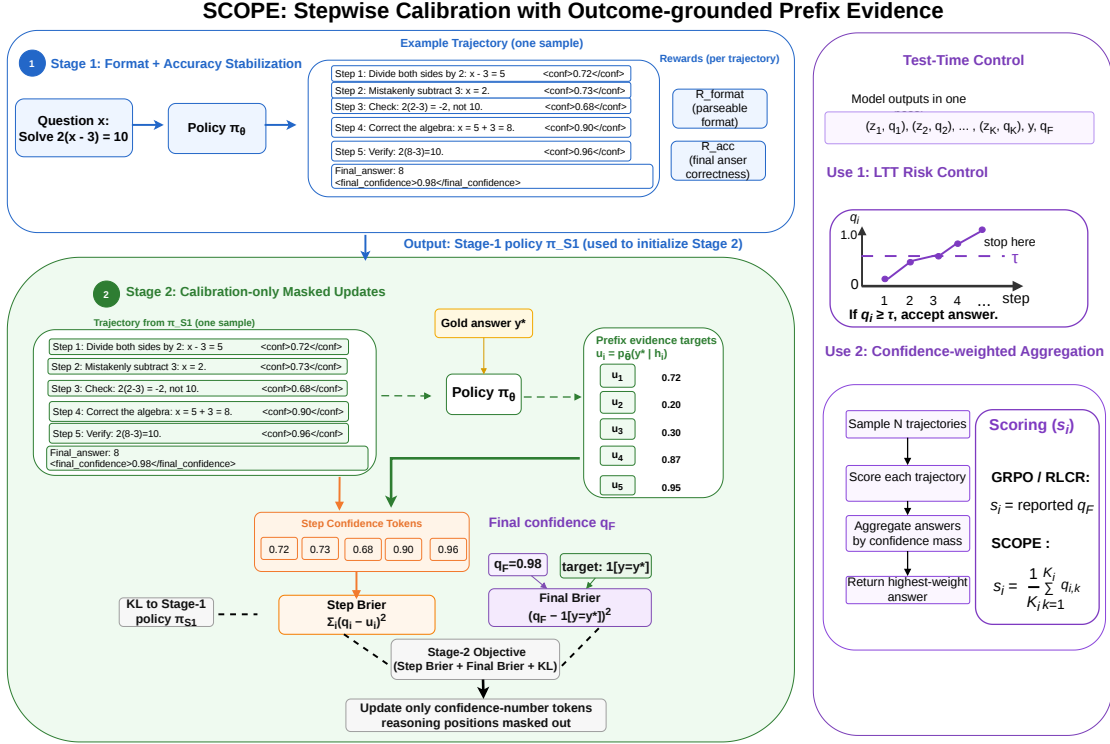


Figure 2: Overview of SCOPE. Stage 1 stabilizes the stepwise reasoning format and final-answer accuracy. Stage 2 computes outcome-grounded prefix evidence by scoring the gold answer under a stop-gradient copy of the current policy, and then routes reasoning rewards to reasoning tokens and calibration rewards to the corresponding confidence-number spans. At inference time, the emitted confidence trace can be used for confidence-based test-time selection and aggregation.

We define the autoregressive prefix before the  $i$ -th confidence statement as

$$h_i = (x, z_1, v_1, \dots, z_{i-1}, v_{i-1}, z_i).$$

This prefix matches the context available to the policy when it emits  $v_i$ : it includes the current reasoning step and all previous reasoning and confidence statements, but not the current confidence value itself.

After each reasoning step, the policy emits a <conf> span containing  $v_i$ . At the end of the trajectory, the policy emits a <final\_answer> span containing  $y$  and a <final\_confidence> span containing  $v_F$ . Let  $\hat{q}_i \in [0, 1]$  and  $\hat{q}_F \in [0, 1]$  denote the numerical confidence values parsed from  $v_i$  and  $v_F$ , respectively. We train  $\hat{q}_i$  to estimate the model’s prefix-conditioned belief in the gold answer, rather than local step correctness.

## 2.2 Outcome-Grounded Prefix Evidence

For each autoregressive prefix  $h_i$ , SCOPE constructs a soft confidence target by scoring the gold

answer under the current policy in a detached scoring pass. Let  $\pi_\theta$  be the policy being trained, and let  $\pi_{\bar{\theta}}$  denote the same policy used for scoring with gradients stopped. This scoring pass uses the latest policy weights but does not introduce a separate external model.

We compute the length-normalized log-likelihood of the gold answer conditioned on  $h_i$ :

$$\ell_i = \frac{1}{|y^*|} \sum_{m=1}^{|y^*|} \log \pi_{\bar{\theta}}(y_m^* | h_i, y_{<m}^*), \quad (1)$$

where  $|y^*|$  denotes the number of tokens in the gold answer. The prefix-evidence target is

$$u_i = \exp(\ell_i). \quad (2)$$

Thus,  $u_i \in [0, 1]$  serves as an operational measure of the model’s prefix-conditioned belief in the gold answer, implemented as the length-normalized likelihood of  $y^*$  under the same autoregressive context used to emit the step confidence.

### 2.3 Confidence Rewards

For the confidence value emitted after step  $i$ , we train the parsed confidence  $\hat{q}_i$  to match the prefix-evidence target  $u_i$ . For a trajectory with  $K$  parsed step-confidence values, we define the stepwise calibration reward as

$$R_{\text{step}} = 1 - \frac{1}{K} \sum_{i=1}^K (\hat{q}_i - u_i)^2. \quad (3)$$

Averaging over steps makes trajectories with different numbers of reasoning steps comparable.

For the final confidence value, we use an answer-level Brier reward against final-answer correctness:

$$R_{\text{brier}} = 1 - (\hat{q}_F - \mathbf{1}[y \equiv y^*])^2. \quad (4)$$

Stepwise confidence is therefore supervised by the continuous prefix-evidence target  $u_i$ , whereas final confidence is supervised by binary final correctness. We also evaluate a log-scoring-rule variant as an ablation; all main SCOPE results use the L2/Brier rewards in Eq. 3 and Eq. 4.

### 2.4 Two-Stage Masked GRPO Training

A single completion-level reward creates ambiguous credit assignment because reasoning accuracy, format compliance, and confidence calibration affect different parts of the generated sequence. If calibration rewards are propagated through the entire trajectory, they may perturb reasoning tokens. Conversely, if confidence spans are optimized together with long reasoning spans, the short numerical confidence tokens may receive diluted gradients. SCOPE therefore separates reasoning stabilization from confidence calibration.

**Stage 1: format-and-accuracy stabilization.** Stage 1 trains the model to produce the required stepwise format while preserving final-answer reasoning ability. For each prompt  $x$ , we sample a group of  $G$  trajectories

$$\tau^{(1)}, \dots, \tau^{(G)} \sim \pi_\theta(\cdot | x).$$

Let  $\text{Fmt}(\tau)$  denote whether trajectory  $\tau$  satisfies the required stepwise format. For trajectory  $\tau^{(g)}$ , we define

$$\begin{aligned} f^{(g)} &= \mathbf{1}[\text{Fmt}(\tau^{(g)}) = 1], \\ c^{(g)} &= \mathbf{1}[y^{(g)} = y^*], \end{aligned}$$

where  $f^{(g)}$  indicates format compliance and  $c^{(g)}$  indicates final-answer correctness. The Stage-1

reward is

$$R^{S1}(\tau^{(g)}) = \lambda_{\text{fmt}} f^{(g)} + \lambda_{\text{acc}} c^{(g)}. \quad (5)$$

In our main experiments, we set  $\lambda_{\text{fmt}} = 0.5$  and  $\lambda_{\text{acc}} = 1.0$ . Stage 1 provides a stable format-and-accuracy initialization for the calibration stage, ensuring parseable stepwise outputs while preserving final-answer reasoning ability.

#### Stage 2: calibration-only masked updates.

Stage 2 starts from the Stage-1 policy and calibrates the emitted confidence numbers. The model still generates complete reasoning trajectories, but reasoning tokens are masked out from the Stage-2 policy-gradient loss. Only numerical tokens inside the per-step `<conf>` spans and the final `<final_confidence>` span receive direct calibration gradients.

For each calibration reward component  $R_c \in \{R_{\text{step}}, R_{\text{brier}}\}$ , we compute a group-normalized GRPO advantage over the  $G$  trajectories:

$$A_c^{(g)} = \frac{R_c^{(g)} - \text{mean}_j(R_c^{(j)})}{\text{std}_j(R_c^{(j)}) + \epsilon}. \quad (6)$$

Let  $M_{\text{step}}$  denote the binary token mask selecting only the numerical tokens inside per-step `<conf>` spans, and let  $M_{\text{final}}$  select only the numerical token inside the final `<final_confidence>` span. For compactness, write

$$\mathcal{G}(R, M) = \mathcal{L}_{\text{mGRPO}}(R, M),$$

where  $\mathcal{L}_{\text{mGRPO}}(R, M)$  denotes the masked GRPO loss using the group-normalized advantage from reward  $R$  and normalizing over the active tokens selected by mask  $M$ .

The Stage-2 objective is

$$\begin{aligned} \mathcal{L}_{S2} &= \lambda_1 \mathcal{G}(R_{\text{step}}, M_{\text{step}}) \\ &\quad + \lambda_2 \mathcal{G}(R_{\text{brier}}, M_{\text{final}}) \\ &\quad + \beta_{\text{KL}} \text{KL}(\pi_\theta \| \pi_{S1}), \end{aligned} \quad (7)$$

where  $\pi_{S1}$  is the frozen Stage-1 policy. The coefficients  $\lambda_1$  and  $\lambda_2$  control the weights of stepwise and final-confidence calibration, respectively. The KL term keeps the calibrated policy close to the format-and-accuracy policy learned in Stage 1.

This region-specific masking is central to SCOPE: it prevents calibration rewards from directly updating reasoning tokens while avoiding gradient dilution on the short confidence spans.

## 2.5 Inference

At inference time, SCOPE generates the full trajectory in a single autoregressive decode,

$$\tau = ((z_1, v_1), \dots, (z_K, v_K), y, v_F).$$

The emitted confidence values can be used directly at test time without access to the gold answer. We parse the confidence trace  $(\hat{q}_1, \dots, \hat{q}_K)$  and use it for stepwise calibration evaluation and confidence-based test-time control.

## 3 Experiments

### 3.1 Experimental Setup

**Models and training.** We evaluate SCOPE on Qwen2.5-Math-1.5B-Base and Qwen2.5-7B-Base. Both model scales are trained with the two-stage procedure described in Section 2.4. Stage 1 optimizes the stepwise output format and final-answer correctness; Stage 2 starts from the Stage-1 checkpoint and applies masked calibration rewards to the confidence-number spans. The prefix-evidence target is computed by scoring the gold answer under the current policy in a detached scoring pass, without using human step-level labels or an external process reward model.

**Datasets.** For math RL training, we use 15,000 problems from the Big-Math-Digits training data. For evaluation, we report results on GSM8K, MATH-500, AMC23, AIME24, and Olympiad-Bench. We additionally include Big-Math-Digits in the test-time aggregation experiments to match the downstream scaling setting. For QA generalization at 7B scale, models are trained on HotPotQA-Modified and evaluated on HotPotQA, TriviaQA, SimpleQA, CommonsenseQA, and GPQA. Full training details and prompt templates are provided in Appendix B.

**Metrics.** For final-answer performance, we report accuracy. For final confidence, we report Brier score, expected calibration error (ECE), and AUROC against final-answer correctness. For stepwise confidence, we evaluate each emitted step confidence against Monte-Carlo prefix success: given a reasoning prefix, we sample  $M = 8$  continuations and compute the fraction that reaches the gold answer. Unless otherwise stated, all methods are evaluated under the same setting.

**Baselines.** We compare against representative baselines spanning prompting-only inference, outcome-only RL, and confidence-aware RL. **Base** denotes the original backbone prompted to emit an answer and confidence. **GRPO** is an outcome-only RL baseline trained with final-answer correctness rewards. **RLCR** is an answer-level verbalized-confidence calibration baseline that applies a Brier-style reward to final confidence. **C<sup>2</sup>GSPG** is a confidence-calibrated policy-gradient baseline based on sequence-level confidence regularization. We also report **SCOPE(Stage-I)** to isolate format-and-accuracy training from Stage-II confidence calibration.

### 3.2 Math Final-Answer Accuracy and Calibration

Table 1 reports macro-average final-answer accuracy and final-confidence calibration on mathematical reasoning benchmarks. The main trend is that Stage-II calibration substantially improves confidence quality without sacrificing final-answer accuracy. At 1.5B scale, SCOPE(Stage-II) achieves the best macro accuracy and improves AUROC from 0.465 for Stage-I to 0.768. This indicates that learning the stepwise format alone is insufficient: Stage-I can emit confidence values, but these values remain poorly calibrated.

At 7B scale, SCOPE(Stage-II) achieves the strongest macro accuracy (0.489) and AUROC (0.792), while matching the best Brier score among calibration-aware baselines. RLCR and C<sup>2</sup>GSPG remain competitive on some calibration metrics, but they do not provide the same overall accuracy-discrimination trade-off. These results support the claim that outcome-grounded prefix evidence can be distilled into useful confidence expressions while preserving the reasoning ability learned through RL. Full per-dataset math results are provided in Appendix C.1.

### 3.3 QA Generalization

Table 2 reports macro-average QA generalization results at 7B scale. Although SCOPE is designed around stepwise confidence rather than task-specific QA calibration, Stage-II achieves the best macro Brier, ECE, and AUROC while maintaining competitive answer accuracy. This suggests that masked confidence calibration transfers beyond mathematical reasoning to broader question answering settings. Full per-dataset QA results are provided in Appendix C.2.

| Model | Method              | Acc.↑        | Brier↓       | ECE↓         | AUROC↑       |
|-------|---------------------|--------------|--------------|--------------|--------------|
| 1.5B  | Base                | 0.221        | 0.674        | 0.697        | 0.565        |
|       | GRPO                | 0.414        | 0.602        | 0.635        | 0.518        |
|       | C <sup>2</sup> GSPG | 0.416        | 0.243        | 0.249        | 0.655        |
|       | RLCR                | 0.425        | 0.214        | 0.243        | 0.691        |
|       | SCOPE(Stage-I)      | 0.429        | 0.461        | 0.458        | 0.465        |
|       | SCOPE(Stage-II)     | <b>0.437</b> | <b>0.170</b> | <b>0.176</b> | <b>0.768</b> |
| 7B    | Base                | 0.249        | 0.633        | 0.632        | 0.492        |
|       | GRPO                | 0.460        | 0.566        | 0.562        | 0.493        |
|       | C <sup>2</sup> GSPG | 0.472        | 0.216        | 0.207        | 0.769        |
|       | RLCR                | 0.472        | 0.170        | <b>0.163</b> | 0.764        |
|       | SCOPE(Stage-I)      | 0.474        | 0.313        | 0.317        | 0.421        |
|       | SCOPE(Stage-II)     | <b>0.489</b> | <b>0.169</b> | 0.176        | <b>0.792</b> |

Table 1: Macro-average final-answer accuracy and final-confidence calibration on math reasoning benchmarks. Macro averages are computed over GSM8K, MATH-500, AMC23, AIME24, and OlympiadBench. For the 7B Base row, AUROC excludes AIME24 because all predictions are incorrect and AUROC is undefined. Full per-dataset results are reported in Appendix C.1.

| Method              | Acc.↑        | Brier↓       | ECE↓         | AUROC↑       |
|---------------------|--------------|--------------|--------------|--------------|
| Base                | 0.479        | 0.456        | 0.446        | 0.533        |
| GRPO                | 0.536        | 0.464        | 0.464        | 0.501        |
| C <sup>2</sup> GSPG | 0.535        | 0.195        | 0.175        | 0.642        |
| RLCR                | 0.536        | 0.217        | 0.176        | 0.659        |
| SCOPE(Stage-I)      | 0.535        | 0.465        | 0.464        | 0.524        |
| SCOPE(Stage-II)     | <b>0.538</b> | <b>0.177</b> | <b>0.130</b> | <b>0.683</b> |

Table 2: Macro-average QA generalization results for 7B models. Macro averages are computed over HotPotQA, TriviaQA, SimpleQA, CommonsenseQA, and GPQA. Following RLCR, QA-trained models are trained on HotPotQA-Modified and evaluated on both in-domain and out-of-domain QA benchmarks. Full per-dataset results are reported in Appendix C.2.

### 3.4 Stepwise Confidence Calibration

Final confidence alone cannot reveal where reasoning becomes unreliable. We therefore evaluate whether emitted step confidences align with prefix-level solvability. For each generated prefix, we estimate Monte-Carlo prefix success by sampling  $M = 8$  continuations and computing the fraction that reaches the gold answer.

Table 3 studies which target source is most suitable for supervising verbalized step confidence. We compare policy-derived prefix evidence with four simple uncertainty proxies: MSP, which uses the maximum token probability; CCP, which uses claim-conditioned probability; PPL, which uses sequence perplexity; and Entropy, which uses the predictive entropy of the model distribution. All proxy scores are normalized so that larger values indicate higher confidence. The policy-derived prefix evidence used by SCOPE achieves the best Brier

| Source       | Brier↓       | ECE↓         | AUROC↑       | $\rho_s$ ↑   |
|--------------|--------------|--------------|--------------|--------------|
| Policy $u_i$ | <b>0.206</b> | 0.168        | <b>0.717</b> | <b>0.388</b> |
| MSP          | 0.218        | <b>0.107</b> | 0.620        | 0.204        |
| CCP          | 0.262        | 0.210        | 0.675        | 0.324        |
| PPL          | 0.274        | 0.242        | 0.642        | 0.286        |
| Entropy      | 0.327        | 0.379        | 0.488        | 0.314        |

Table 3: Target-source ablation for stepwise verbalized confidence. All metrics compare emitted step confidence with Monte-Carlo prefix success.  $\rho_s$  denotes Spearman correlation. Policy  $u_i$  corresponds to SCOPE’s outcome-grounded prefix-evidence target.

| Variant           | Acc.↑        | Brier↓       | ECE↓         | AUROC↑       |
|-------------------|--------------|--------------|--------------|--------------|
| SCOPE             | <b>0.437</b> | <b>0.170</b> | <b>0.176</b> | <b>0.768</b> |
| Stage-I only      | 0.429        | 0.461        | 0.458        | 0.465        |
| w/o region mask   | 0.425        | 0.252        | 0.271        | 0.725        |
| w/o KL-to-Stage-I | 0.420        | 0.203        | 0.224        | 0.735        |

Table 4: Key design ablations of SCOPE. We report macro averages over the math ablation suite.

score, AUROC, and Spearman correlation, indicating the strongest overall alignment with Monte-Carlo prefix success. Although MSP obtains the lowest ECE, its lower AUROC and Spearman correlation suggest that it is weaker at ranking reliable and unreliable prefixes. These results support using outcome-grounded prefix evidence as the supervision target for stepwise confidence.

### 3.5 Ablation Study

Table 4 isolates the main components of SCOPE. Stage-I only reaches reasonable accuracy but yields poor confidence calibration, confirming that format-and-accuracy training alone does not make stepwise or final confidence reliable. Removing region masking hurts both accuracy and calibration, supporting the need to route calibration rewards to the confidence-number spans rather than propagating them through the entire reasoning trace. Removing the KL anchor to Stage-I also degrades performance, indicating that the calibration stage benefits from staying close to the format-and-accuracy policy. Overall, the full model gives the best accuracy-calibration trade-off.

**Summary.** Across math, QA, stepwise calibration, and ablation evaluations, SCOPE consistently improves confidence reliability while preserving final-answer performance. Stage-II improves final-confidence discrimination over Stage-I, policy-derived prefix evidence best aligns emitted

step confidence with Monte-Carlo prefix success, and the ablations show that both confidence-region masking and the Stage-I anchor are important for stable calibration.

## 4 Downstream Applications

We evaluate whether SCOPE’s confidence traces support two downstream inference-time uses without access to gold answers: Learn-Then-Test risk control and confidence-weighted aggregation.

### 4.1 Learn-Then-Test Risk Control

We evaluate whether the confidence signals produced by SCOPE can support selective prediction under a Learn-Then-Test (LTT) protocol. Given a target accuracy level  $1 - \epsilon$ , LTT selects a confidence threshold on a calibration split and applies the selected threshold to a held-out test split. A useful confidence signal should yield realized test accuracy close to the target accuracy: falling below the diagonal indicates insufficient risk control, while staying far above the diagonal indicates overly conservative thresholding.

Figure 3 reports realized test accuracy as the target accuracy varies. All methods generally satisfy the target accuracy, but their behavior differs in how tightly they track the ideal calibration line. SCOPE follows the diagonal more closely across most target levels, indicating that its stepwise confidence trace provides a more precise signal for threshold selection. In contrast, GRPO and RLCR tend to lie noticeably above the diagonal, suggesting more conservative thresholds that achieve higher-than-requested accuracy but provide less fine-grained control over the target risk level. These results show that calibrated stepwise confidence can serve as a useful reliability signal for LTT-based test-time risk control.

### 4.2 Confidence-Weighted Aggregation

We further evaluate whether the emitted confidence trace improves test-time scaling over multiple sampled trajectories. Given  $N$  rollouts with predicted answers  $y_1, \dots, y_N$ , majority voting selects the most frequent answer:

$$\hat{y}_{\text{maj}} = \arg \max_y \sum_{i=1}^N \mathbf{1}[y_i = y].$$

Confidence-weighted aggregation instead assigns each rollout a reliability score  $s_i$  and selects the

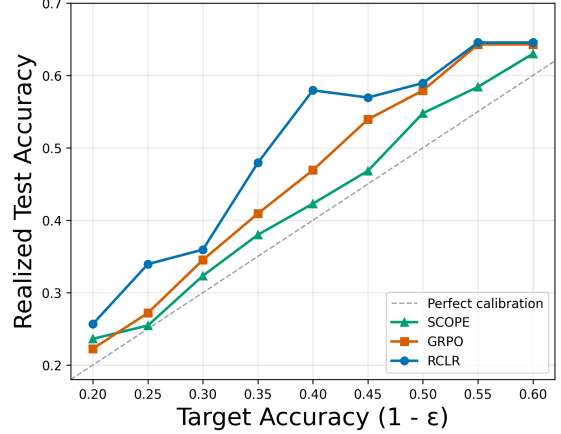


Figure 3: Learn-Then-Test risk control over five math benchmarks. SCOPE matches realized accuracy to the target accuracy more closely than GRPO and RLCR, indicating more precise confidence-based threshold selection.

answer with the largest total confidence mass:

$$\hat{y}_{\text{conf}} = \arg \max_y \sum_{i: y_i = y} s_i.$$

For GRPO and RLCR,  $s_i$  is the reported final confidence. For SCOPE, we use the mean of the emitted stepwise confidence values as the rollout-level reliability score:

$$s_i = \frac{1}{K_i} \sum_{k=1}^{K_i} \hat{q}_{i,k},$$

where  $K_i$  is the number of reasoning steps in rollout  $i$ .

Table 5 reports macro-average accuracy across six math benchmarks under the same sampling budget. SCOPE achieves the best confidence-weighted aggregation result, improving over both GRPO and RLCR. Moreover, confidence-weighted aggregation improves over majority voting for SCOPE, suggesting that the emitted stepwise confidence trace provides reliability information beyond answer frequency.

## 5 Related Work

**Verbalized confidence calibration.** A growing body of work studies how to train language models to express calibrated uncertainty. RLCR (Damani et al., 2025) augments outcome rewards with a Brier-style reward on verbalized confidence against final-answer correctness. C<sup>2</sup>GSPG (Liu et al., 2025a) incorporates a confidence-calibration regularizer into group-based policy optimization to

| Method | Pass@1 $\uparrow$ | Majority $\uparrow$ | Conf.-weighted $\uparrow$ |
|--------|-------------------|---------------------|---------------------------|
| GRPO   | 0.517             | 0.567               | 0.564                     |
| RLCR   | 0.500             | 0.566               | 0.568                     |
| SCOPE  | 0.516             | <b>0.569</b>        | <b>0.573</b>              |

Table 5: Average accuracy across six math benchmarks under different aggregation strategies. SCOPE improves confidence-weighted aggregation over GRPO and RLCR.

better align normalized sequence-level confidence with answer correctness. Rewarding Doubt (Bani-Harouni et al., 2025) uses a log scoring rule to encourage calibrated confidence expression, Conf-Tuner (Li et al., 2026) studies confidence prediction at the token level, and DCPO (Ma et al., 2026) decouples reasoning and confidence gradients to mitigate the accuracy–calibration conflict. These methods establish a response-level calibration paradigm: the model produces an answer, emits a scalar confidence score, and receives confidence supervision from the final outcome label.

**Likelihood-based rewards and outcome-grounded reasoning signals.** Recent verifier-free RL methods use the probability or log-likelihood of a reference answer as a supervision signal. VeriFree (Zhou et al., 2025) and RLPR (Yu et al., 2025) replace external verifiers with reference-answer probability rewards, while NOVER (Liu et al., 2025b) uses the perplexity of the ground-truth answer conditioned on generated reasoning as a reward proxy. A systematic study of likelihood-based reward designs further compares probability- and log-probability-based rewards across verifiable and non-verifiable settings (Kwiatkowski et al., 2026). Closest to our setting, PACR (Yoon et al., 2025) tracks the prefix-conditioned probability of the ground-truth answer and uses its stepwise change as a dense process reward. Together, these works show that reference-answer likelihood provides useful outcome-grounded information about reasoning trajectories.

**Process supervision and intermediate uncertainty.** A separate line of work provides feedback at intermediate reasoning states using process reward models, rollout-based value estimates, or hidden-state probes. Process reward models train auxiliary verifiers to score step quality (Lightman et al., 2024; Wang et al., 2024; Zhang et al., 2025; Yuan et al., 2024), while rollout-based meth-

ods estimate prefix or step value through continuation sampling (Setlur et al., 2025). Other approaches derive uncertainty signals from the policy or its internal states, including entropy, trace length, self-evaluation, and hidden-state probes (Fu et al., 2025; Devic et al., 2025; Kadavath et al., 2022; Ballon et al., 2026). RLCM (Wang et al., 2026) is particularly relevant: it supervises process-level confidence with a probe-based margin objective and shows benefits for risk control and confidence-weighted aggregation.

**Positioning of our method.** SCOPE connects verbalized confidence calibration, likelihood-based reasoning signals, and process-level uncertainty estimation. Unlike response-level calibration methods such as RLCR, SCOPE trains the model to expose confidence throughout the reasoning process rather than only after the final answer. Unlike likelihood-based RL methods such as PACR, SCOPE does not use prefix-conditioned gold-answer likelihood merely as a reward or scoring function; it turns this outcome-grounded signal into a soft target for the model’s own stepwise verbalized confidence. Unlike probe-based process-confidence methods such as RLCM, SCOPE does not attach an auxiliary confidence head or rely on hidden-state confidence at test time. Instead, the language model itself emits a calibrated confidence value after each reasoning step, producing a process-level confidence trace that can be used directly for downstream risk control and confidence-weighted aggregation.

## 6 Conclusion

We introduced SCOPE, a framework for training reasoning models to verbalize calibrated confidence after each reasoning step. Rather than treating step confidence as local step correctness or calibrating only a single post-answer score, SCOPE interprets stepwise confidence as the model’s prefix-conditioned belief in the gold answer. We instantiate this target with outcome-grounded prefix evidence, computed from the likelihood of the gold answer conditioned on each reasoning prefix under the current policy. To learn this signal, SCOPE uses a two-stage masked RL objective: Stage 1 stabilizes the stepwise output format and final-answer accuracy, while Stage 2 applies calibration-focused updates to the numerical confidence tokens. Experiments show that SCOPE improves stepwise and final-confidence calibration while preserving

final-answer reasoning performance. The resulting confidence traces further support confidence-based inference-time control, including Learn-Then-Test risk control and confidence-weighted aggregation.

## Limitations

Due to limited computational resources, our experiments are conducted on 1.5B and 7B models, and we have not yet evaluated SCOPE on larger backbones. In addition, SCOPE relies on verifiable final answers to construct outcome-grounded prefix-evidence targets during training, making the current formulation most directly applicable to mathematical reasoning and question answering tasks with reliable gold answers. Extending the method to open-ended generation, subjective evaluation settings, and broader test-time control scenarios remains an important direction for future work.

## References

- Marthe Ballon, Brecht Verbeken, Vincent Ginis, and Andres Algaba. 2026. [Probing the trajectories of reasoning traces in large language models](#). *arXiv preprint arXiv:2601.23163*.
- David Bani-Harouni, Chantal Pellegrini, Paul Stangel, Ege Özsoy, Kamilia Zaripova, Nassir Navab, and Matthias Keicher. 2025. [Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models](#). *arXiv preprint arXiv:2503.02623*.
- Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenefeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. 2025. [Beyond binary rewards: Training LMs to reason about their uncertainty](#). *arXiv preprint arXiv:2507.16806*.
- Siddhartha Devic, Charlotte Peale, Arwen Bradley, Sinead Williamson, Preetum Nakkiran, and Aravind Gollakota. 2025. [Trace length is a simple uncertainty signal in reasoning models](#). *arXiv preprint arXiv:2510.10409*.
- Yichao Fu, Xuwei Wang, Yuandong Tian, and Jiawei Zhao. 2025. [Deep think with confidence](#). *arXiv preprint arXiv:2508.15260*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. [Deepseek-r1 incentivizes reasoning in LLMs through reinforcement learning](#). *Nature*, 645(8081):633–638. *ArXiv:2501.12948*.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. [Language models \(mostly\) know what they know](#). *arXiv preprint arXiv:2207.05221*.
- Ariel Kwiatkowski, Natasha Butt, Ismail Labiad, Julia Kempe, and Yann Ollivier. 2026. [Likelihood-based reward designs for general LLM reasoning](#). *arXiv preprint arXiv:2602.03979*.
- Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. 2026. Conftuner: Training large language models to express their confidence verbally. *Advances in Neural Information Processing Systems*, 38:53484–53513.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.
- Haotian Liu, Shuo Wang, and Hongteng Xu. 2025a. [C<sup>2</sup>gspg: Confidence-calibrated group sequence policy gradient towards self-aware reasoning](#). *Preprint, arXiv:2509.23129*.
- Wei Liu, Siya Qi, Xinyu Wang, Chen Qian, Yali Du, and Yulan He. 2025b. [NOVER: Incentive training for language models via verifier-free reinforcement learning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7439–7458, Suzhou, China. Association for Computational Linguistics. *ArXiv:2505.16022*.
- Zhengzhao Ma, Xueru Wen, Boxi Cao, Yaojie Lu, Hongyu Lin, Jinglin Yang, Min He, Xianpei Han, and Le Sun. 2026. [Decoupling reasoning and confidence: Resurrecting calibration in reinforcement learning from verifiable rewards](#). *arXiv preprint arXiv:2603.09117*.
- OpenAI. 2024. [Learning to reason with LLMs](#). OpenAI blog. Accessed 2026-05-25.
- Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2025. [Rewarding progress: Scaling automated process verifiers for LLM reasoning](#). In *International Conference on Learning Representations*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- Liaoyaqi Wang, Chunsheng Zuo, William Jurayj, Benjamin Van Durme, and Anqi Liu. 2026. Process supervision of confidence margin for calibrated llm reasoning. *arXiv preprint arXiv:2604.23333*.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Eunseop Yoon, Hee Suk Yoon, Jaehyun Jang, SooHwan Eom, Qi Dai, Chong Luo, Mark A. Hasegawa-Johnson, and Chang D. Yoo. 2025. **PACR: Progressively ascending confidence reward for LLM reasoning.** *arXiv preprint arXiv:2510.22255*.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. 2025. **RLPR: Extrapolating RLVR to general domains without verifiers.** *arXiv preprint arXiv:2506.18254*.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. 2024. **Free process rewards without process labels.** *arXiv preprint arXiv:2412.01981*.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. **The lessons of developing process reward models in mathematical reasoning.** In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516, Vienna, Austria. Association for Computational Linguistics.

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. 2025. **Reinforcing general reasoning without verifiers.** *arXiv preprint arXiv:2505.21493*.

## A Confidence Distribution Analysis

Figure 4 compares the distribution of confidence values produced by SCOPE and RLCR. For SCOPE, we aggregate the emitted step-level confidence values across reasoning traces. For RLCR, we use the reported final-answer confidence. The two distributions exhibit different behaviors: RLCR final confidence is concentrated in the high-confidence region, with mean 0.71 and median 0.80, whereas SCOPE produces a broader step-confidence distribution with mean 0.49 and median 0.41. This suggests that final-answer confidence tends to be coarse and high-valued, while stepwise confidence exposes more fine-grained uncertainty throughout the reasoning process.

## B Training and Evaluation Details

### B.1 RL Training Configuration

Table 6 summarizes the training configuration used for the main two-stage RL SCOPE runs.

### B.2 Prompt Templates

Table 7 summarizes the math and QA prompt formats. We provide compact examples below the table; the qualitative examples in Appendix E show full model outputs.

#### Math prompt example.

```
Solve the math problem step by step. After each step, assign a
confidence value
in [0.00, 1.00] estimating how likely the current trajectory is to
lead to the
correct final answer.

FORMAT:
<reasoning>...</reasoning>, <conf>0.XX</conf>
...
<final_answer>...</final_answer>, <final_confidence>0.XX</
final_confidence>

EXAMPLE:
Question: A train travels at 60 mph for 2.5 hours. How far does it
go?

<reasoning>Use distance = speed x time.</reasoning>, <conf>0.70</
conf>
<reasoning>Compute 60 x 2.5 = 150.</reasoning>, <conf>0.90</conf>
<final_answer>150</final_answer>, <final_confidence>0.96</
final_confidence>
```

#### QA prompt example.

```
Answer the question step by step. After each step, assign a
confidence value
in [0.00, 1.00] estimating how likely the current trajectory is to
lead to the
correct final answer.

FORMAT:
<reasoning>...</reasoning>, <conf>0.XX</conf>
...
<final_answer>...</final_answer>, <final_confidence>0.XX</
final_confidence>

EXAMPLE:
Question: Which film was released first, Inception or The Dark
Knight?

<reasoning>The Dark Knight was released in 2008.</reasoning>, <conf
>0.88</conf>
<reasoning>Inception was released in 2010, so The Dark Knight was
first.</reasoning>, <conf>0.96</conf>
<final_answer>The Dark Knight</final_answer>, <final_confidence
>0.95</final_confidence>
```

## C Full Results

This section reports the per-dataset results corresponding to the macro-average tables in the main text.

### C.1 Full Math Results

### C.2 Full QA Generalization Results

### C.3 Detailed Test-Time Aggregation

Table 10 reports confidence-weighted aggregation by dataset. The main paper reports the macro-average summary in Table 5.

## D Additional Analysis and Ablations

### D.1 Prefix-Evidence Diagnostic and Target Validation

Before training SCOPE, we diagnose whether final-confidence calibrated models already exhibit hidden prefix-level reliability failures and whether

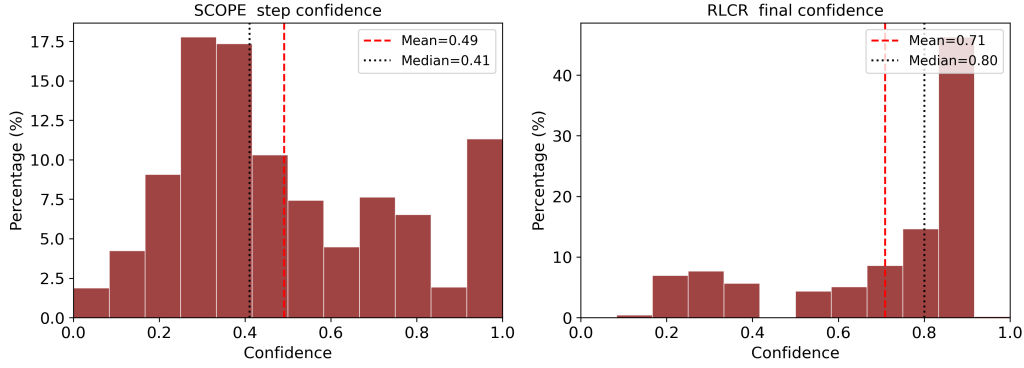


Figure 4: Confidence distribution comparison between SCOPE and RLCR. SCOPE reports step-level confidence throughout the reasoning process, while RLCR reports final-answer confidence. Compared with RLCR, SCOPE yields a broader and less over-concentrated confidence distribution, suggesting that stepwise confidence provides a more fine-grained uncertainty signal.

| Parameter                   | Math Stage 1                 | Math Stage 2                           | QA Stage 1          | QA Stage 2                             |
|-----------------------------|------------------------------|----------------------------------------|---------------------|----------------------------------------|
| Model scale                 | Qwen2.5-Math-1.5B / 7B       | Qwen2.5-Math-1.5B / 7B                 | Qwen2.5-7B          | Qwen2.5-7B                             |
| Initialization              | Base model                   | Stage-1 checkpoint                     | Base instruct model | Stage-1 checkpoint                     |
| Learning rate               | $5 \times 10^{-6}$           | $5 \times 10^{-6}$                     | $1 \times 10^{-6}$  | $1 \times 10^{-6}$                     |
| LR schedule                 | linear, warmup 0.10          | linear, warmup 0.10                    | linear, warmup 0.10 | linear, warmup 0.10                    |
| KL coefficient $\beta$      | 0.0                          | 0.01                                   | 0.0                 | 0.01                                   |
| Max steps                   | 200                          | 200                                    | 200                 | 200                                    |
| Epochs                      | 1.0                          | 1.0                                    | 1.0                 | 1.0                                    |
| Per-device batch size       | 8                            | 8                                      | 4                   | 4                                      |
| Gradient accumulation steps | 32                           | 32                                     | 32                  | 32                                     |
| Number of generations       | 8                            | 8                                      | 8                   | 8                                      |
| Rollout temperature         | 1.0                          | 1.0                                    | 1.0                 | 1.0                                    |
| Reward functions            | format, accuracy             | format, accuracy, step L2, final Brier | format, accuracy    | format, accuracy, step L2, final Brier |
| Reward weights              | [0.5, 1.0]                   | [0.5, 0.5]                             | [0.5, 1.0]          | [0.5, 0.5]                             |
| Reward token masking        | false                        | true                                   | false               | true                                   |
| Scale rewards               | true                         | true                                   | true                | true                                   |
| Hardware                    | 1.5B: 2×H100 NVL; 7B: 2×B200 | 1.5B: 2×H100 NVL; 7B: 2×B200           | 2×B200              | 2×B200                                 |

Table 6: RL training configurations for math and QA experiments. Stage 1 uses format-and-accuracy rewards, while Stage 2 starts from the corresponding Stage 1 checkpoint and adds stepwise and final-confidence calibration rewards.

prefix evidence  $u_i$  is a useful target. We use the public RLCR checkpoint and generate one trajectory per problem on 240 problems sampled from MATH-500 and AIME. We segment reasoning into steps via blank-line splitting, cap trajectories at 20 steps, and compute token entropy, policy log-probability, vanilla verbalized confidence, and prefix evidence  $u_i$  at each prefix. On a 100-problem subset, we additionally estimate  $p_i^{\text{MC}}$  with  $M = 8$  continuation rollouts, yielding 496 prefix-rollout pairs.

**Final-confidence blind spot.** Correct trajectories show a clear increase in prefix evidence from 0.29 to 0.86, whereas wrong trajectories remain near 0.13. At the trajectory level,  $u_i$  is highly non-monotonic: 55% of correct trajectories exhibit a dip-and-recover pattern, and 93% of wrong trajectories contain at least one prefix with  $u_i < 0.3$ . Final confidence also overshoots prefix evidence on errors: the median gap  $\text{final\_conf} - u_K$  is +0.40 for wrong trajectories and −0.20 for correct trajec-

tries, with 68% of wrong trajectories overshooting by more than 0.3 compared with 8% for correct trajectories.

**Target validity.** Table 11 compares candidate prefix-level signals against MC prefix success. Policy-derived prefix evidence aligns best with MC prefix success in terms of Spearman correlation, Pearson correlation, Brier score, and AUROC. Entropy and prefix log-probability are anti-correlated with prefix success, suggesting that distributional sharpness alone is not a reliable proxy for outcome-grounded prefix reliability.

## D.2 Target-Source Ablation Details

Table 3 in the main text reports the target-source ablation. Policy-derived prefix evidence obtains the best Brier score, AUROC, and Spearman correlation, while MSP achieves the lowest binning-based calibration error. This suggests that simple confidence proxies can be coarsely calibrated but are weaker at ranking reliable prefixes.

| Task | Prompt name          | Instruction and output format                                                                                                                                                                                                                                                                                                 |
|------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Math | tabc_step_fewshot    | Solve the math problem step by step. After each reasoning step, the model emits a numerical confidence in [0.00, 1.00] estimating how likely the correct final answer is under the reasoning prefix so far. The output consists of repeated <reasoning> and <conf> fields, followed by <final_answer> and <final_confidence>. |
| QA   | tabc_step_fewshot_qa | Answer the question step by step. After each reasoning step, the model emits a numerical confidence in [0.00, 1.00] estimating how likely the correct final answer is under the current reasoning prefix. The final answer must be a concise factual string, such as an entity, name, number, or yes/no answer.               |

Table 7: Prompt templates for math and QA stepwise-confidence generation. Compact prompt examples are shown below the table.

| Model | Method              | GSM8K        |              |              |              | MATH-500     |              |              |              | AMC23        |              |              |              | AIME24       |              |              |              | OlympiadBench |              |              |              |
|-------|---------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|--------------|--------------|--------------|
|       |                     | Acc↑         | Brier↓       | ECE↓         | AUROC↑       | Acc↑         | Brier↓       | ECE↓         | AUROC↑       | Acc↑         | Brier↓       | ECE↓         | AUROC↑       | Acc↑         | Brier↓       | ECE↓         | AUROC↑       | Acc↑          | Brier↓       | ECE↓         | AUROC↑       |
| 1.5B  | Base                | 0.457        | 0.424        | 0.441        | 0.751        | 0.250        | 0.667        | 0.679        | 0.499        | 0.325        | 0.688        | 0.713        | 0.566        | 0.011        | 0.790        | 0.823        | 0.520        | 0.062         | 0.802        | 0.831        | 0.489        |
|       | GRPO                | 0.796        | 0.340        | 0.403        | 0.523        | 0.581        | 0.549        | 0.602        | 0.514        | 0.384        | 0.551        | 0.560        | 0.487        | 0.067        | 0.744        | 0.817        | 0.529        | 0.244         | 0.825        | 0.791        | 0.536        |
|       | C <sup>2</sup> GSPG | 0.821        | 0.159        | 0.115        | 0.628        | 0.560        | 0.213        | 0.287        | 0.622        | 0.375        | 0.230        | 0.217        | 0.616        | 0.067        | 0.341        | 0.338        | 0.723        | 0.259         | 0.271        | 0.288        | 0.684        |
|       | RLCR                | 0.820        | 0.156        | 0.136        | 0.598        | 0.583        | 0.243        | 0.256        | 0.670        | 0.387        | 0.236        | 0.219        | 0.742        | 0.067        | 0.180        | 0.326        | 0.728        | 0.267         | 0.255        | 0.276        | 0.715        |
|       | SCOPE(Stage-I)      | 0.826        | 0.147        | 0.050        | 0.499        | 0.592        | 0.312        | 0.269        | 0.485        | 0.393        | 0.545        | 0.560        | 0.375        | 0.056        | 0.749        | 0.827        | 0.510        | 0.279         | 0.554        | 0.583        | 0.458        |
|       | SCOPE(Stage-II)     | <b>0.836</b> | 0.142        | 0.123        | <b>0.795</b> | 0.585        | 0.241        | 0.234        | <b>0.777</b> | <b>0.410</b> | <b>0.149</b> | <b>0.153</b> | <b>0.808</b> | <b>0.078</b> | <b>0.103</b> | <b>0.171</b> | <b>0.730</b> | 0.274         | <b>0.214</b> | <b>0.200</b> | <b>0.732</b> |
| 7B    | Base                | 0.735        | 0.240        | 0.215        | 0.524        | 0.468        | 0.485        | 0.492        | 0.591        | 0.024        | 0.667        | 0.667        | 0.313        | 0.000        | 1.000        | 1.000        | –            | 0.016         | 0.775        | 0.788        | 0.538        |
|       | GRPO                | 0.906        | 0.210        | 0.194        | 0.465        | 0.592        | 0.438        | 0.427        | 0.451        | 0.434        | 0.573        | 0.573        | 0.500        | 0.078        | 0.898        | 0.901        | 0.530        | 0.288         | 0.710        | 0.713        | 0.517        |
|       | C <sup>2</sup> GSPG | 0.906        | 0.089        | 0.102        | 0.679        | 0.621        | 0.233        | 0.167        | 0.687        | 0.461        | 0.204        | 0.230        | <b>0.801</b> | 0.108        | 0.323        | 0.291        | 0.877        | 0.265         | 0.233        | 0.245        | 0.800        |
|       | RLCR                | 0.896        | 0.100        | 0.119        | 0.632        | 0.598        | 0.227        | <b>0.120</b> | 0.665        | 0.470        | <b>0.197</b> | <b>0.094</b> | 0.780        | 0.111        | 0.140        | 0.266        | <b>0.910</b> | <b>0.283</b>  | <b>0.188</b> | <b>0.214</b> | 0.833        |
|       | SCOPE(Stage-I)      | 0.905        | 0.173        | 0.295        | 0.346        | 0.620        | 0.243        | 0.083        | 0.376        | 0.446        | 0.325        | 0.255        | 0.426        | 0.100        | 0.456        | 0.592        | 0.544        | <b>0.300</b>  | 0.366        | 0.362        | 0.414        |
|       | SCOPE(Stage-II)     | <b>0.909</b> | <b>0.064</b> | <b>0.058</b> | <b>0.730</b> | <b>0.640</b> | <b>0.213</b> | 0.203        | <b>0.715</b> | <b>0.480</b> | 0.216        | 0.220        | 0.780        | <b>0.121</b> | <b>0.095</b> | <b>0.149</b> | 0.893        | 0.294         | 0.257        | 0.252        | <b>0.841</b> |

Table 8: Full per-dataset final-answer accuracy and final-confidence calibration across math reasoning benchmarks. The macro-average summary is reported in Table 1. Acc measures final-answer correctness. Brier, ECE, and AUROC evaluate the quality of the reported final-answer confidence.

### D.3 Full Ablations and Hyperparameter Scans

## E Qualitative Examples

This section provides representative successful and failed model outputs. Prompt templates are reported in Appendix B.2.

### E.1 Successful Examples

**Example 1: Series identity.** This example is from MATH-500. The model correctly derives the identity and gradually increases confidence after the key reindexing and simplification steps.

Question:  
Define  
 $p = \sum_{k=1}^n \frac{1}{k^2}$  and  $q = \sum_{k=1}^n \frac{1}{k^3}$ .

Find a way to write  
 $\sum_{j=1}^n \sum_{k=1}^n \frac{1}{(j+k)^3}$   
in terms of  $p$  and  $q$ .

Gold:  $p - q$   
Prediction:  $p - q$   
Result: Correct

Model output:  
<reasoning>Recognize that the double sum can be rewritten by changing the index of summation. Let  $n = j + k$ . Then for each fixed  $n$ ,  $j$  ranges from 1 to  $n-1$  and  $k = n - j$ . Therefore, the double sum becomes  $\sum_{n=2}^{\infty} \sum_{j=1}^{n-1} \frac{1}{n^3}$ .</reasoning>, <conf>0.40</conf>  
<reasoning>Notice that  $\sum_{j=1}^{n-1} \frac{1}{n^3} = \frac{n-1}{n^3}$ , so the sum becomes  $\sum_{n=2}^{\infty} \frac{(n-1)}{n^3} = \sum_{n=2}^{\infty} \frac{1}{n^2} - \frac{1}{n^3}$ .</reasoning>, <conf>0.40</conf>

<reasoning>This can be split into two sums:  $\sum_{n=2}^{\infty} \frac{1}{n^2} - \sum_{n=2}^{\infty} \frac{1}{n^3}$ .</reasoning>, <conf>0.40</conf>  
<reasoning>We know that  $\sum_{n=1}^{\infty} \frac{1}{n^2} = p$  and  $\sum_{n=1}^{\infty} \frac{1}{n^3} = q$ , so  $\sum_{n=2}^{\infty} \frac{1}{n^2} = p - 1$  and  $\sum_{n=2}^{\infty} \frac{1}{n^3} = q - 1$ .</reasoning>, <conf>0.60</conf>  
<reasoning>Therefore, the original double sum is  $(p - 1) - (q - 1) = p - q$ .</reasoning>, <conf>0.88</conf>  
<final\_answer> $p - q$ </final\_answer>, <final\_confidence>0.99</final\_confidence>

**Example 2: Function evaluation.** This example is from MATH-500. The model correctly evaluates each function value and assigns high confidence after completing the arithmetic aggregation.

Question:  
If  $f(x) = (3x-2)/(x-2)$ , what is the value of  $f(-2) + f(-1) + f(0)$ ?  
Express your answer as a common fraction.

Gold: 14/3  
Prediction: 14/3  
Result: Correct

Model output:  
<reasoning>First, we need to evaluate the function  $f(x) = (3x-2)/(x-2)$  at  $x = -2$ ,  $x = -1$ , and  $x = 0$ .</reasoning>, <conf>0.30</conf>  
<reasoning>For  $f(-2)$ , substitute  $x = -2$  into the function:  $f(-2) = (3(-2) - 2)/(-2 - 2) = (-6 - 2)/(-4) = (-8)/(-4) = 2$ .</reasoning>, <conf>0.60</conf>  
<reasoning>For  $f(-1)$ , substitute  $x = -1$  into the function:  $f(-1) = (3(-1) - 2)/(-1 - 2) = (-3 - 2)/(-3) = (-5)/(-3) = 5/3$ .</reasoning>, <conf>0.60</conf>  
<reasoning>For  $f(0)$ , substitute  $x = 0$  into the function:  $f(0) = (3(0) - 2)/(0 - 2) = (-2)/(-2) = 1$ .</reasoning>, <conf>0.70</conf>  
<reasoning>Now, add the values together:  $f(-2) + f(-1) + f(0) = 2 + 5/3 + 1 = 3 + 5/3 = 9/3 + 5/3 = 14/3$ .</reasoning>

| Model | Method              | HotPotQA       |                    |                  |                  | TriviaQA       |                    |                  |                  | SimpleQA       |                    |                  |                  | CSQA           |                    |                  |                  | GPQA           |                    |                  |                  |
|-------|---------------------|----------------|--------------------|------------------|------------------|----------------|--------------------|------------------|------------------|----------------|--------------------|------------------|------------------|----------------|--------------------|------------------|------------------|----------------|--------------------|------------------|------------------|
|       |                     | Acc $\uparrow$ | Brier $\downarrow$ | ECE $\downarrow$ | AUROC $\uparrow$ | Acc $\uparrow$ | Brier $\downarrow$ | ECE $\downarrow$ | AUROC $\uparrow$ | Acc $\uparrow$ | Brier $\downarrow$ | ECE $\downarrow$ | AUROC $\uparrow$ | Acc $\uparrow$ | Brier $\downarrow$ | ECE $\downarrow$ | AUROC $\uparrow$ | Acc $\uparrow$ | Brier $\downarrow$ | ECE $\downarrow$ | AUROC $\uparrow$ |
| 7B    | Base                | 0.397          | 0.525              | 0.535            | 0.540            | 0.578          | 0.381              | 0.371            | 0.510            | 0.135          | 0.773              | 0.809            | 0.502            | 0.889          | 0.099              | 0.004            | 0.623            | 0.395          | 0.504              | 0.509            | 0.492            |
|       | GRPO                | 0.630          | 0.370              | 0.370            | 0.500            | 0.622          | 0.377              | 0.377            | 0.502            | 0.124          | 0.875              | 0.875            | 0.501            | 0.908          | 0.092              | 0.092            | 0.500            | 0.395          | 0.605              | 0.605            | 0.500            |
|       | C <sup>2</sup> GSPG | 0.617          | 0.254              | 0.090            | 0.659            | 0.627          | 0.195              | 0.210            | 0.745            | 0.103          | 0.201              | 0.310            | 0.549            | 0.909          | 0.090              | 0.097            | 0.693            | 0.417          | 0.233              | 0.169            | 0.563            |
|       | RLCR                | 0.621          | 0.210              | 0.030            | 0.687            | 0.608          | 0.202              | 0.058            | 0.731            | 0.121          | 0.241              | 0.337            | 0.598            | 0.913          | 0.167              | 0.302            | 0.728            | 0.415          | 0.267              | 0.155            | 0.550            |
|       | SCOPE(Stage-I)      | 0.610          | 0.390              | 0.390            | 0.500            | 0.610          | 0.418              | 0.417            | 0.503            | 0.119          | 0.883              | 0.887            | 0.497            | 0.907          | <b>0.090</b>       | 0.079            | 0.606            | 0.431          | 0.543              | 0.548            | 0.515            |
|       | SCOPE(Stage-II)     | 0.603          | <b>0.204</b>       | 0.050            | <b>0.708</b>     | 0.623          | <b>0.165</b>       | 0.067            | <b>0.770</b>     | 0.109          | <b>0.197</b>       | <b>0.295</b>     | <b>0.655</b>     | <b>0.919</b>   | <b>0.090</b>       | 0.083            | 0.697            | <b>0.437</b>   | <b>0.230</b>       | <b>0.153</b>     | <b>0.583</b>     |

Table 9: Full per-dataset QA generalization results for Qwen/Qwen2.5-7B models. The macro-average summary is reported in Table 2. Following RLCR, QA-trained models are trained on HotPotQA-Modified and evaluated on HotPotQA and out-of-domain QA benchmarks. Acc measures final-answer correctness. Brier, ECE, and AUROC evaluate the quality of the reported final-answer confidence. For 7B baselines, Base, GRPO, and RLCR numbers are taken from RLCR; the GRPO row corresponds to the RLVR correctness-only GRPO baseline. CSQA denotes CommonsenseQA.

| Dataset         | GRPO   |          |                | RLCR   |          |                | SCOPE        |              |                |
|-----------------|--------|----------|----------------|--------|----------|----------------|--------------|--------------|----------------|
|                 | Pass@1 | Majority | Conf.-weighted | Pass@1 | Majority | Conf.-weighted | Pass@1       | Majority     | Conf.-weighted |
| GSM8K           | 0.916  | 0.942    | 0.943          | 0.914  | 0.944    | 0.945          | 0.909        | 0.938        | 0.944          |
| MATH-500        | 0.615  | 0.658    | 0.652          | 0.562  | 0.612    | 0.614          | <b>0.640</b> | <b>0.698</b> | <b>0.710</b>   |
| AMC23           | 0.475  | 0.554    | 0.542          | 0.465  | 0.578    | 0.590          | <b>0.480</b> | 0.542        | 0.494          |
| AIME24          | 0.102  | 0.133    | 0.133          | 0.092  | 0.156    | 0.144          | <b>0.121</b> | 0.122        | <b>0.144</b>   |
| OlympiadBench   | 0.304  | 0.355    | 0.353          | 0.276  | 0.337    | 0.341          | 0.294        | <b>0.371</b> | <b>0.396</b>   |
| Big-Math-Digits | 0.691  | 0.759    | 0.760          | 0.689  | 0.768    | 0.772          | 0.656        | 0.743        | 0.747          |
| Overall         | 0.517  | 0.567    | 0.564          | 0.500  | 0.566    | 0.568          | 0.516        | <b>0.569</b> | <b>0.573</b>   |

Table 10: Detailed confidence-weighted aggregation results. GRPO and RLCR use final confidence; SCOPE uses a trace-derived score. All methods are evaluated with the same rollouts used for majority voting. Overall denotes the macro average over the six datasets. Bold entries indicate SCOPE results that are best or tied-best within the corresponding metric.

| Signal                         | $\rho_s \uparrow$ | $\rho_p \uparrow$ | Brier $\downarrow$ | ECE $\downarrow$ | AUROC $\uparrow$ |
|--------------------------------|-------------------|-------------------|--------------------|------------------|------------------|
| Token entropy                  | -0.17             | -0.15             | -                  | -                | 0.60             |
| Prefix log-prob.               | -0.18             | -0.15             | -                  | -                | 0.40             |
| Verbalized conf.               | 0.32              | 0.30              | 0.21               | 0.06             | 0.67             |
| Progress $\Delta u_i$          | 0.20              | 0.22              | -                  | -                | 0.61             |
| <b>Policy <math>u_i</math></b> | <b>0.56</b>       | <b>0.56</b>       | <b>0.18</b>        | 0.16             | <b>0.82</b>      |
| MC prefix success              | 1.00              | 1.00              | 0.00               | 0.00             | 1.00             |

Table 11: Validation of prefix-level signals against MC prefix success on 496 prefix-rollout pairs. The policy-derived prefix evidence  $u_i$  best aligns with MC prefix success, while entropy and prefix log-probability are anti-correlated with prefix success.

| Variant                     | Acc. $\uparrow$ | Brier $\downarrow$ | ECE $\downarrow$ | AUROC $\uparrow$ |
|-----------------------------|-----------------|--------------------|------------------|------------------|
| SCOPE                       | <b>0.437</b>    | <b>0.170</b>       | <b>0.176</b>     | <b>0.768</b>     |
| w/o region mask             | 0.425           | 0.252              | 0.271            | 0.725            |
| w/o KL-to-Stage-I           | 0.420           | 0.203              | 0.224            | 0.735            |
| Stage-I only                | 0.429           | 0.461              | 0.459            | 0.465            |
| Group-level Brier           | 0.4189          | 0.2534             | 0.2452           | 0.6712           |
| Lower Brier weight          | 0.4366          | 0.4684             | 0.4775           | 0.5582           |
| Log score instead of Brier  | 0.413           | 0.232              | 0.211            | 0.724            |
| SFT warmup $\rightarrow$ RL | 0.417           | 0.3490             | 0.3252           | 0.5551           |

Table 12: Full ablation study of SCOPE. We report macro averages over six math benchmarks. Step-level calibration metrics are reported separately in Table 3.

```
>, <conf>0.97</conf>
<final_answer>14/3</final_answer>, <final_confidence>1.00</final_confidence>
```

## E.2 Failure Example

**Example 3: Parenthesization failure.** This example shows a failed trajectory. The model assigns low stepwise confidence throughout the reasoning process, and the final confidence remains low, indicating that the confidence trace can expose uncertainty even when the final answer is incorrect.

Question:  
The expression  $1 + 2 + 3 - 4 + 5 + 6$  is equivalent to 13. If we add parentheses in various locations, we can change this value. For example,

$1+2+3-(4+5)+6 = 1+2+3-9+6 = 6-9+6 = 3$ . If we only add parentheses and do not rearrange the order of the terms, what is the minimum possible value this expression can obtain? We cannot change addition or subtraction into multiplication.

Enter your answer as a single value.

Gold: -9  
Prediction: 1  
Result: Incorrect

Step confidence trace: [0.19, 0.13, 0.32, 0.31]  
Final confidence: 0.22

Model output:  
<reasoning>We need to minimize the value of the expression  $1 + 2 + 3 - 4 + 5 + 6$  by strategically placing parentheses. The key is to group the positive terms in such a way that their sum is as small as possible when compared to the negative term  $-4$ .</reasoning>, <conf>0.19</conf>

| Weights / $\beta$    | 0             | 0.005         | 0.01                 | 0.02                 |
|----------------------|---------------|---------------|----------------------|----------------------|
| w0 [.1, .1, .5, .5]  | 0.420 / 0.735 | 0.406 / 0.796 | 0.421 / 0.774        | 0.405 / 0.771        |
| w1 [0, 0, .5, .5]    | 0.415 / 0.732 | 0.419 / 0.759 | <b>0.437</b> / 0.768 | 0.414 / 0.784        |
| w2 [.1, .1, .75, .5] | 0.423 / 0.740 | 0.423 / 0.768 | 0.432 / 0.787        | 0.429 / <b>0.799</b> |
| w3 [.1, .1, 1.0, .5] | 0.434 / 0.756 | 0.391 / 0.725 | 0.427 / 0.769        | 0.410 / 0.785        |

Table 13:  $\beta$  scan on 6 math datasets: GSM8K, MATH-500, AMC23, AIME24, OlympiadBench, and Big-Math-Digits. Each cell reports Acc / AUROC, where both metrics are computed as the unweighted macro average across the six datasets. Rows vary the reward-weight vector  $w = [w_{\text{fmt}}, w_{\text{acc}}, w_{\text{step}}, w_{\text{final}}]$ , where the four entries correspond to the format reward, accuracy reward, step-level Brier reward, and final-confidence Brier reward, respectively.

```

<reasoning>Consider different groupings:
- 1 + (2 + 3 - 4) + (5 + 6) = 1 + 1 + 11 = 13.
- 1 + 2 + (3 - 4 + 5) + 6 = 1 + 2 + 4 + 6 = 13.
- 1 + 2 + 3 - (4 - 5 + 6) = 1 + 2 + 3 - 5 = 1.
- (1 + 2 + 3 - 4) + 5 + 6 = 2 + 5 + 6 = 13.
- 1 + (2 + 3 - 4 + 5) + 6 = 1 + 6 + 6 = 13.
</reasoning>, <conf>0.13</conf>
<reasoning>From the above groupings, the minimum value we can
  obtain is 1.</reasoning>, <conf>0.32</conf>
<reasoning>To verify, check whether we can get a value less
  than 1. If we group the terms differently, we will
  always end up with a sum that is at least 1 because the
  smallest negative term is -4 and the sum of the
  positive terms is at least 13, so the minimum value we
  can get is 1 by using the grouping 1 + 2 + 3 - (4 - 5 +
  6) = 1.</reasoning>, <conf>0.31</conf>
<final_answer>1</final_answer>, <final_confidence>0.22</
  final_confidence>

```