ON THE PITFALLS OF VERBALIZED CONFIDENCE PRIORS FOR CALIBRATING LARGE REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large reasoning models (LRMs) often suffer from overconfidence when expressing their uncertainty. Confidence-aware reinforcement learning (RL) offers a promising way to optimize calibration. However, it relies on on-policy rollouts and is thus constrained by the model's pre-RL confidence distribution, which we term *confidence prior*. In this work, we reveal that off-the-shelf LRMs exhibit a confidence prior heavily concentrated on a few high values, which persists throughout RL. Theoretically, we prove that this concentration suppresses policy gradient updates for rarely sampled confidence values and inflates the lower bound on expected Brier risk. To overcome this exploration bottleneck, we propose **CalibSFT**, a plug-and-play supervised fine-tuning stage that shapes a calibrated confidence prior with broad support before RL. For each question, CalibSFT constructs confidence targets combining its success rate with response-level correctness, which provably preserves proper-scoring optimality, and then balances training responses across the confidence spectrum to enable diverse confidence exploration during RL. To learn from incorrect responses without imitating their reasoning, CalibSFT introduces correctness-conditional supervision, guiding confidence across all responses while supervising reasoning only on correct ones. Across 16 mathematical and general reasoning benchmarks, incorporating CalibSFT reduces calibration errors and improves discrimination across five representative RL algorithms while preserving comparable accuracy. Furthermore, CalibSFT delivers practical benefits for downstream selective prediction and model routing.

1 INTRODUCTION

Large reasoning models (LRMs) have made substantial progress on complex tasks such as mathematics and coding (Guo et al., 2025; Yang et al., 2025). However, these accuracy gains do not imply reliable confidence: LRMs often remain overconfident in incorrect answers (Mei et al., 2025). A reliable model should express *well-calibrated* confidence (Guo et al., 2017), matching its probability of correctness. Calibrated confidence indicates when an answer can be trusted, which enables practical applications such as selective prediction (Shen, 2026) and model routing (Hao et al., 2026).

Beyond estimates derived from token probabilities, internal representations, or sampling consistency (Luo et al., 2025a; Srey et al., 2026; Kuhn et al., 2023), models can directly verbalize confidence alongside their answers. Unlike these alternatives, verbalized confidence can be obtained in a single black-box generation. However, off-the-shelf LRMs typically produce miscalibrated scores that concentrate on a few high values regardless of correctness (Mei et al., 2025; Cacioli, 2026). To address this issue, existing post-training approaches calibrate verbalized confidence via either supervised fine-tuning (SFT) on constructed targets (Xu et al., 2024; Zhang et al., 2026a) or reinforcement learning (RL) guided by calibration rewards (Damani et al., 2026; Ma et al., 2026).

Among these approaches, confidence-aware RL is particularly promising owing to its superior generalization beyond the training distribution (Chu et al., 2025). However, because confidence-aware RL explores via on-policy rollouts, its optimization is fundamentally constrained by the confidence distribution before RL, which we term *confidence prior*. In practice, this prior concentrates heavily on a narrow set of high values across both correct and incorrect responses (Figure 1). Although calibration rewards explicitly penalize miscalibration, this concentration persists throughout policy

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.

optimization and limits the exploration of diverse confidence values (Figure 2). Our theoretical analysis in Section 3 reveals that the policy gradient for each confidence value scales with its sampling probability, which suppresses updates on rarely sampled values. The analysis further shows that a concentrated prior induces miscalibration by raising the lower bound on expected Brier risk. As a result, post-RL confidence fails to reflect task difficulty in practice: mean confidence on MinervaMath is only 9.7% lower than on Math500 despite a 39.1% drop in Pass@1 (Figure 3).

To overcome this exploration bottleneck, we propose **CalibSFT**, a supervised fine-tuning stage that shapes a calibrated confidence prior with broad support before confidence-aware RL. Specifically, for each training question, CalibSFT samples multiple responses from the base model and assigns each a confidence target that balances the question-level success rate with response-level correctness. We further prove that this mixed target preserves the question-level optimal confidence under any strictly proper scoring rule. To enable diverse confidence exploration during RL, it employs target-balanced sampling, drawing an equal number of responses for each target value. During fine-tuning, it introduces correctness-conditional supervision, guiding confidence on all responses while restricting reasoning and answer supervision to correct ones, so that the model learns low confidence on incorrect responses without imitating their solutions. Since CalibSFT reshapes the confidence prior without altering RL objectives, it can be seamlessly integrated with any confidence-aware RL framework.

We evaluate CalibSFT on 16 benchmarks spanning both in-distribution mathematical reasoning and out-of-distribution general reasoning of varying difficulty. Across 5 confidence-aware RL methods, initializing with CalibSFT outperforms base-model initialization in both calibration and discrimination without compromising task accuracy (Table 1). Consistent with our analysis, CalibSFT mitigates confidence concentration after RL (Figure 4) and strengthens the correlation between task-level confidence and accuracy (Table 2). It also remains effective across different confidence formats and model families (Tables 3 and 4). Furthermore, CalibSFT provides practical benefits for downstream applications like selective prediction and model routing (Appendix E.1–E.2).

Our contributions are summarized as follows:

1. We identify that LRMs exhibit a concentrated confidence prior that limits on-policy RL exploration, and theoretically prove that this concentration suppresses policy gradients and hinders calibration by increasing the lower bound on expected Brier risk.
2. We propose CalibSFT, a supervised fine-tuning stage that constructs proper-scoring confidence targets and balances them across the confidence spectrum to enable RL exploration.
3. Extensive evaluations across 16 benchmarks demonstrate that CalibSFT delivers substantial calibration and discrimination gains across all five RL baselines, while exhibiting remarkable robustness across diverse confidence formats, architectures, and downstream applications.

2 Preliminaries

In this work, we study the calibration of verbalized numerical confidence in large reasoning language models (LRMs) (Xiong et al., 2024; Damani et al., 2026). Given a question x, an LRM π_θ generates a response $o = (r, a, c)$, where r is a reasoning trace, a is an answer, and $c \in [0, 1]$ is the model's estimated probability that a is correct. The model verbalizes c conditioned on the prefix (x, r, a) via $\pi_\theta(c \mid x, r, a)$. An automated verifier V evaluates the answer against the ground truth to provide a correctness label $z = V(x, a) \in \{0, 1\}$.

2.1 Calibration Metrics

To evaluate verbalized confidence c, we assess both calibration and discrimination. Let $C \in [0, 1]$ and $Z \in \{0, 1\}$ denote confidence and correctness random variables, respectively. Confidence is perfectly calibrated if $\Pr(Z = 1 \mid C) = C$ almost surely, and discriminative if it ranks correct responses above incorrect ones. We measure calibration via **Expected Calibration Error (ECE)** (Guo et al., 2017), defined as $\mathbb{E}_C[|\Pr(Z = 1 \mid C) - C|]$. With N responses in B bins $\{\mathcal{I}_b\}_{b=1}^{B}$, ECE is estimated as:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|\mathcal{I}_b|}{N} \left|\mathrm{acc}(\mathcal{I}_b) - \mathrm{conf}(\mathcal{I}_b)\right|, \tag{1}$$

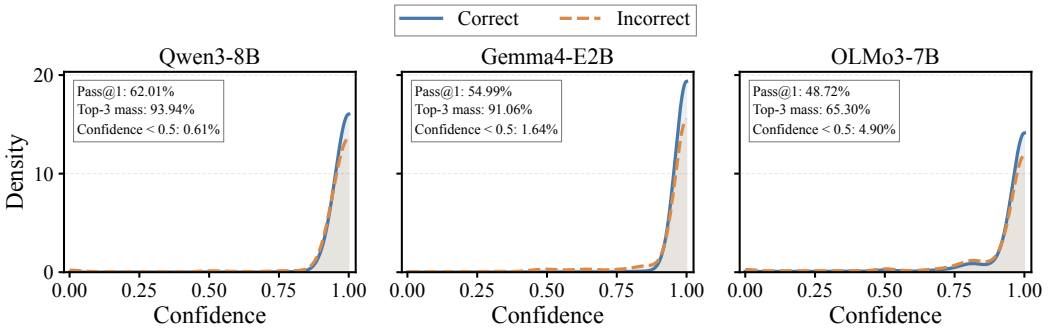

Figure 1: Confidence distributions from three base models on DeepScaleR-Train. Confidence concentrates on a few high values for both correct and incorrect responses.

where $\mathrm{acc}(\mathcal{I}_b)$ and $\mathrm{conf}(\mathcal{I}_b)$ are the mean correctness and confidence in bin b. While ECE assesses bin-level calibration, the **Brier score** (Brier, 1950) computes response-level squared error, $\frac{1}{N}\sum_{j=1}^{N}(c_j - z_j)^2$. However, calibration does not guarantee discrimination. For example, assigning constant confidence equal to overall accuracy yields zero ECE while giving correct and incorrect responses identical scores (Eusebi et al., 2026). We therefore report **AUROC** to evaluate threshold-agnostic discrimination between correct and incorrect answers.

2.2 Confidence-aware Reinforcement Learning

Confidence-aware rewards. Reinforcement learning with verifiable rewards (RLVR) trains the policy using the correctness label z as the reward. To optimize calibration alongside accuracy, confidence-aware RL introduces an additional calibration reward depending on both c and z. For example, RLCR (Damani et al., 2026) adopts the negative Brier loss, yielding the combined reward:

$$R(o) = z - (c - z)^2. \tag{2}$$

Policy optimization. Confidence-aware RL typically optimizes this reward with Group Relative Policy Optimization (GRPO) (Shao et al., 2024). For each question, GRPO samples G responses $\{o_i\}_{i=1}^{G}$ from $\pi_{\theta_{\text{old}}}$ and normalizes the group-wise rewards $R_i = R(o_i)$ to obtain the advantage:

$$\widehat{A}_i = \frac{R_i - \bar{R}}{\sigma_R + \epsilon}, \qquad \bar{R} = \frac{1}{G}\sum_{j=1}^{G} R_j, \qquad \sigma_R = \mathrm{std}\big(\{R_j\}_{j=1}^{G}\big), \tag{3}$$

where $\epsilon > 0$ ensures numerical stability. GRPO then updates the policy by maximizing

$$J_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{L_i}\sum_{t=1}^{L_i}\min\Big(\rho_{i,t}(\theta)\widehat{A}_i,\ \mathrm{clip}(\rho_{i,t}(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})\,\widehat{A}_i\Big)\right], \tag{4}$$

where $o_{i,t}$ is the t-th token of o_i, L_i is the length of o_i, $\rho_{i,t}(\theta) = \pi_\theta(o_{i,t} \mid x, o_{i,<t})/\pi_{\theta_{\text{old}}}(o_{i,t} \mid x, o_{i,<t})$ is the importance ratio, and ϵ_{clip} is the clipping parameter. Because this optimization is on-policy, it only learns from confidence values sampled during rollouts, which motivates us to examine the confidence distribution of LRMs before and during RL.

3 Motivation

While confidence-aware RL penalizes overconfidence through explicit reward formulations (Damani et al., 2026; Ma et al., 2026; Yang et al., 2026b), its efficacy remains constrained by the support of sampled confidence values. To understand this limitation, we examine the confidence prior before RL, its dynamics during RL, and calibration at test time.

Confidence priors concentrate on a few high values. We refer to the confidence distribution before RL as the *confidence prior*. To characterize this prior, we sample 10 responses per question

on DeepScaleR-Train using Qwen3-8B (Yang et al., 2025), Gemma4-E2B (Abd et al., 2026), and OLMo3-7B (Ettinger et al., 2025). As shown in Figure 1, all models place most of their confidence mass on a few high values, with fewer than 5% of responses below 0.5. This concentration at high confidence is also observed among incorrect answers. Consequently, RL starts from a concentrated confidence prior and rarely samples low confidence values early in training.

RL improves calibration but confidence concentration persists. To examine whether confidence-aware RL overcomes this concentration, we train Qwen3-8B with RLCR (Damani et al., 2026) on DeepScaleR-Train. As shown in Figure 2(a), while the calibration reward lowers ECE, confidence discrimination (AUROC) fails to increase. Figure 2(b) further explains this failure: the number of distinct confidence values sampled per question closely tracks the confidence token entropy, which remains low throughout training and limits exploration across the confidence space. Consequently, the model merely shifts probability mass among a few dominant values to satisfy average calibration, without learning to distinguish correct from incorrect responses. In Appendix D.1, we show that this restricted exploration and low confidence entropy persist across other confidence-aware RL methods.

To understand why this concentration persists, we formulate verbalized confidence generation as action selection under a softmax policy (Garg et al., 2022). Let $h = \phi(x, r, a)$ denote the context representation on which the policy conditions to verbalize confidence, so that $\pi_\theta(c \mid x, r, a) = \pi_\theta(c \mid h)$, and let $p_h = \Pr(Z = 1 \mid h)$. Since $Z = V(x, a)$ is determined by the prefix and C depends on the prefix only through h, we have $C \perp Z \mid h$.

Proposition 1 (Policy gradient suppression under concentrated priors)**.** *Given a context representation h and a finite set of confidence values $\{c_1, \ldots, c_K\} \subseteq [0, 1]$, let $\eta = (\eta_1, \ldots, \eta_K)$ denote the logits with $\pi_i = \Pr(C = c_i \mid h) = \exp(\eta_i)/\sum_{j=1}^{K} \exp(\eta_j)$. Define the expected reward $u_i(h) = -\mathbb{E}[(c_i - Z)^2 \mid h]$. Let $J_h(\eta) = \sum_{i=1}^{K} \pi_i u_i(h)$ and $\Delta_h = \max_i u_i(h) - \min_i u_i(h) \leq 1$. Then*

$$\frac{\partial J_h}{\partial \eta_i} = \pi_i\big(u_i(h) - J_h\big), \qquad \left|\frac{\partial J_h}{\partial \eta_i}\right| \leq \pi_i \Delta_h. \tag{5}$$

If $\pi_k = 1 - \varepsilon$ for some dominant action $k \in \{1, \ldots, K\}$, then

$$\|\nabla_\eta J_h\|_1 \leq 2\varepsilon\Delta_h \leq 2\varepsilon. \tag{6}$$

Proposition 1 shows that the logit gradient for any action c_i scales with its sampling probability π_i, consistent with the $\mathcal{O}(\pi_i)$ gradient decay in softmax policies (Mei et al., 2020). Under a concentrated prior ($\pi_k = 1 - \varepsilon$), the total gradient norm is bounded by $2\varepsilon\Delta_h$. Crucially, when the dominant value c_k is far from p_h, the gradient that shifts probability mass away from c_k is also bounded by $\varepsilon\Delta_h$, which hinders on-policy RL from escaping this suboptimal initialization and explains why it struggles to correct overconfidence even under calibration rewards. We provide the proof in Appendix B.1.

Persistent confidence concentration hinders downstream calibration. Beyond restricting training exploration, this concentrated prior misaligns confidence with task difficulty at test time. As shown in Figure 3, across two benchmarks of contrasting difficulty (Math500 and MinervaMath), Pass@1 drops sharply by 39.1%, yet mean confidence changes by only 9.7%. Because the RL-trained model still concentrates its confidence on a few high values, it struggles to assign lower confidence to hard questions. To theoretically characterize the cost of this miscalibration, we analyze the expected Brier loss $\mathbb{E}[(C - Z)^2 \mid h]$ (Gneiting & Raftery, 2007) and establish the following lower bound:

Proposition 2 (Brier risk under prior misalignment)**.** *Suppose $C \perp Z \mid h$. For any $\delta > 0$, let $m_\delta(h) = \Pr(|C - p_h| < \delta \mid h)$. Then*

$$\mathbb{E}[(C - Z)^2 \mid h] \geq p_h(1 - p_h) + \delta^2\big(1 - m_\delta(h)\big). \tag{7}$$

Proposition 2 lower-bounds the expected Brier loss by the conditional variance $p_h(1 - p_h)$ and an excess error $\delta^2\big(1 - m_\delta(h)\big)$ from confidence deviations. When the policy rarely samples values near p_h (i.e., $m_\delta(h) \approx 0$), this second term approaches δ^2. Because confidence-aware RL only receives reward feedback on sampled values and suppresses gradients on rarely sampled values (Proposition 1), calibration rewards struggle to allocate mass to values near p_h that the policy rarely explores. In Appendix B.2, we provide the full proof and extend this bound to best-of-G sampling.

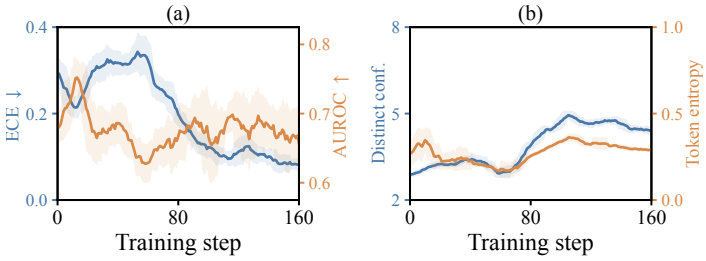

Figure 2: Training dynamics of RLCR on Qwen3-8B. (a) The calibration reward lowers ECE, but AUROC does not improve. (b) The number of distinct confidence values sampled per question closely tracks confidence token entropy, and their persistently low levels prevent sufficient exploration across the confidence space.

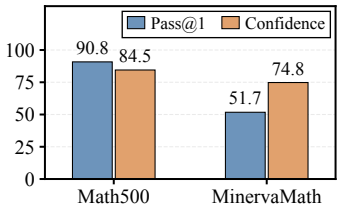

Figure 3: Pass@1 and mean confidence on Math500 and MinervaMath of RLCR on Qwen3-8B. Mean confidence shifts far less than Pass@1 across tasks.

4 METHOD

Section 3 shows that confidence concentration persists throughout RL, since rarely sampled confidence values receive minimal policy gradients. Instead of relying on on-policy exploration, we propose CalibSFT to shape a well-calibrated confidence prior before policy optimization. CalibSFT first constructs confidence targets that mix question-level success rates with response-level correctness, preserving each question's optimal confidence while supporting within-question discrimination. It then applies target-balanced sampling to broaden support across the confidence spectrum, so that RL can explore diverse confidence values. Finally, it supervises confidence on all responses but restricts reasoning and answer supervision to correct ones, so that the model learns to express low confidence on errors without reinforcing incorrect solutions.

4.1 PROPER-SCORING CONFIDENCE TARGETS

Under strictly proper scoring rules, optimal confidence matches the true correctness probability (Gneiting & Raftery, 2007). As this probability is not observed directly, we estimate it from base-model rollouts. For each question x, we sample n responses with correctness $z_i \in \{0, 1\}$ and success rate $q_x = \frac{1}{n}\sum_{i=1}^{n} z_i$. However, q_x alone assigns identical targets to all responses and removes within-question discrimination. To balance difficulty calibration with response discrimination, we construct the confidence target for response i as:

$$c_i^* = \lambda q_x + (1 - \lambda) z_i, \tag{8}$$

where the mixing weight $\lambda \in [0, 1]$ balances question-level success rate with response correctness. We further show that c_i^* preserves the question-level optimal confidence under any strictly proper scoring rule.

Proposition 3 (Proper-scoring optimality of the mixed target)**.** *Let $\ell(\hat{c}, z)$ be a strictly proper scoring loss for a prediction $\hat{c} \in [0, 1]$ and $z \in \{0, 1\}$, extended to $t \in [0, 1]$ via $\ell(\hat{c}, t) = t\,\ell(\hat{c}, 1) + (1 - t)\,\ell(\hat{c}, 0)$. For a question x, let $Z_1, \ldots, Z_n$ denote the correctness of n independent base-model rollouts with $p_x = \Pr(Z_i = 1 \mid x)$, and let $C_i^* = \lambda Q_x + (1 - \lambda) Z_i$ with $Q_x = \frac{1}{n}\sum_{j=1}^{n} Z_j$. Then $\mathbb{E}[C_i^* \mid x] = p_x$, and for any $\hat{c}$ at which ℓ is finite,*

$$\mathbb{E}[\ell(\hat{c}, C_i^*) \mid x] = \mathbb{E}[\ell(\hat{c}, Z_i) \mid x]. \tag{9}$$

Consequently, p_x remains the unique minimizer of the expected scoring loss on question x.

Proposition 3 shows that, under the rollout distribution, the mixed target preserves the optimal confidence p_x for each question. A larger n reduces target variance, while λ sets a target margin of $1 - \lambda$ between correct and incorrect responses within a question. We use $n = 50$ and $\lambda = 0.5$ by default, and provide the proof, variance derivations, and conditioning analysis in Appendix B.3.

In practice, we replace each response's confidence with c_i^*, preserving its reasoning trace and answer. Because many questions are solved by all or none of their rollouts, 61% of raw targets are exactly 0 or 1, so training directly on this pool would concentrate the SFT prior at these extremes. We therefore

apply *target-balanced sampling*, which draws an equal number of responses for each target value (Appendix C.1), establishing a well-supported initial prior before downstream RL. Since balancing reweights responses away from the rollout distribution in Proposition 3, we verify in Table 14 that it maintains comparable calibration while broadening confidence support.

4.2 Correctness-Conditional Supervision

The constructed data contain both correct and incorrect responses. The incorrect ones are kept because their lower targets guide the model to express low confidence on its own errors, whereas imitating their reasoning would train the model on incorrect solutions. CalibSFT therefore supervises confidence on all responses and restricts reasoning and answer supervision to correct ones.

Formally, let $\widetilde{o}$ denote a response whose confidence has been replaced by its constructed target, and let $\mathcal{T}_{r,a}$ and $\mathcal{T}_c$ denote the positions of its reasoning/answer tokens and its confidence tokens, respectively. The CalibSFT objective is then

$$\begin{aligned}\mathcal{L}_{\text{CalibSFT}} = (1-\beta)\,\mathbb{E}_{\text{correct}}\left[-\frac{1}{|\mathcal{T}_{r,a}|}\sum_{t\in\mathcal{T}_{r,a}}\log\pi_\theta(\widetilde{o}_t \mid x, \widetilde{o}_{<t})\right] \\ +\beta\,\mathbb{E}_{\text{all}}\left[-\frac{1}{|\mathcal{T}_c|}\sum_{t\in\mathcal{T}_c}\log\pi_\theta(\widetilde{o}_t \mid x, \widetilde{o}_{<t})\right].\end{aligned} \tag{10}$$

Here $\mathbb{E}_{\text{correct}}$ averages over the correct responses and $\mathbb{E}_{\text{all}}$ over all of them. Averaging within each segment prevents the longer reasoning sequences from dominating the loss by token count. The weight β balances confidence supervision against reasoning supervision, and we set $\beta = 0.5$ so that the two terms contribute equally.

Integration with confidence-aware RL. CalibSFT provides an initialization for confidence-aware RL without changing its objective, and thus applies to different RL methods. We evaluate CalibSFT across RL methods in Section 5.2, analyze its impact on confidence distributions in Section 5.3, and test its generality across confidence formats and base models in Section 5.4. Finally, Appendix D.5 ablates the target construction, mixing weight λ, and supervision design.

5 Experiments

5.1 Experimental Setup

Baselines. We compare methods spanning answer-only and confidence-aware training. **Base** is the off-the-shelf checkpoint, while **RLVR** optimizes only answer correctness. We evaluate five representative confidence-aware RL baselines covering two training paradigms: (1) **joint training**, which simultaneously optimizes correctness and confidence, including **RLCR** (Damani et al., 2026), **CoCA** (Li et al., 2026a), **DCPO** (Ma et al., 2026), and **C3RL** (Yang et al., 2026b). (2) **two-stage training**, where **ReDoubt** (Bani-Harouni et al., 2026) freezes reasoning optimization and tunes confidence on top of a trained RLVR model. For each method, we compare initialization from the base model and from CalibSFT. All methods use Qwen3-8B (Yang et al., 2025) as the default backbone within the **DAPO** training framework (Yu et al., 2025).

Datasets. We use **DeepScaleR-Preview** (Luo et al., 2025b) as the mathematical training dataset. We evaluate on a broad range of in-distribution (ID) and out-of-distribution (OOD) datasets spanning different levels of difficulty. For ID mathematical reasoning, we use **DeepScaleR-Eval**, **MATH-500** (Hendrycks et al., 2021), **MinervaMath** (Lewkowycz et al., 2022), **OlympiadBench** (He et al., 2024), **GSM8K** (Cobbe et al., 2021), and **AIME 2024–2026** (Dekoninck et al., 2026). For general reasoning, we use **HotpotQA** (Yang et al., 2018), **MuSiQue** (Trivedi et al., 2022), **TriviaQA** (Joshi et al., 2017), **NQOpen** (Lee et al., 2019), **PopQA** (Mallen et al., 2023), **WebQuestions** (Berant et al., 2013), **DROP** (Dua et al., 2019), and **LiveBenchReasoning** (White et al., 2025).

Evaluation metrics. We evaluate model performance using complementary metrics that capture answer correctness, confidence discrimination, and absolute calibration. We report **Pass@1** for

Table 1: Performance comparison on in-distribution and out-of-distribution benchmarks. Each entry averages the datasets within the corresponding split. CalibSFT initialization improves calibration and discrimination for all RL baselines while also raising Pass@1.

	In-distribution				Out-of-distribution			
Method	Pass@1 ↑	AUROC ↑	Brier ↓	ECE ↓	Pass@1 ↑	AUROC ↑	Brier ↓	ECE ↓
Base	59.32	71.90	33.20	34.36	45.36	**65.71**	45.15	46.94
RLVR	**64.16**	73.16	31.10	31.52	43.88	64.19	50.29	51.52
CalibSFT	58.51	**79.93**	**17.30**	**14.23**	**46.37**	61.20	**29.15**	**25.49**
RLCR	65.33	82.36	14.43	14.46	43.96	69.38	31.14	31.17
+ CalibSFT	**67.20**	**86.32**	**12.77**	**8.59**	**45.71**	**71.31**	**23.24**	**18.20**
ReDoubt	65.59	81.05	15.48	16.94	43.71	69.88	25.07	23.98
+ CalibSFT	**67.59**	**86.99**	**12.90**	**14.28**	**45.55**	**73.93**	**22.42**	**18.89**
CoCA	63.62	80.66	16.31	19.95	43.68	66.79	26.67	24.12
+ CalibSFT	**65.07**	**85.63**	**16.06**	**18.51**	**45.55**	**71.22**	**22.94**	**18.89**
DCPO	64.68	78.21	14.69	12.16	43.99	68.13	32.85	32.41
+ CalibSFT	**64.84**	**85.08**	**12.83**	**10.39**	**44.82**	**72.75**	**21.73**	**17.31**
C3RL	65.62	68.36	25.44	28.06	43.44	62.19	45.54	48.00
+ CalibSFT	**66.17**	**79.70**	**15.91**	**14.35**	**44.60**	**63.03**	**23.89**	**17.38**

answer correctness, **AUROC** for ranking correct responses above incorrect responses, and **Brier score** (Brier, 1950) and **ECE** (Guo et al., 2017) for probabilistic calibration. We present additional metrics for calibration and confidence diversity in Appendix D.4.

Implementation details. During evaluation, we use temperature 0.6 and sample 32 responses per problem for the AIME benchmarks and 4 responses per problem for all other benchmarks. All experiments are conducted on a single node with eight NVIDIA B20Z GPUs. We report our training details in Appendix C and training cost analysis in Appendix D.8.

5.2 Overall Results

CalibSFT consistently benefits diverse confidence-aware RL methods. As shown in Table 1, initializing RL from CalibSFT reduces average Brier score and ECE and increases AUROC across all five confidence-aware RL baselines on both mathematical and general-reasoning benchmarks. As a concrete example, CalibSFT reduces DCPO's ECE from 29.05 to 8.83 while lifting its AUROC from 63.57 to 76.00 on MinervaMath. Moreover, these calibration gains come without sacrificing average accuracy. The average Pass@1 improves by 1.20 in-distribution and 1.49 out-of-distribution across the five methods. We provide detailed per-dataset results in Tables 6–7 (Appendix D.2) and verify stability across random seeds in Appendix D.3. Overall, these results establish CalibSFT as a plug-and-play initialization that consistently strengthens confidence-aware RL for better calibration.

CalibSFT and RL contribute complementary gains. To disentangle their contributions, we compare CalibSFT and RLVR individually. CalibSFT alone substantially mitigates overconfidence, with ECE dropping from 34.36 to 14.23 in-distribution and from 46.94 to 25.49 out-of-distribution. In contrast, Base and RLVR remain miscalibrated, indicating that answer correctness alone is insufficient to reshape confidence. However, calibration does not imply better confidence ranking: CalibSFT's OOD AUROC even declines from 65.71 to 61.20, reflecting the limited generalization of supervised training. Confidence-aware RL fills exactly this gap by directly optimizing discrimination, and combining it with CalibSFT yields the strongest overall performance.

5.3 Calibration Analysis

CalibSFT aligns confidence with accuracy across tasks. A well-calibrated model should express higher confidence on tasks it can reliably solve and lower confidence on harder ones. To test this, we compute Pearson and Spearman correlations between per-dataset mean confidence and per-dataset accuracy for each RL method on the ID and OOD splits separately. As shown in Table 2, CalibSFT improves Spearman correlation for all RL baselines on both mathematical and general reasoning

Table 2: Pearson r and Spearman ρ correlation between accuracy and mean confidence across ID and OOD datasets. Both correlations increase for all RL methods when incorporating CalibSFT, indicating that confidence can better reflect task-level accuracy.

		RLCR		ReDoubt		CoCA		DCPO		C3RL	
Dataset	Method	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
ID	Baseline	0.768	0.619	0.829	0.762	0.750	0.643	0.669	0.619	0.635	0.595
	+CalibSFT	**0.963**	**0.952**	**0.976**	**0.976**	**0.887**	**0.976**	**0.996**	**0.976**	**0.896**	**0.833**
OOD	Baseline	0.505	0.548	0.667	0.762	0.447	0.571	0.467	0.548	0.681	0.714
	+CalibSFT	**0.729**	**0.619**	**0.738**	**0.881**	**0.605**	**0.619**	**0.611**	**0.643**	**0.848**	**0.929**

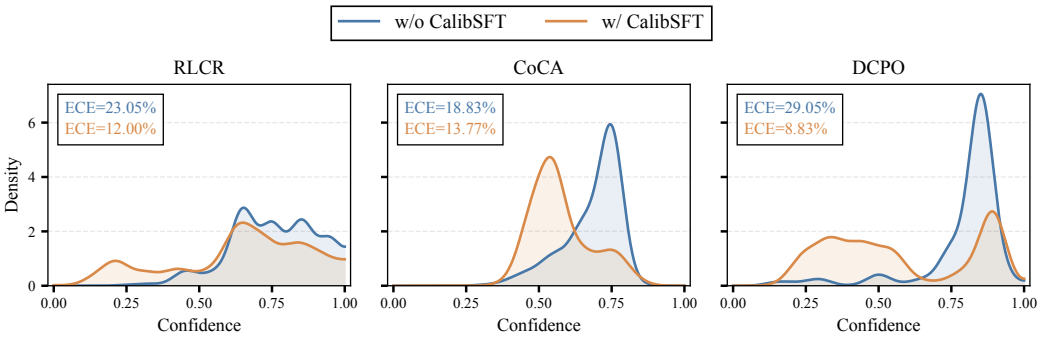

Figure 4: Confidence distributions on MinervaMath for RLCR, CoCA, and DCPO. CalibSFT reduces concentration at dominant confidence values and lowers ECE for all three methods.

benchmarks. For example, DCPO's Spearman correlation rises from 0.619 to 0.976 on mathematical tasks and from 0.548 to 0.643 on general reasoning. Pearson correlation also improves consistently across all baselines and splits (e.g., from 0.505 to 0.729 for RLCR on OOD). These results indicate that CalibSFT helps the model assign higher confidence to tasks it is more likely to answer correctly (see Appendix D.7 for reliability diagrams across benchmarks of varying difficulty).

CalibSFT mitigates confidence concentration after RL. Prior work observes that confidence estimates often concentrate on a few dominant values even after confidence-aware RL training (Wang et al., 2026; Yang et al., 2026b). To examine whether CalibSFT initialization alleviates this concentration, we compare RLCR, CoCA, and DCPO on MinervaMath, with and without CalibSFT. As shown in Figure 4, all three methods trained from Base exhibit heavy peaks at a few high-confidence values, while initializing from CalibSFT flattens these peaks and aligns the mean confidence with the actual accuracy. For example, DCPO's dominant peak near high confidence is redistributed toward lower values while accuracy is preserved, reducing ECE from 29.05% to 8.83%. These results suggest that CalibSFT initialization propagates lower concentration and better calibration into the downstream RL policy.

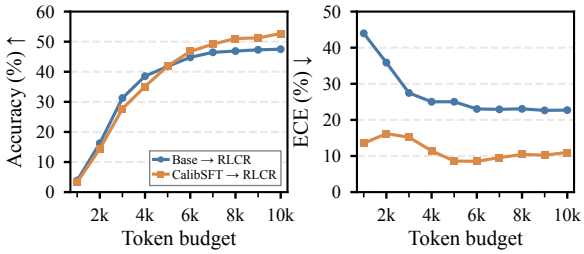

Figure 5: Accuracy (↑) and ECE (↓) on AIME 2026 across token budgets for RLCR. CalibSFT initialization yields lower ECE across all budgets while maintaining accuracy.

The calibration gains of CalibSFT persist across reasoning budgets. We compare RLCR with and without CalibSFT on AIME 2026 across a range of test-time token budgets. As shown in Figure 5, CalibSFT achieves substantially lower ECE at every budget while accuracy remains comparable to the RLCR baseline. For example, at a 10,000-token budget, ECE drops from 22.71 to 10.97.

Table 3: AUROC (↑) and ECE (↓) of RLCR on Qwen3-8B across confidence formats. "Base" denotes standard RLCR, and "+Ours" denotes RLCR initialized from CalibSFT. Across all four formats, CalibSFT consistently improves AUROC and reduces ECE across the 8 ID and 8 OOD benchmarks.

	ID				OOD			
	ECE ↓		AUROC ↑		ECE ↓		AUROC ↑	
Confidence Format	Base	+Ours	Base	+Ours	Base	+Ours	Base	+Ours
Linguistic (low / medium / high)	17.71	**15.90**	73.56	**79.58**	24.06	**20.31**	64.20	**67.73**
Integer (0–9)	14.69	**14.06**	79.00	**84.14**	26.31	**16.70**	67.00	**69.36**
Probability $\{0.00, 0.05, \ldots, 1.00\}$	14.05	**12.13**	81.81	**86.05**	32.81	**19.30**	68.72	**71.23**
Probability $\{0.00, 0.01, \ldots, 1.00\}$	14.05	**8.59**	81.81	**86.32**	32.81	**18.20**	68.72	**71.31**

Table 4: AUROC (↑) and ECE (↓) of RLCR on Gemma4-E2B-Instruct across benchmarks of varying difficulty. CalibSFT noticeably improves calibration across all difficulty levels.

		In-distribution			Out-of-distribution			
Metric	Method	Math500	Olympiad	Minerva	DROP	HotpotQA	NQOpen	Average
ECE ↓	RLCR	7.84	17.69	30.34	19.92	26.41	38.25	23.41
	+CalibSFT	**5.34**	**7.76**	**12.74**	**13.30**	**13.04**	**22.77**	**12.49**
AUROC ↑	RLCR	75.47	83.24	66.29	60.60	60.48	76.85	70.49
	+CalibSFT	**80.34**	**87.35**	**70.75**	**65.45**	**62.66**	**77.13**	**73.95**

5.4 Generality of CalibSFT

CalibSFT is effective across diverse confidence formats. We train RLCR with CalibSFT using four confidence target formats: two-decimal probabilities on a fine grid $\{0.00, 0.01, \ldots, 1.00\}$ or a coarse grid $\{0.00, 0.05, \ldots, 1.00\}$, integer scores from 0 to 9, and linguistic labels (*low*, *medium*, *high*) (see Appendix C.3 for details). As shown in Table 3, CalibSFT consistently improves both ECE and AUROC across all four formats, indicating that its benefit does not rely on a particular way of expressing confidence. The reduction in ECE is largest with the fine probability grid, whose granularity lets confidence approximate the probability of correctness most closely. In Appendix D.6, we further show that CalibSFT consistently improves calibration under alternative confidence estimators, including internal token probabilities and self-consistency voting.

CalibSFT is agnostic to the base model. To examine the transferability of CalibSFT across model architectures, we train Gemma4-E2B (Abd et al., 2026) using RLCR with CalibSFT, and then evaluate it on three ID and three OOD benchmarks spanning a range of difficulty. As shown in Table 4, CalibSFT improves both ECE and AUROC on all benchmarks, reducing the average ECE from 23.41 to 12.49 and raising the average AUROC from 70.49 to 73.95. These results indicate that the calibration benefit of CalibSFT holds across model families on benchmarks of varying difficulty.

6 Conclusion

In this work, we address the challenge of verbalized overconfidence in large reasoning models. We identify that concentrated confidence priors restrict exploration during confidence-aware RL, and propose CalibSFT to shape a well-calibrated prior with broad support before policy optimization. CalibSFT derives confidence targets from self-sampled rollouts that provably preserve proper-scoring optimality, and supervises confidence without imitating incorrect reasoning. Experimental results across 16 benchmarks confirm that CalibSFT improves calibration and discrimination across diverse RL frameworks, establishing a foundation for reliable and trustworthy reasoning models.

AI USE STATEMENT

In this work, we used generative AI tools for text polishing, grammar correction, and improving language readability. All suggested edits were thoroughly reviewed, verified, and finalized by the authors, who take full responsibility for the contents of this paper.

ETHICS STATEMENT

This work focuses on understanding and mitigating overconfidence in large reasoning language models to improve their reliability and safety. All experiments are conducted on publicly available, standard academic benchmarks and open-weight models, and do not involve human subjects, personally identifiable data, or sensitive applications.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our theoretical and empirical findings, complete details are provided throughout the paper and appendix. For experimental results, the rollout collection process, target-balanced sampling, and full training hyperparameters (e.g., learning rates, batch sizes, and optimization steps) are documented in Appendix C. System prompts for all confidence formats are provided in Table 5, and benchmark grading rules are described in Appendix C. The code will be publicly released upon publication.

REFERENCES

Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998.

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

David Bani-Harouni, Chantal Pellegrini, Paul Stangel, Ege Özsoy, Kamilia Zaripova, Nassir Navab, and Matthias Keicher. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, 2013.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Jon-Paul Cacioli. Verbal confidence saturation in 3–9B open-weight instruction-tuned LLMs: A pre-registered psychometric validity screen. *arXiv preprint arXiv:2604.22215*, 2026.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*, 2025.

Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

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.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Yuyang Dai and Yuxia Wang. Rescaling confidence: What scale design reveals about LLM metacognition. *arXiv preprint arXiv:2603.09309*, 2026.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training LMs to reason about their uncertainty. In *International Conference on Learning Representations*, 2026.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond Benchmarks: MathArena as an Evaluation Platform for Mathematics with LLMs. In *3rd AI for Math Workshop: Toward Self-Evolving Scientific Agents, ICML Workshop AI4Math*, 2026.

Maksym Del, Markus K"angsepp, Marharyta Domnich, Ardi Tampuu, Lisa Yankovskaya, Meelis Kull, and Mark Fishel. How uncertainty estimation scales with sampling in reasoning models. *arXiv preprint arXiv:2603.19118*, 2026.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2368–2378, 2019.

Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Aliai Eusebi, Alexander Herzog, Xiaoyu Liang, Marie Vasek, Enrico Mariconti, and Lorenzo Cavallaro. Reading calibrated uncertainty from language model trajectories. *arXiv preprint arXiv:2605.22864*, 2026.

Conor Finlay, Joshua Kurien, Saurabh Dash, Marzieh Fadaee, and Beyza Ermis. CALIBER: Calibrating Confidence Before and After Reasoning in Language Models. *arXiv preprint arXiv:2606.24281*, 2026.

Shivam Garg, Samuele Tosatto, Yangchen Pan, Martha White, and A. Rupam Mahmood. An alternate policy gradient estimator for softmax policies. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pp. 6630–6689. PMLR, 2022.

Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. In *International Conference on Learning Representations*, 2019.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 conference of the north American chapter of the association for computational linguistics: human language technologies (volume 1: long papers)*, pp. 6577–6595, 2024.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330, 2017.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Sophia Hager, David Mueller, Kevin Duh, and Nicholas Andrews. Uncertainty distillation: Teaching language models to express semantic confidence. *arXiv preprint arXiv:2503.14749*, 2025.

Sai Hao, Hao Zeng, Hongxin Wei, and Bingyi Jing. RACER: Risk-aware calibrated efficient routing for large language models. *arXiv preprint arXiv:2603.06616*, 2026.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. *arXiv preprint arXiv:2402.14008*, 2024.

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 *Advances in Neural Information Processing Systems*, 2021.

Huipeng Huang, Wenbo Liao, Huajun Xi, Hao Zeng, Mengchen Zhao, and Hongxin Wei. Model-agnostic selective labeling with provable statistical guarantees. *arXiv preprint arXiv:2510.14581*, 2025.

Xinmeng Huang, Shuo Li, Mengxin Yu, Matteo Sesia, Hamed Hassani, Insup Lee, Osbert Bastani, and Edgar Dobriban. Uncertainty in language models: Assessment through rank-calibration. *arXiv preprint arXiv:2404.03163*, 2024.

Chaeyun Jang, Moonseok Choi, Yegon Kim, Hyungi Lee, and Juho Lee. Verbalized confidence triggers self-verification: Emergent behavior without explicit reasoning supervision. *arXiv preprint arXiv:2506.03723*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Yuqing Kong, Mingyu Song, Yizhou Wang, and Yifan Wu. Calibration without ground truth. *arXiv preprint arXiv:2601.19862*, 2026.

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

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, 2019.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in LLMs: Reward calibration in RLHF. In *International Conference on Learning Representations*, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, 2022.

Changcheng Li, Jiancan Wu, Hengheng Zhang, Zhengsu Chen, Guo An, Junxiang Qiu, Xiang Wang, and Qi Tian. Confidence Before Answering: A Paradigm Shift for Efficient LLM Uncertainty Estimation. *arXiv preprint arXiv:2603.05881*, 2026a.

Chen Li, Xiaoling Hu, Songzhu Zheng, Jiawei Zhou, and Chao Chen. ORCE: Order-aware alignment of verbalized confidence in large language models. *arXiv preprint arXiv:2605.12446*, 2026b.

Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. ConfTuner: Training large language models to express their confidence verbally. *arXiv preprint arXiv:2508.18847*, 2025.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Shudong Liu, Zhaocong Li, Xuebo Liu, Runzhe Zhan, Derek F. Wong, Lidia S. Chao, and Min Zhang. Can LLMs learn uncertainty on their own? expressing uncertainty effectively in a self-training manner. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21635–21645, 2024a.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6107–6117, 2025.

Xin Liu, Muhammad Khalifa, and Lu Wang. LitCab: Lightweight language model calibration over short- and long-form responses. In *International Conference on Learning Representations*, 2024b.

Beier Luo, Shuoyuan Wang, Sharon Li, and Hongxin Wei. Your pre-trained LLM is secretly an unsupervised confidence calibrator. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-Preview with a 1.5B Model by Scaling RL, 2025b. Notion Blog.

Zhengzhao Ma, Xueru Wen, Boxi Cao, Yaojie Lu, Hongyu Lin, Jinglin Yang, Min He, Xianpei Han, and Le Sun. Decoupling reasoning and confidence: Resurrecting calibration in reinforcement learning from verifiable rewards. In *Forty-third International Conference on Machine Learning*, 2026.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International conference on machine learning*, pp. 6820–6829. PMLR, 2020.

Zhiting Mei, Christina Zhang, Tenny Yin, Justin Lidard, Ola Shorinwa, and Anirudha Majumdar. Reasoning about uncertainty: Do reasoning models know when they don't know? *arXiv preprint arXiv:2506.18183*, 2025.

Shiyu Ni, Keping Bi, Lulu Yu, and Jiafeng Guo. Are large language models more honest in their probabilistic or verbalized confidence? *arXiv preprint arXiv:2408.09773*, 2024.

Ki Jung Seo, Sehun Lim, and Taeuk Kim. ADVICE: Answer-dependent verbalized confidence estimation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23942–23962. Association for Computational Linguistics, 2026.

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.

Jianru Shen. Provable limits and certified deferral for verbalized uncertainty in small language models. *arXiv preprint arXiv:2608.05064*, 2026.

Kaiwen Shi, Zheyuan Zhang, and Yanfang Ye. SAGE: Answer-conditioned uncertainty targets for verbal uncertainty alignment. *arXiv preprint arXiv:2606.11512*, 2026.

Ponhvoan Srey, Xiaobao Wu, Cong-Duy Nguyen, Quang Minh Nguyen, Duc Anh Vu, and Anh Tuan Luu. From signals to transfer: A factorised study of probe-based uncertainty estimation in large language models. *arXiv preprint arXiv:2606.27679*, 2026.

Chee Heng Tan, Zhuoyi Lin, Mehul Motani, and Wee Sun Lee. On the effectiveness of reward functions in reinforcement learning for confidence calibration of large language models. *arXiv preprint arXiv:2607.04332*, 2026.

Linwei Tao, Yi-Fan Yeh, Minjing Dong, Tao Huang, Philip Torr, and Chang Xu. Revisiting uncertainty estimation and calibration of large language models. *arXiv preprint arXiv:2505.23854*, 2025.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in LLMs. *arXiv preprint arXiv:2502.06233*, 2025.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*, 2023.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Liaoyaqi Wang, Chunsheng Zuo, William Jurayj, Benjamin Van Durme, and Anqi Liu. Process supervision of confidence margin for calibrated LLM reasoning. In *Conference on Language Modeling*, 2026.

Victor Wang and Elias Stengel-Eskin. Calibrating verbalized confidence with self-generated distractors. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Colin White, Samuel Dooley, Manley Roberts, et al. LiveBench: A challenging, contamination-limited LLM benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 91595–91631, 2025.

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

Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. SaySelf: Teaching LLMs to express confidence with self-reflective rationales. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5985–5998, 2024.

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

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for LLMs. *arXiv preprint arXiv:2412.14737*, 2024.

Sin-Han Yang, Cheng-Kuang Wu, Chieh-Yen Lin, Yun-Nung Chen, Hung-yi Lee, and Shao-Hua Sun. On calibration of large language models: From response to capability. *arXiv preprint arXiv:2602.13540*, 2026a.

Xuqing Yang, Yi Yuan, Shanzhe Lei, and Xuhong Wang. Scaling with Confidence: Calibrating Confidence of LLMs for Adaptive Test Time Scaling. *arXiv preprint arXiv:2607.01612*, 2026b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Yi-Fan Yeh, Linwei Tao, Minjing Dong, Tao Huang, Jialin Yu, Philip Torr, and Chang Xu. Retrieval-augmented linguistic calibration. *arXiv preprint arXiv:2605.19344*, 2026.

Chengyao Yu, Hao Zeng, Youxin Zhu, Jianguo Huang, Huajun Zeng, and Bingyi Jing. Anytime safe PAC efficient reasoning. *International Conference on Machine Learning*, 2026a.

Fengfei Yu, Ruijia Niu, Dongxia Wu, Yian Ma, and Rose Yu. Calibrating LLMs with semantic-level reward. *arXiv preprint arXiv:2605.15588*, 2026b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Hao Zeng, Jianguo Huang, Bingyi Jing, Hongxin Wei, and Bo An. On the provable performance guarantee of efficient reasoning models. In *The First Workshop on Efficient Spatial Reasoning, ICLR 2026 Workshop ES-Reasoning*, 2026.

Caiqi Zhang, Xiaochen Zhu, Chengzu Li, Nigel Collier, and Andreas Vlachos. LoVeC: Reinforcement learning for better verbalized confidence in long-form generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33336–33363. Association for Computational Linguistics, 2026a.

Chuang Zhang, Zizhen Zhu, Yihao Wei, Bing Tian, Junyi Liu, Henan Wang, Xavier Wang, and Yaxiao Liu. Confidence-calibrated small-large language model collaboration for cost-efficient reasoning. *arXiv preprint arXiv:2603.03752*, 2026b.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R. Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 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*, 2024.

APPENDIX

A Related Work **16**

B Theoretical Analysis **17**

B.1 Proof of Proposition 1 17

B.2 Proof of Proposition 2 18

B.3 Proof of Proposition 3 19

C Implementation Details **20**

C.1 Training Pipelines 20

C.2 Evaluation Protocol 21

C.3 Confidence Formats 22

D Detailed Experimental Results **22**

D.1 Training Dynamics 22

D.2 Per-Dataset Main Results 22

D.3 Robustness across Random Seeds 26

D.4 Evaluation on Additional Metrics 26

D.5 Ablation Studies 28

D.6 Integration with Other Confidence Estimators 31

D.7 Confidence Visualization 31

D.8 Running Time Analysis 33

E Downstream Applications **33**

E.1 Selective Risk Control 33

E.2 Confidence-Guided Model Routing 35

A RELATED WORK

Confidence estimation in LLMs. Confidence estimates for a model answer can be derived from many signals, of which four families cover most existing methods and differ in their model-access requirements and inference costs (Geng et al., 2024; Liu et al., 2025). The most direct signal is the probability the model assigns to its own answer tokens, used either directly or after recalibration by a layer fitted on labeled data (Kadavath et al., 2022; Liu et al., 2024b). Probing methods instead train a classifier on the hidden states or on the layer-wise trajectory that produces the answer (Azaria & Mitchell, 2023; Eusebi et al., 2026; Srey et al., 2026), which requires access to activations rather than logits. When a model exposes neither, sampling-based methods recover confidence from behavior alone and measure how far several sampled answers agree (Kuhn et al., 2023; Del et al., 2026; Taubenfeld et al., 2025), at the cost of repeated generations per question. Verbalized confidence removes that cost as well: the model states a numerical or linguistic score alongside its answer within a single generation (Lin et al., 2022; Tian et al., 2023; Xiong et al., 2024; Yang et al., 2024). We study the verbalized setting for reasoning models, where the score follows a reasoning trace and the final answer. Such a score is informative only once it matches accuracy: the values an uncalibrated base model reports concentrate on a few points of the scale (Dai & Wang, 2026; Cacioli, 2026) and separate its errors less well than its own token probabilities do (Ni et al., 2024; Tao et al., 2025).

Confidence calibration for LLMs. Calibration requires the reported confidence to match the empirical probability that the answer is correct, and a method either rescales a score after the model has produced it or changes the model that produces it. Rescaling is the standard treatment for probability-based signals: a temperature or a bias layer is fitted on a labeled held-out set (Guo et al., 2017; Liu et al., 2024b), and label-free variants align the score with a reference model or project the two predictive distributions onto a calibrated one (Luo et al., 2025a; Kong et al., 2026). Probes need no separate step, since they are classifiers already trained on correctness labels (Azaria & Mitchell, 2023; Srey et al., 2026), and a sampling-based estimate is calibrated by choosing how agreement among samples maps to a probability (Kuhn et al., 2023; Del et al., 2026). A verbalized score can be rescaled the same way, but only through extra queries per question or an extra pass that rewrites the reported value (Wang & Stengel-Eskin, 2026; Yeh et al., 2026), and neither changes how the model generates the score in the first place.

Post-training for verbalized confidence. Post-training directly optimizes models to express confidence alongside their answers. Supervised approaches train on constructed confidence targets, while reinforcement learning uses rewards that encourage reliable confidence estimates. SFT methods derive these targets from correctness signals on the model's own responses (Lin et al., 2022; Zhang et al., 2024; Jang et al., 2025; Liu et al., 2024a; Xu et al., 2024; Hager et al., 2025; Seo et al., 2026; Zhang et al., 2026a; Li et al., 2025). Reinforcement learning adds a calibration term to the reward of RLVR, which by itself optimizes answer correctness and leaves the confidence unsupervised (Shao et al., 2024; Yu et al., 2025); the term is a proper scoring rule on the reported confidence, and the methods differ in where the score is placed, what it is trained toward, and how credit reaches its tokens (Leng et al., 2025; Damani et al., 2026; Bani-Harouni et al., 2026; Tan et al., 2026; Li et al., 2026a; Finlay et al., 2026; Shi et al., 2026; Yu et al., 2026b; Li et al., 2026b; Ma et al., 2026; Wang et al., 2026; Yang et al., 2026b; Zhang et al., 2026b). However, on-policy confidence-aware RL relies on confidence values sampled from the current policy. Our analysis in Section 3 shows how a concentrated confidence prior can restrict this exploration. CalibSFT addresses this limitation by shaping the confidence prior through supervised fine-tuning before confidence-aware RL.

B Theoretical Analysis

B.1 Proof of Proposition 1

Proof. Fix the context representation h and the corresponding expected rewards $u_i(h)$, where $p_h = \Pr(Z = 1 \mid h)$. Expanding the squared error gives:

$$u_i(h) = -\mathbb{E}[(c_i - Z)^2 \mid h] = -\left[p_h(1-p_h) + (c_i - p_h)^2\right]. \tag{11}$$

Since $c_i, p_h \in [0,1]$, the reward spread satisfies $\Delta_h = \max_i u_i(h) - \min_i u_i(h) = \max_i (c_i - p_h)^2 - \min_i (c_i - p_h)^2 \le 1$.

Differentiating $J_h(\eta) = \sum_{j=1}^K \pi_j u_j(h)$ with respect to logit η_i, the softmax derivative yields $\partial \pi_j / \partial \eta_i = \pi_j(\mathbf{1}\{j = i\} - \pi_i)$. It follows that

$$\frac{\partial J_h}{\partial \eta_i} = \sum_{j=1}^K u_j(h)\pi_j(\mathbf{1}\{j=i\} - \pi_i) \tag{12}$$

$$= \pi_i u_i(h) - \pi_i \sum_{j=1}^K \pi_j u_j(h) \tag{13}$$

$$= \pi_i\big(u_i(h) - J_h\big). \tag{14}$$

Since J_h is a convex combination of $\{u_j(h)\}_{j=1}^K$, we have $\min_j u_j(h) \le J_h \le \max_j u_j(h)$, which implies $|u_i(h) - J_h| \le \Delta_h$. This establishes the coordinate-wise bound:

$$\left|\frac{\partial J_h}{\partial \eta_i}\right| \le \pi_i \Delta_h. \tag{15}$$

If the policy concentrates on a dominant action k such that $\pi_k = 1 - \varepsilon$, then

$$|u_k(h) - J_h| = \left| \sum_{i \neq k} \pi_i \big(u_k(h) - u_i(h)\big) \right| \leq \sum_{i \neq k} \pi_i \Delta_h = \varepsilon \Delta_h. \tag{16}$$

Consequently, the L_1 norm of the gradient satisfies:

$$\|\nabla_\eta J_h\|_1 = \pi_k |u_k(h) - J_h| + \sum_{i \neq k} \pi_i |u_i(h) - J_h| \tag{17}$$

$$\leq (1 - \varepsilon)\varepsilon \Delta_h + \varepsilon \Delta_h \tag{18}$$

$$= (2 - \varepsilon)\varepsilon \Delta_h \leq 2\varepsilon \Delta_h \leq 2\varepsilon. \tag{19}$$

□

Connection to confidence-aware RL. Under $C \perp Z \mid h$, the gradient is the expectation of the likelihood-ratio estimator $U \nabla_\eta \log \pi(I \mid h)$, where $I \sim \pi(\cdot \mid h)$ and $U = -(c_I - Z)^2$. In practical algorithms such as RLCR, the total reward is $Z - (c_I - Z)^2$. The label term Z has conditional expectation $\mathbb{E}[Z \mid h] = p_h$, which is independent of policy parameters η and contributes zero gradient. Subtracting any baseline $b(h)$ that does not depend on the sampled action I leaves this expectation unchanged. GRPO's group-mean baseline includes the response's own reward, which scales the expected gradient by $(G - 1)/G$ without changing its direction. Its standard-deviation normalization further introduces a data-dependent scale $1/(\sigma_R + \epsilon)$; treating this scale as fixed within a group, the suppression bound holds up to a positive factor.

Moving beyond a concentrated prior. Under the suppression bound, each update moves the logit of a suboptimal dominant value c_k by at most $\mathcal{O}(\varepsilon)$. Since policy entropy tends to decrease during RL for reasoning models (Cui et al., 2025), and confidence token entropy remains low throughout our RL runs (Figure 7), ε stays small and the policy remains concentrated on a few confidence values within the training budget.

B.2 Proof of Proposition 2

Proof. Recall that $p_h = \Pr(Z = 1 \mid h)$.

Decomposing the error into $C - Z = (C - p_h) - (Z - p_h)$, conditional independence yields:

$$\mathbb{E}[(C - p_h)(Z - p_h) \mid h] = \mathbb{E}[C - p_h \mid h] \cdot \mathbb{E}[Z - p_h \mid h] = 0, \tag{20}$$

since $\mathbb{E}[Z \mid h] = p_h$. Expanding the squared error therefore gives:

$$\mathbb{E}[(C - Z)^2 \mid h] = \mathbb{E}[(Z - p_h)^2 \mid h] + \mathbb{E}[(C - p_h)^2 \mid h] \tag{21}$$

$$= p_h(1 - p_h) + \mathbb{E}[(C - p_h)^2 \mid h]. \tag{22}$$

For any $\delta > 0$, the second term satisfies:

$$\mathbb{E}[(C - p_h)^2 \mid h] \geq \mathbb{E}\big[(C - p_h)^2 \cdot \mathbf{1}\{|C - p_h| \geq \delta\} \mid h\big] \tag{23}$$

$$\geq \delta^2 \Pr(|C - p_h| \geq \delta \mid h) \tag{24}$$

$$= \delta^2 \big(1 - m_\delta(h)\big), \tag{25}$$

where $m_\delta(h) = \Pr(|C - p_h| < \delta \mid h)$. Substituting this back into the equality yields:

$$\mathbb{E}[(C - Z)^2 \mid h] \geq p_h(1 - p_h) + \delta^2 \big(1 - m_\delta(h)\big). \tag{26}$$

Thus, assigning little probability mass near p_h incurs an excess Brier loss even when confidence values in that neighborhood remain possible. □

Restricted confidence support. As a special case, suppose $\Pr(C \in S_h \mid h) = 1$ for a nonempty set $S_h \subseteq [0,1]$, and define $d_h = \mathrm{dist}(p_h, S_h) = \inf_{c \in S_h} |c - p_h|$. If $d_h > 0$, then $m_{d_h}(h) = 0$, and Proposition 2 with $\delta = d_h$ gives:

$$\mathbb{E}[(C - Z)^2 \mid h] \geq p_h(1 - p_h) + \mathrm{dist}(p_h, S_h)^2. \tag{27}$$

For $d_h = 0$, the same bound follows directly from the decomposition above. If S_h is closed, a predictor that always outputs a nearest point in S_h attains the bound. When verbalized confidence values concentrate on a small discrete set of high values (Figure 1), $\mathrm{dist}(p_h, S_h)$ measures the distance from the nearest available value to the true posterior p_h, directly translating the concentrated prior into excess Brier risk.

Best-of-G sampling under a shared context. GRPO samples one confidence value per response context. As a more favorable case for exploration, draw G independent confidence samples $C_1, \ldots, C_G$ from $\pi(\cdot \mid h)$ for the same context h. Each sample lies outside $\{c : |c - p_h| < \delta\}$ with probability $1 - m_\delta(h)$. Since the samples are independent given h, all G samples do so with probability $\big(1 - m_\delta(h)\big)^G$. On that event, even the sample closest to p_h has squared error at least δ^2, yielding:

$$\mathbb{E}\left[\min_{i \leq G}(C_i - p_h)^2 \,\middle|\, h\right] \geq \delta^2\big(1 - m_\delta(h)\big)^G. \tag{28}$$

Since a fixed value c has expected Brier loss $p_h(1 - p_h) + (c - p_h)^2$, this bounds the excess Brier loss of the best of G samples. With $G = 8$, the group size in our RL runs, the G samples contain a value within δ of p_h with probability only $1 - \big(1 - m_\delta(h)\big)^8$, which remains small when $m_\delta(h) \approx 0$. Thus, even repeated sampling from the same context rarely reaches values near p_h.

B.3 Proof of Proposition 3

We analyze the unrounded targets under the rollout distribution before filtering and balancing.

Proof. Fix x and denote $v_x = p_x(1 - p_x)$. Under independent rollout sampling, $\mathbb{E}[Q_x \mid x] = \mathbb{E}[Z_i \mid x] = p_x$ and $\mathrm{Var}(Q_x \mid x) = v_x/n$. Since Z_i is included in the summation of Q_x, their covariance is $\mathrm{Cov}(Q_x, Z_i \mid x) = v_x/n$. By linearity of expectation:

$$\mathbb{E}[C_i^* \mid x] = \lambda \mathbb{E}[Q_x \mid x] + (1 - \lambda)\mathbb{E}[Z_i \mid x] = p_x. \tag{29}$$

Expanding the conditional variance yields:

$$\mathrm{Var}(C_i^* \mid x) = \lambda^2 \frac{v_x}{n} + (1 - \lambda)^2 v_x + 2\lambda(1 - \lambda)\frac{v_x}{n} \tag{30}$$

$$= v_x \left[\frac{1}{n} + (1 - \lambda)^2 \left(1 - \frac{1}{n}\right)\right]. \tag{31}$$

Now fix any prediction $\hat{c}$ at which ℓ is finite. Since $\ell(\hat{c}, t)$ is affine in t, taking expectations gives:

$$\mathbb{E}[\ell(\hat{c}, C_i^*) \mid x] = p_x \ell(\hat{c}, 1) + (1 - p_x)\ell(\hat{c}, 0) = \mathbb{E}[\ell(\hat{c}, Z_i) \mid x]. \tag{32}$$

Because ℓ is strictly proper, $\mathbb{E}[\ell(\hat{c}, Z_i) \mid x]$ is uniquely minimized at $\hat{c} = p_x$. Therefore, the mixed target preserves p_x as the unique population minimizer under any strictly proper scoring loss. □

Variance reduction and discrimination margin. At $\lambda = 0$, the target reduces to the binary label with maximum variance v_x. At $\lambda = 1$, the target is the question success rate with minimum variance v_x/n, but provides zero within-question discrimination. For any $\lambda \in (0, 1)$, the variance is strictly smaller than v_x. Within any observed rollout group, because q_x is shared across all responses, the target difference between a correct response i ($z_i = 1$) and an incorrect response j ($z_j = 0$) is exactly:

$$c_i^* - c_j^* = (1 - \lambda)(z_i - z_j) = 1 - \lambda. \tag{33}$$

Furthermore, taking expectation over the rollout sampling, the expected target separation between correct and incorrect responses satisfies $\mathbb{E}[C_i^* \mid Z_i = 1, x] - \mathbb{E}[C_i^* \mid Z_i = 0, x] = 1 - \lambda + \frac{\lambda}{n} \equiv \alpha$. With $n = 50$ and $\lambda = 0.5$, the within-group margin is 0.50 and the expected margin is $\alpha \approx 0.51$, so incorrect responses receive clearly lower targets, while the target variance drops from v_x to about $0.265 v_x$, a reduction of over 73% relative to binary labels.

Conditioning on response context and shrinkage. Proposition 3 establishes question-level calibration under the rollout distribution. When conditioned on a specific response context $h_i = \phi(x, r_i, a_i)$, let $p_{h_i} = \Pr(Z_i = 1 \mid h_i)$. Assuming remaining rollouts are conditionally independent of h_i given x, the expected target satisfies:

$$\mathbb{E}[C_i^* \mid h_i] = \alpha\, p_{h_i} + (1 - \alpha)\, p_x, \tag{34}$$

where α is the expected margin defined above. Eq. equation 34 shows that the target shrinks the noisy single-sample correctness signal toward the question-level success rate p_x. This shrinkage introduces a response-level bias of $(1 - \alpha)(p_x - p_{h_i})$ in exchange for lower target variance, while keeping a margin of $1 - \lambda$ between correct and incorrect responses within each group. These theoretical results characterize the supervision signal under the rollout distribution; the empirical effects of subsequent target balancing and autoregressive fine-tuning are evaluated in Section 5.2 and Appendix D.5.

C Implementation Details

C.1 Training Pipelines

We use Qwen3-8B as the default base model throughout the experiments. For DeepScaleR, we remove duplicate questions and questions with conflicting reference answers, resulting in 38,915 unique questions. We randomly partition these questions into a held-out evaluation set of 2,000 questions (DeepScaleR-Eval) and a training set of 36,915 questions (DeepScaleR-Train), where the latter is used to construct both the SFT dataset for CalibSFT and the downstream RL training data.

SFT data construction. We sample $n = 50$ responses per question from DeepScaleR-Train with the base model, using temperature 1.0, top-p 0.95, and a maximum length of 8,192 tokens. This yields 1,845,750 responses in total. Before filtering, we compute the empirical success rate q_x for each question from all 50 responses. We then discard responses that fail to follow the `<think>`/`<answer>`/`<confidence>` format or exceed the token limit without finishing. For each retained response, we compute the confidence target $c_i^* = \lambda q_x + (1 - \lambda) z_i$ with $\lambda = 0.5$ and replace the original confidence value with c_i^*, while keeping reasoning trace and answer unchanged. In the default probability format, each target is rounded to two decimal places within the `<confidence>` tag, producing 100 distinct target values. To perform target-balanced sampling, we group examples by their target value and uniformly sample an equal number of responses from each group, with the number set by the smallest group. This yields a balanced dataset containing 349 examples per target, totaling 34,900 examples across 11,539 unique questions for main experiments.

CalibSFT. We fine-tune the model on the balanced set using correctness-conditional supervision with $\beta = 0.5$, where correct responses supervise both the reasoning/answer and confidence tokens, while incorrect responses supervise only the confidence tokens. The optimizer is AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, zero weight decay). We employ a constant learning rate of 2×10^{-6} after 10 warm-up steps, with a global batch size of 256 sequences truncated to 8,192 tokens. The model is trained for 180 optimization steps on 8 NVIDIA B20Z GPUs.

Confidence-aware RL. For downstream reinforcement learning, each prompt group contains 8 rollouts, with each response capped at 8,192 tokens. We train the policy with a batch size of 64 prompt groups for 160 optimization steps. For methods using DAPO dynamic sampling, which keeps only groups with mixed correctness and discards groups that are entirely correct or incorrect, this corresponds to approximately 600 generation batches. To accelerate training for these dynamic-sampling methods, we pre-filter the training set by removing questions where all base-model rollouts are already correct. Each RL baseline follows its designated paradigm while preserving its original reward formulation and credit-assignment mechanism:

- **Joint RL baselines (RLCR, CoCA, DCPO, and C3RL).** These methods simultaneously optimize answer correctness and confidence rewards starting directly from the base checkpoint (Base $\rightarrow$ RL), whereas their CalibSFT counterparts initialize from the fine-tuned prior (Base $\rightarrow$ CalibSFT $\rightarrow$ RL). For RLCR (Damani et al., 2026), CoCA (Li et al., 2026a), and DCPO (Ma et al., 2026), training uses the default probability format along with DAPO dynamic sampling. In contrast, C3RL (Yang et al., 2026b) integrates reference-accuracy rewards using an integer confidence format (0–9),

Table 5: System prompts for the three confidence formats.

Format	System prompt
Probability	A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and analyzes its confidence about the solution and then provides the user with the final answer as well as its confidence level. The confidence level indicates how certain the Assistant is about its answer, expressed as a decimal between 0 and 1 with exactly two decimal places, enclosed within <confidence> </confidence> tags. The response must strictly follow this format: <think> reasoning process here </think> <answer> final short answer only </answer> <confidence> 0.xx </confidence>. The <answer> tag must contain only the final answer string needed for exact-match evaluation, not a full sentence, explanation, or reasoning.
Integer	A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and analyzes its confidence about the solution and then provides the user with the final answer as well as its confidence score. The confidence score indicates how certain the Assistant is about its answer, expressed as a single integer from 0 to 9, enclosed within <confidence> </confidence> tags. A score of 0 means very low confidence and the answer is likely incorrect. A score of 9 means very high confidence and the answer is very likely correct. The response must strictly follow this format: <think> reasoning process here </think> <answer> final short answer only </answer> <confidence>X</confidence>, where X is one of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. The <answer> tag must contain only the final answer string needed for exact-match evaluation, not a full sentence, explanation, or reasoning.
Linguistic	A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and analyzes its confidence about the solution and then provides the user with the final answer as well as its confidence level. The confidence level indicates how certain the Assistant is about its answer, expressed as exactly one of low, medium, or high, enclosed within <confidence> </confidence> tags. Low means the answer is likely incorrect, medium means the answer is uncertain, and high means the answer is likely correct. The response must strictly follow this format: <think> reasoning process here </think> <answer> final short answer only </answer> <confidence>label</confidence>. The <answer> tag must contain only the final answer string needed for exact-match evaluation, not a full sentence, explanation, or reasoning.

and its CalibSFT initialization is adapted accordingly. To preserve C3RL's category-dependent reference signals (all-correct, partially-correct, or all-incorrect), we train both the baseline and CalibSFT runs on the full dataset without dynamic sampling.

- **Two-stage RL baseline (ReDoubt).** Following Bani-Harouni et al. (2026), ReDoubt decouples accuracy and confidence by optimizing a clipped log-score confidence objective on top of an accuracy-aligned policy (Base $\rightarrow$ RLVR $\rightarrow$ ReDoubt). To seamlessly integrate with this paradigm, we fine-tune the trained RLVR checkpoint with CalibSFT before running ReDoubt RL (Base $\rightarrow$ RLVR $\rightarrow$ CalibSFT $\rightarrow$ ReDoubt).

C.2 Evaluation Protocol

For our primary evaluations, we decode at temperature 0.6 and sample 32 responses per question for the AIME benchmarks and 4 responses per question for all other benchmarks. Pass@1 is reported as the average answer correctness across these sampled outputs. To evaluate calibration, the predicted confidence value is parsed from the <confidence> tag and normalized to $[0, 1]$, with ECE calculated using 10 equal-width bins. Unless stated otherwise, all RL evaluations report the final checkpoint.

Answer verification. The verification rules yield the binary correctness labels z_i that construct the CalibSFT targets and provide verifiable rewards during downstream RL. Depending on the benchmark protocol, we use three types of verification:

- **Symbolic and normalized matching:** Mathematical benchmarks and HotpotQA verify correctness via symbolic equivalence under `math_verify` or exact text match after standard normalization (lowercasing, removing punctuation/articles, and collapsing whitespace). NQOpen and MuSiQue apply the same normalization against any annotated gold alias.
- **Official benchmark protocols:** For datasets with established grading rules, we strictly follow their official scripts. WebQuestions requires an exact set match, DROP applies its official normalization for exact string matching, PopQA accepts predictions containing a gold alias, and LiveBenchReasoning employs its native rule-based graders for reasoning subtasks.
- **LLM-as-a-judge:** Since TriviaQA allows open-ended surface variations, we employ Qwen3-8B as an LLM judge. It takes the question, reference answer, and model prediction as input to render a binary correctness decision.

C.3 Confidence Formats

To evaluate format generality in Section 5.4, we keep the training data fixed and only alter how the confidence target c^* is expressed. The corresponding prompts for each format are provided in Table 5.

- **Probability:** Expressed as a two-decimal float. The fine grid with step 0.01 is our default format, while the coarse grid with step 0.05 is evaluated as an ablation.
- **Integer:** Mapped to a single digit $\lfloor 9c^* + 0.5 \rfloor \in \{0, \ldots, 9\}$, and normalized by dividing by 9 during evaluation.
- **Linguistic:** Categorized into *low* ($c^* < 1/3$), *medium* ($1/3 \leq c^* < 2/3$), and *high* ($c^* \geq 2/3$), which are evaluated at fixed values of 0.25, 0.50, and 0.75, respectively.

D Detailed Experimental Results

D.1 Training Dynamics

In Section 3, we observed that confidence remains concentrated on a few high values when training RLCR with its calibration reward. To examine whether this limitation is universal, we further train Qwen3-8B with CoCA (Li et al., 2026a) and DCPO (Ma et al., 2026), tracking their training trajectories with and without CalibSFT initialization.

CalibSFT mitigates the calibration–discrimination trade-off. As illustrated in Figure 6, this calibration-discrimination tension is indeed a shared failure mode across standard confidence-aware RL methods. Across all three baselines, optimizing the calibration reward successfully lowers calibration error (e.g., ECE drops from 23.8–25.8% to 7.8–14.2%), yet AUROC steadily degrades (falling from 70.8–73.5% to 64.1–68.0%). Consequently, the policy merely minimizes the aggregate calibration penalty without learning to distinguish correct from incorrect answers. In contrast, initializing policy optimization with CalibSFT sustains lower Brier score and superior AUROC throughout training while preserving task accuracy.

CalibSFT broadens confidence exploration during RL. Figure 7 tracks the corresponding confidence exploration. Without CalibSFT, confidence-token entropy remains low, and each question produces fewer than 5.3 distinct confidence values on average across the eight rollouts. With so few distinct values, incorrect responses rarely receive low confidence during rollouts, leaving the policy gradient little signal to lower their confidence. CalibSFT increases token entropy by more than threefold and roughly doubles the number of distinct confidence values sampled per question throughout training. The resulting broader support is consistent with the improved discrimination shown in Figure 6.

D.2 Per-Dataset Main Results

Tables 6 and 7 present the comprehensive results across all 16 individual benchmarks, complementing the split-level averages in Table 1. The overall average across the 8 datasets within each split is summarized in the final subtable.

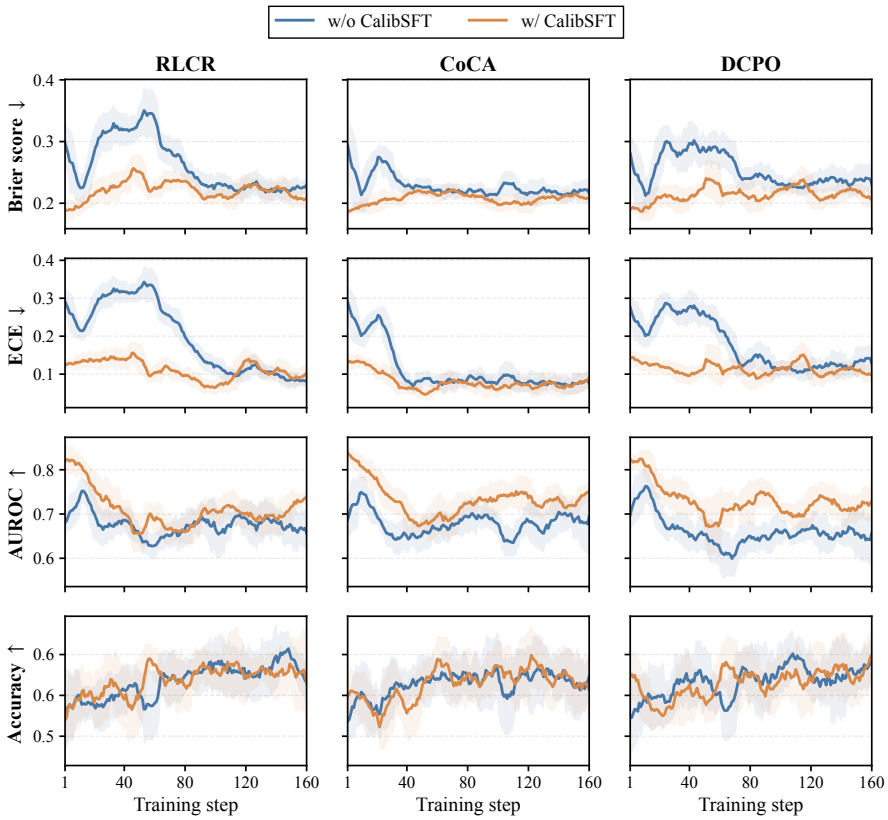

Figure 6: Training dynamics of RLCR, CoCA, and DCPO with and without CalibSFT initialization. Standard RL improves Brier score and ECE but degrades AUROC over training. In contrast, CalibSFT consistently achieves markedly better Brier score, ECE, and AUROC while preserving task accuracy.

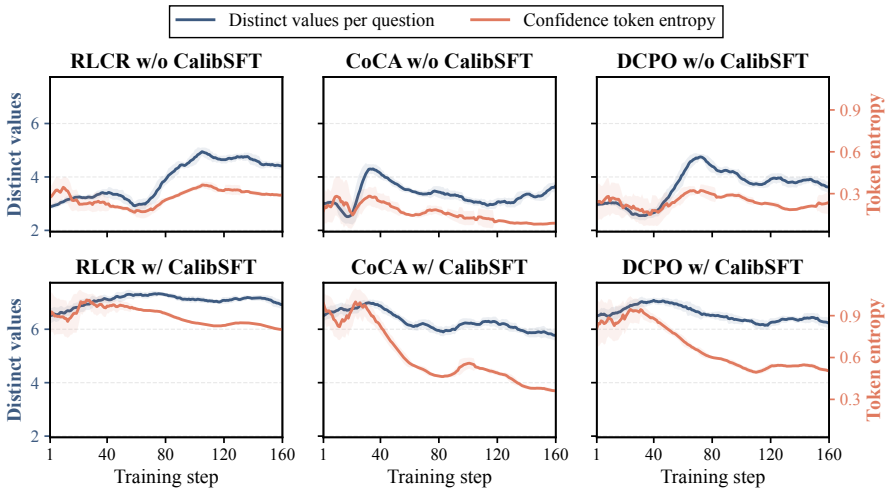

Figure 7: Confidence exploration dynamics during training across methods. The number of distinct confidence values sampled per question (left axis) closely tracks confidence token entropy (right axis). In standard RL (top), confidence token entropy remains low, restricting the number of distinct confidence values sampled per question. In contrast, CalibSFT initialization (bottom) substantially elevates token entropy throughout policy optimization, doubling distinct sampled values and enabling active exploration across the confidence spectrum.

Table 6: Per-dataset results on eight in-domain mathematical benchmarks spanning a range of difficulty. Each entry reports Pass@1, AUROC, Brier score, and ECE. CalibSFT initialization improves calibration and discrimination on most datasets, with the largest gains on benchmarks where baseline RL is heavily overconfident.

(a) DeepScaleR-Eval

Method	P@1	AUROC	Brier	ECE
Base	61.00	66.38	33.94	33.90
RLVR	**70.26**	67.94	26.85	26.26
CalibSFT	61.30	**76.39**	**22.65**	**17.28**
RLCR	71.47	75.45	16.56	5.02
+ CalibSFT	**72.47**	**79.19**	**15.43**	**4.63**
ReDoubt	71.62	75.91	17.12	10.28
+ CalibSFT	**72.42**	**77.98**	**16.07**	**6.71**
CoCA	70.33	73.21	17.69	8.50
+ CalibSFT	**71.54**	**78.46**	**16.45**	**7.76**
DCPO	**71.66**	71.72	17.23	6.36
+ CalibSFT	70.54	**77.54**	**16.16**	**5.87**
C3RL	70.84	62.79	24.50	24.86
+ CalibSFT	**71.25**	**71.52**	**20.08**	**14.85**

(b) MATH-500

Method	P@1	AUROC	Brier	ECE
Base	84.50	65.22	14.18	13.27
RLVR	**89.50**	67.63	**9.80**	**8.41**
CalibSFT	84.00	**81.51**	16.49	14.06
RLCR	**90.80**	76.45	7.42	7.53
+ CalibSFT	90.35	**89.86**	**6.46**	**3.89**
ReDoubt	**90.10**	75.41	9.79	**15.06**
+ CalibSFT	89.70	**89.44**	**9.59**	16.99
CoCA	**90.10**	73.02	10.57	18.22
+ CalibSFT	90.05	**87.05**	**9.08**	**15.69**
DCPO	89.65	70.59	**7.67**	**6.52**
+ CalibSFT	**90.40**	**84.55**	7.71	7.83
C3RL	**89.85**	60.68	**8.97**	**9.30**
+ CalibSFT	88.05	**84.35**	10.47	9.71

(c) MinervaMath

Method	P@1	AUROC	Brier	ECE
Base	47.52	60.84	46.83	47.65
RLVR	**52.85**	60.81	43.34	43.51
CalibSFT	44.94	**66.31**	**25.37**	**17.42**
RLCR	51.75	61.57	29.17	23.05
+ CalibSFT	**51.93**	**73.00**	**22.59**	**12.00**
ReDoubt	**52.30**	60.64	25.36	10.67
+ CalibSFT	51.38	**74.47**	**20.96**	**9.35**
CoCA	49.63	64.28	26.82	18.83
+ CalibSFT	**50.55**	**76.38**	**21.78**	**13.77**
DCPO	**50.64**	63.57	31.72	29.05
+ CalibSFT	50.55	**76.00**	**20.53**	**8.83**
C3RL	50.37	57.50	45.46	46.34
+ CalibSFT	**51.01**	**73.31**	**21.79**	**13.06**

(d) OlympiadBench

Method	P@1	AUROC	Brier	ECE
Base	**61.42**	74.08	32.51	33.33
RLVR	61.39	73.08	34.64	34.90
CalibSFT	60.50	**85.75**	**16.72**	**11.61**
RLCR	63.17	83.46	16.73	**8.48**
+ CalibSFT	**63.28**	**89.00**	**14.60**	9.25
ReDoubt	62.80	81.10	17.14	**7.73**
+ CalibSFT	**63.58**	**90.10**	**14.29**	14.69
CoCA	**61.57**	81.47	18.37	**14.56**
+ CalibSFT	**61.57**	**88.17**	**17.05**	17.07
DCPO	62.24	79.80	18.80	12.97
+ CalibSFT	**62.91**	**87.43**	**14.76**	**8.93**
C3RL	63.06	64.34	30.64	32.22
+ CalibSFT	**63.61**	**80.73**	**18.05**	**15.26**

(e) GSM8K

Method	P@1	AUROC	Brier	ECE
Base	94.39	67.91	5.16	3.35
RLVR	**94.71**	**68.09**	**4.99**	**2.98**
CalibSFT	94.60	66.46	25.93	30.61
RLCR	**94.86**	78.70	4.75	5.64
+ CalibSFT	94.77	**82.24**	**4.65**	**2.81**
ReDoubt	**95.00**	78.24	**7.07**	**16.31**
+ CalibSFT	94.77	**80.01**	7.48	17.69
CoCA	**94.66**	68.97	8.91	21.10
+ CalibSFT	94.64	**71.42**	**8.25**	**18.18**
DCPO	**94.58**	68.17	**5.44**	9.17
+ CalibSFT	94.50	**73.10**	5.82	**7.54**
C3RL	**94.43**	57.33	**5.28**	5.18
+ CalibSFT	94.37	**64.41**	5.30	**5.16**

(f) AIME2024

Method	P@1	AUROC	Brier	ECE
Base	48.23	81.93	39.61	42.50
RLVR	**52.92**	82.05	39.49	41.15
CalibSFT	47.08	**89.04**	**10.35**	**7.73**
RLCR	59.48	**93.59**	13.17	19.70
+ CalibSFT	**64.17**	92.42	**10.97**	**9.79**
ReDoubt	55.62	91.94	15.40	22.77
+ CalibSFT	**61.56**	**93.81**	**11.89**	**14.37**
CoCA	53.85	92.22	**16.27**	**21.37**
+ CalibSFT	**58.54**	**95.41**	16.34	23.56
DCPO	57.92	88.84	12.32	**8.47**
+ CalibSFT	**58.33**	**93.89**	**11.02**	11.85
C3RL	59.69	80.93	25.25	30.19
+ CalibSFT	**64.27**	**85.24**	**14.04**	**12.99**

(g) AIME2025

Method	P@1	AUROC	Brier	ECE
Base	35.00	82.75	48.25	52.81
RLVR	**43.44**	84.95	45.87	49.00
CalibSFT	35.83	**86.49**	**9.61**	**6.59**
RLCR	43.33	**95.53**	**13.60**	23.06
+ CalibSFT	**48.02**	92.40	13.87	**14.28**
ReDoubt	44.90	94.22	15.19	26.51
+ CalibSFT	**51.77**	**96.32**	**11.28**	**16.79**
CoCA	40.31	**95.39**	**17.04**	29.31
+ CalibSFT	**45.00**	94.21	19.99	**26.43**
DCPO	41.15	91.19	**13.33**	**14.30**
+ CalibSFT	**41.25**	**95.07**	13.81	17.73
C3RL	**43.96**	79.01	36.25	42.69
+ CalibSFT	43.12	**86.74**	**21.86**	**25.49**

(h) AIME2026

Method	P@1	AUROC	Brier	ECE
Base	42.50	76.08	45.13	48.07
RLVR	**48.23**	80.70	43.82	45.92
CalibSFT	39.79	**87.44**	**11.26**	**8.52**
RLCR	47.81	**94.10**	14.08	23.19
+ CalibSFT	**52.60**	92.48	**13.57**	**12.04**
ReDoubt	52.40	90.93	16.78	26.20
+ CalibSFT	**55.52**	**93.82**	**11.66**	**17.69**
CoCA	48.54	**96.74**	**14.84**	27.72
+ CalibSFT	**48.65**	93.96	19.52	**25.63**
DCPO	49.58	91.78	**11.04**	**10.41**
+ CalibSFT	**50.21**	**93.07**	12.87	14.52
C3RL	52.81	84.29	27.21	33.72
+ CalibSFT	**53.65**	**91.31**	**15.69**	**18.30**

(i) Overall (ID)

Method	P@1	AUROC	Brier	ECE
Base	59.32	71.90	33.20	34.36
RLVR	**64.16**	73.16	31.10	31.52
CalibSFT	58.51	**79.93**	**17.30**	**14.23**
RLCR	65.33	82.36	14.43	14.46
+ CalibSFT	**67.20**	**86.32**	**12.77**	**8.59**
ReDoubt	65.59	81.05	15.48	16.94
+ CalibSFT	**67.59**	**86.99**	**12.90**	**14.28**
CoCA	63.62	80.66	16.31	19.95
+ CalibSFT	**65.07**	**85.63**	**16.06**	**18.51**
DCPO	64.68	78.21	14.69	12.16
+ CalibSFT	**64.84**	**85.08**	**12.83**	**10.39**
C3RL	65.62	68.36	25.44	28.06
+ CalibSFT	**66.17**	**79.70**	**15.91**	**14.35**

CalibSFT improves calibration on difficult benchmarks. Across individual tasks, CalibSFT initialization lowers ECE in 61 of 80 method–dataset pairs, with the most pronounced improvements where baseline RL models suffer from severe overconfidence. Specifically, on the six benchmarks where baseline RL methods exhibit an ECE exceeding 25% (MinervaMath, MuSiQue, LiveBenchReasoning, NQOpen, PopQA, and WebQuestions), initializing with CalibSFT consistently reduces both Brier score and ECE across all five RL algorithms. On high-accuracy tasks such as DROP, baseline RL models achieve deceptively low ECE (e.g., 2.30% for DCPO) because their confidence clusters around the task accuracy, yet their discrimination remains weak (AUROC $\approx 59\%$). CalibSFT breaks this uninformative concentration and substantially improves discrimination (e.g., DCPO's AUROC

Table 7: Per-dataset results on eight out-of-domain general reasoning benchmarks spanning a range of difficulty. Each entry reports Pass@1, AUROC, Brier score, and ECE. CalibSFT initialization improves calibration and discrimination on most datasets, with the largest gains on benchmarks where baseline RL is heavily overconfident.

(a) HotpotQA

Method	P@1	AUROC	Brier	ECE
Base	64.68	**57.68**	32.82	32.16
RLVR	64.58	54.16	33.10	32.17
CalibSFT	**66.10**	55.08	**30.84**	**22.43**
RLCR	65.80	59.10	25.97	19.77
+ CalibSFT	**66.27**	**59.23**	**24.25**	**10.85**
ReDoubt	**64.83**	60.64	23.52	12.11
+ CalibSFT	64.68	**63.08**	**21.70**	**1.37**
CoCA	64.90	56.43	**22.77**	**6.02**
+ CalibSFT	**66.10**	**61.57**	22.92	11.39
DCPO	**66.15**	57.88	25.33	18.77
+ CalibSFT	65.97	**60.70**	**24.43**	**14.92**
C3RL	65.38	52.47	33.39	33.12
+ CalibSFT	**66.50**	**58.94**	**28.45**	**19.78**

(b) TriviaQA

Method	P@1	AUROC	Brier	ECE
Base	55.24	75.80	35.68	36.93
RLVR	54.97	**76.01**	40.59	41.40
CalibSFT	**55.53**	64.10	**26.74**	**15.17**
RLCR	55.24	**81.66**	23.84	23.30
+ CalibSFT	**55.25**	81.53	**18.89**	**10.71**
ReDoubt	55.69	82.96	**18.67**	**13.58**
+ CalibSFT	**56.14**	**83.92**	19.27	15.58
CoCA	54.77	77.62	21.63	**12.97**
+ CalibSFT	**54.89**	**81.85**	**20.98**	13.16
DCPO	54.55	79.86	25.93	23.91
+ CalibSFT	**55.12**	**82.56**	**17.59**	**8.25**
C3RL	53.76	**73.01**	35.61	38.53
+ CalibSFT	**54.11**	71.36	**22.75**	**12.50**

(c) DROP

Method	P@1	AUROC	Brier	ECE
Base	80.36	59.13	18.22	17.03
RLVR	**83.53**	**59.32**	**15.34**	**13.31**
CalibSFT	79.76	58.30	33.30	32.64
RLCR	81.44	63.34	**14.72**	**5.43**
+ CalibSFT	**82.32**	**74.44**	14.99	10.75
ReDoubt	**83.63**	63.27	**14.76**	**10.80**
+ CalibSFT	82.53	**75.90**	15.60	17.36
CoCA	82.47	61.58	**15.03**	**11.56**
+ CalibSFT	**82.50**	**73.91**	17.95	21.99
DCPO	**83.45**	59.34	**13.25**	**2.30**
+ CalibSFT	80.42	**75.92**	16.54	13.64
C3RL	**82.65**	54.56	**16.37**	**16.07**
+ CalibSFT	77.44	**64.78**	23.65	22.03

(d) MuSiQue

Method	P@1	AUROC	Brier	ECE
Base	45.84	**62.90**	41.91	42.33
RLVR	47.18	58.12	46.19	46.75
CalibSFT	**48.09**	58.78	**28.43**	**16.30**
RLCR	47.77	66.21	25.68	15.35
+ CalibSFT	**49.97**	**67.87**	**23.22**	**6.13**
ReDoubt	46.90	66.37	24.07	11.21
+ CalibSFT	**48.52**	**70.04**	**22.42**	**5.52**
CoCA	46.69	**67.03**	23.69	10.18
+ CalibSFT	**48.76**	66.88	**23.56**	**7.72**
DCPO	47.34	65.48	28.28	20.53
+ CalibSFT	**48.96**	**70.49**	**22.75**	**8.55**
C3RL	45.55	63.39	39.88	40.04
+ CalibSFT	**47.47**	**66.39**	**24.46**	**11.87**

(e) LiveBenchReasoning

Method	P@1	AUROC	Brier	ECE
Base	52.88	61.81	40.53	40.59
RLVR	29.25	60.32	61.75	64.59
CalibSFT	**55.75**	**69.92**	**34.64**	**35.20**
RLCR	31.75	68.00	28.49	29.09
+ CalibSFT	**42.25**	**68.41**	**21.94**	**7.75**
ReDoubt	25.75	68.44	23.51	24.86
+ CalibSFT	**42.25**	**80.49**	**19.95**	**12.38**
CoCA	32.12	66.75	25.79	23.64
+ CalibSFT	**41.50**	**68.03**	**23.24**	**7.96**
DCPO	31.13	67.14	32.10	33.12
+ CalibSFT	**37.75**	**71.77**	**21.12**	**8.83**
C3RL	33.00	**57.18**	53.55	55.46
+ CalibSFT	**44.62**	53.33	**28.11**	**22.78**

(f) NQOpen

Method	P@1	AUROC	Brier	ECE
Base	21.12	66.31	66.04	70.59
RLVR	**24.13**	**67.07**	68.92	71.55
CalibSFT	21.57	59.58	**25.69**	**26.36**
RLCR	23.32	70.48	43.84	52.43
+ CalibSFT	**23.80**	**72.24**	**26.44**	**30.89**
ReDoubt	**24.67**	70.28	32.15	39.19
+ CalibSFT	24.10	**70.60**	**26.80**	**31.99**
CoCA	22.76	66.59	35.28	43.26
+ CalibSFT	**23.72**	**70.77**	**24.86**	**28.69**
DCPO	**23.69**	69.80	46.27	53.69
+ CalibSFT	23.58	**73.13**	**22.65**	**25.97**
C3RL	**22.43**	**64.77**	63.63	68.49
+ CalibSFT	22.08	61.18	**22.25**	**17.54**

(g) PopQA

Method	P@1	AUROC	Brier	ECE
Base	20.85	**74.94**	57.89	64.10
RLVR	**21.37**	72.78	68.14	71.96
CalibSFT	21.06	64.33	**24.14**	**25.70**
RLCR	**21.29**	**76.88**	37.71	47.90
+ CalibSFT	21.16	76.55	**24.76**	**32.22**
ReDoubt	**21.37**	76.83	26.82	36.23
+ CalibSFT	21.25	**77.87**	**24.54**	**32.32**
CoCA	**21.06**	73.92	30.84	39.88
+ CalibSFT	20.88	**76.99**	**24.12**	**31.76**
DCPO	20.99	**77.02**	41.10	49.96
+ CalibSFT	**21.18**	76.52	**21.29**	**28.02**
C3RL	**20.81**	**71.98**	54.60	61.92
+ CalibSFT	20.66	66.98	**16.56**	**12.95**

(h) WebQuestions

Method	P@1	AUROC	Brier	ECE
Base	21.88	**67.10**	68.16	71.80
RLVR	**26.02**	65.76	68.29	70.41
CalibSFT	23.11	59.54	**29.45**	**30.13**
RLCR	**25.07**	69.40	48.86	56.11
+ CalibSFT	24.63	**70.20**	**31.41**	**36.33**
ReDoubt	**26.84**	**70.22**	37.08	43.81
+ CalibSFT	24.91	69.57	**29.11**	**34.59**
CoCA	24.65	64.40	38.35	45.46
+ CalibSFT	**26.06**	**69.71**	**25.93**	**28.47**
DCPO	24.59	68.48	50.52	57.04
+ CalibSFT	**25.61**	**70.87**	**27.45**	**30.32**
C3RL	**23.94**	60.19	67.27	70.35
+ CalibSFT	**23.94**	**61.29**	**24.87**	**19.55**

(i) Overall (OOD)

Method	P@1	AUROC	Brier	ECE
Base	45.36	**65.71**	45.15	46.94
RLVR	43.88	64.19	50.29	51.52
CalibSFT	**46.37**	61.20	**29.15**	**25.49**
RLCR	43.96	69.38	31.14	31.17
+ CalibSFT	**45.71**	**71.31**	**23.24**	**18.20**
ReDoubt	43.71	69.88	25.07	23.98
+ CalibSFT	**45.55**	**73.93**	**22.42**	**18.89**
CoCA	43.68	66.79	26.67	24.12
+ CalibSFT	**45.55**	**71.22**	**22.94**	**18.89**
DCPO	43.99	68.13	32.85	32.41
+ CalibSFT	**44.82**	**72.75**	**21.73**	**17.31**
C3RL	43.44	62.19	45.54	48.00
+ CalibSFT	**44.60**	**63.03**	**23.89**	**17.38**

rises from 59.34% to 75.92%). While RLVR improves Pass@1, it leaves calibration largely unaddressed and even worsens OOD Brier score and ECE relative to the base model. In contrast, CalibSFT alone substantially lowers Brier score and ECE across both splits, effectively shifting the concentrated confidence prior toward a well-calibrated scale. Although CalibSFT alone exhibits limited OOD discrimination, subsequent confidence-aware RL effectively recovers this ranking capability, lifting OOD AUROC from 61.20% to over 71% for four of the five methods. Ultimately, CalibSFT establishes a better-scaled confidence prior, while confidence-aware RL improves discrimination outside the training domain.

Table 8: Performance stability across random seeds on Qwen3-8B. Each entry reports the mean and standard deviation over 3 random seeds (43, 44, and 45), following the evaluation protocol of Table 1. The variation remains small across all metrics, and CalibSFT consistently outperforms RL baselines while maintaining task accuracy.

	In-distribution				Out-of-distribution			
Method	Pass@1 ↑	AUROC ↑	Brier ↓	ECE ↓	Pass@1 ↑	AUROC ↑	Brier ↓	ECE ↓
RLCR	$65.50_{\pm 0.66}$	$82.08_{\pm 0.38}$	$14.64_{\pm 0.29}$	$14.26_{\pm 0.29}$	$43.53_{\pm 0.61}$	$69.05_{\pm 0.47}$	$31.67_{\pm 0.75}$	$31.99_{\pm 1.15}$
+ CalibSFT	$\mathbf{67.15}_{\pm 0.92}$	$\mathbf{86.29}_{\pm 0.19}$	$\mathbf{12.98}_{\pm 0.47}$	$\mathbf{9.36}_{\pm 1.51}$	$\mathbf{45.69}_{\pm 0.12}$	$\mathbf{71.77}_{\pm 0.65}$	$\mathbf{22.70}_{\pm 0.77}$	$\mathbf{17.75}_{\pm 0.64}$
DCPO	$64.69_{\pm 0.21}$	$78.40_{\pm 0.41}$	$14.63_{\pm 0.19}$	$11.78_{\pm 0.33}$	$43.77_{\pm 0.39}$	$68.37_{\pm 0.54}$	$33.22_{\pm 1.15}$	$32.69_{\pm 1.09}$
+ CalibSFT	$\mathbf{64.90}_{\pm 0.19}$	$\mathbf{85.22}_{\pm 0.15}$	$\mathbf{12.45}_{\pm 0.29}$	$\mathbf{10.15}_{\pm 0.60}$	$\mathbf{44.89}_{\pm 0.12}$	$\mathbf{72.91}_{\pm 0.42}$	$\mathbf{21.60}_{\pm 0.58}$	$\mathbf{17.39}_{\pm 0.36}$

D.3 Robustness across Random Seeds

The main results report single-run evaluations per method. To confirm that the improvements from CalibSFT consistently exceed training variance, we repeat RLCR and DCPO on Qwen3-8B across three random seeds (43, 44, and 45) under the protocol of Table 1.

CalibSFT improvements are stable across random seeds. As shown in Table 8, training variance across runs is minimal. Across all evaluated seeds, CalibSFT consistently outperforms RL baselines on all calibration and discrimination metrics across both in-distribution and out-of-distribution benchmarks. Meanwhile, task accuracy (Pass@1) remains comparable to the baseline. These results confirm that the improvements brought by CalibSFT are stable.

D.4 Evaluation on Additional Metrics

While Table 1 evaluates core accuracy, calibration, and discrimination, we further assess the policies along two complementary dimensions: fine-grained calibration and confidence diversity. All metrics are computed on the same results from the main evaluation.

Fine-grained calibration. To thoroughly evaluate confidence reliability, we consider 4 additional metrics that evaluate question-level calibration, failure prediction, and calibration ranking:

- **Positive Calibration Error (PCE)** (Ma et al., 2026) focuses on overconfidence by restricting ECE to bins where average confidence is higher than accuracy.
- **Capability Brier score (Cap. Brier)** (Yang et al., 2026a) measures how well a question's mean confidence matches its empirical rollout success rate, tracking calibration against problem difficulty.
- **Excess Area Under the Risk–Coverage Curve (E-AURC)** (Geifman et al., 2019) evaluates failure prediction by comparing the model's risk–coverage curve to a perfect ranking at the same accuracy.
- **Rank Calibration Error (RCE)** (Huang et al., 2024) measures whether confidence increases monotonically with expected correctness across equal-mass bins, which is independent of model accuracy and score scales.

CalibSFT improves confidence quality beyond standard metrics. As shown in Table 9, CalibSFT initialization greatly outperforms standard RL baselines across these metrics. In particular, the large drop in RCE indicates that higher confidence more reliably reflects higher accuracy. Meanwhile, the sharp decrease in out-of-distribution PCE shows that the method effectively lowers overconfident predictions on unfamiliar problems, without merely shifting all confidence scores down.

Confidence diversity. To examine whether models explore a broader range of confidence values, we evaluate 4 diversity metrics:

- **Support Size** is the number of unique confidence values generated across all responses in a split, directly measuring the size of the confidence support.

Table 9: Calibration performance on additional metrics, on the in-distribution and out-of-distribution splits and under the protocol of Table 1. PCE is the part of ECE contributed by the overconfident bins, Cap. Brier scores each question's mean confidence against its rollout success rate, E-AURC is the excess of the risk–coverage area over a perfect ranking, and RCE measures the deviation from a monotone correspondence between confidence and expected correctness. Across all baseline RL methods, incorporating CalibSFT reduces overconfidence and misranking errors (lower PCE, E-AURC, and RCE) while improving question-level confidence alignment (lower Cap. Brier).

	In-distribution				Out-of-distribution			
Method	PCE ↓	Cap. Brier ↓	E-AURC ↓	RCE ↓	PCE ↓	Cap. Brier ↓	E-AURC ↓	RCE ↓
RLCR	9.71	10.63	7.75	13.50	30.99	27.53	19.74	11.45
+ CalibSFT	**7.30**	**8.66**	**5.61**	**5.30**	**14.77**	**18.72**	**17.14**	**4.93**
ReDoubt	6.71	11.98	8.02	15.23	22.51	21.71	18.34	14.77
+ CalibSFT	**5.54**	**9.33**	**4.77**	**7.69**	**15.58**	**18.80**	**15.85**	**4.70**
CoCA	**10.96**	12.82	8.88	19.49	22.64	22.84	21.52	15.13
+ CalibSFT	11.40	**12.24**	**5.50**	**8.58**	**13.39**	**19.41**	**17.82**	**5.79**
DCPO	10.04	10.95	10.10	18.79	31.97	28.95	19.62	15.40
+ CalibSFT	**8.09**	**9.00**	**5.92**	**9.21**	**12.95**	**17.54**	**16.23**	**6.18**
C3RL	28.01	20.75	16.11	33.15	47.96	41.92	25.22	27.57
+ CalibSFT	**10.72**	**10.82**	**8.76**	**17.24**	**8.80**	**18.46**	**21.27**	**19.08**

Table 10: Confidence diversity on the in-distribution and out-of-distribution splits, under the protocol of Table 1. Support counts the unique confidence values a model produces, Top-1 and Top-3 are the share of responses taking its most frequent one and three values, and Effective Support (Eff. Supp.) is the exponential of the entropy of the value distribution. Baseline RL methods collapse into narrow confidence supports with heavy concentration on top values, whereas CalibSFT broadens both support size and effective support for all objectives.

	In-distribution				Out-of-distribution			
Method	Support ↑	Top-1 ↓	Top-3 ↓	Eff. Supp. ↑	Support ↑	Top-1 ↓	Top-3 ↓	Eff. Supp. ↑
RLCR	15.75	33.30	67.74	6.90	18.62	33.18	66.22	7.03
+ CalibSFT	**86.88**	**12.59**	**20.57**	**49.42**	**94.50**	**3.73**	**9.90**	**72.83**
ReDoubt	7.62	43.17	92.47	3.50	10.88	44.84	86.31	4.20
+ CalibSFT	**58.00**	**9.43**	**23.64**	**32.03**	**59.50**	**7.83**	**20.70**	**31.62**
CoCA	12.75	55.50	77.38	4.85	13.62	46.41	67.06	6.06
+ CalibSFT	**47.62**	**13.62**	**30.12**	**26.22**	**52.12**	**10.71**	**26.88**	**24.69**
DCPO	15.75	59.32	80.94	4.40	18.62	43.97	73.64	6.06
+ CalibSFT	**72.75**	**21.54**	**41.93**	**27.14**	**84.25**	**5.78**	**15.07**	**53.93**
C3RL	8.50	83.23	96.46	1.97	9.38	71.04	91.77	2.79
+ CalibSFT	**10.00**	**46.09**	**70.97**	**5.68**	**10.00**	**36.37**	**65.54**	**6.46**

- **Top-1** and **Top-3** are the percentage of responses falling on the single and three most frequent confidence values, tracking how heavily the distribution concentrates on a few points.
- **Effective Support** is the exponential of the confidence entropy ($\exp(H)$), which measures the effective number of values used by the model. Unlike raw support size, this metric is not inflated by rare values that appear only once or twice.

CalibSFT broadens the confidence distribution. As shown in Table 10, CalibSFT universally expands confidence support across all five RL frameworks on both evaluation splits. In the baseline RL models, predictions collapse onto a tiny set of values. For instance, the top three values often account for 70% to 90% of all responses, yielding an effective support of fewer than 7 values. CalibSFT breaks this concentration: Top-1 and Top-3 shares drop sharply, while both Support Size and Effective Support increase by several times (for example, Effective Support rises from 7.03 to 72.83 for RLCR out-of-distribution). Because both Support Size and Effective Support increase together, the model uses many different confidence values regularly, rather than generating a few rare numbers by chance. Finally, this broader support is consistent with the improved calibration observed in Tables 9 and 2.

Table 11: Distribution (%) of training targets across ten confidence intervals. Target scores derived from prior methods (UaIT, SaySelf, LoVeC) are concentrated or skewed, whereas CalibSFT provides a balanced, near-uniform distribution across the entire range.

Target	$[0.0, 0.1]$	$(0.1, 0.2]$	$(0.2, 0.3]$	$(0.3, 0.4]$	$(0.4, 0.5]$	$(0.5, 0.6]$	$(0.6, 0.7]$	$(0.7, 0.8]$	$(0.8, 0.9]$	$(0.9, 1.0]$
CSFT	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	11.11	0.00
UaIT	0.00	0.00	0.00	0.00	0.00	0.00	6.48	56.06	35.15	2.31
SaySelf	22.07	14.41	11.93	10.22	9.04	8.26	8.89	6.63	5.56	3.00
LoVeC	27.33	3.78	8.17	0.20	1.37	0.63	0.22	1.54	15.80	40.96
CalibSFT	11.11	11.11	11.11	11.11	5.56	11.11	11.11	11.11	11.11	5.56

D.5 Ablation Studies

D.5.1 Confidence Target Construction

The construction of confidence targets in CalibSFT is inspired by proper scoring rules, which serve as a foundation for calibration. To verify the effectiveness of this target design, we compare CalibSFT with targets based on group-level correctness (CSFT) (Jang et al., 2025), token probabilities (UaIT) (Liu et al., 2024a), semantic consistency (SaySelf) (Xu et al., 2024), and rubric-based LLM judgments (LoVeC) (Zhang et al., 2026a), using a shared SFT training set and the same optimization objective.

For a fair comparison, we construct the training data from a shared response pool sampled uniformly across empirical success rates so that the comparison covers both easier and harder questions. For each question in DeepScaleR-Train, we use 10 responses generated by Qwen3-8B, with $z_i \in \{0, 1\}$ denoting correctness and $q_x = \frac{1}{10}\sum_{i=1}^{10} z_i$ the empirical success rate. We select 300 questions at each $q_x \in \{0.1, 0.2, \ldots, 0.9\}$ and retain one correct and one incorrect response per question, yielding 5,400 responses from 2,700 questions with balanced correctness at each success rate. SaySelf uses only its correct responses following its SFT setup. On this shared subset, we construct the five confidence targets as follows:

- **CSFT.** Both responses receive the group success rate, $c_i^* = q_x$.
- **UaIT.** $c_i^* = \exp(-\mathrm{NLL}_i)$, where NLL_i is the mean negative log-likelihood of the sampled response tokens under the generating model.
- **SaySelf.** We embed the ten responses to each question with Qwen3-Embedding-8B and cluster them at a cosine similarity threshold of 0.98. Each retained correct response is assigned the frequency of its cluster, $c_i^* = |\mathcal{M}_i|/10$, where $\mathcal{M}_i$ is the cluster containing response i.
- **LoVeC.** An LLM judge scores each response on a 0–10 rubric (10: correct with sound reasoning; 8–9: correct with minor gaps; 5–7: partially correct or ambiguous; 2–4: meaningful but incorrect; 0–1: invalid or clearly wrong), normalized as $c_i^* = s_i/10$.
- **CalibSFT.** Following our main design, the target combines group success rate with response correctness, $c_i^* = 0.5q_x + 0.5z_i$.

All variants are initialized from Qwen3-8B and use the same supervision objective: reasoning and answer tokens are supervised on correct responses, while confidence tokens are supervised on all responses. We train for 80 steps (40 for SaySelf) with a batch size of 256 and a learning rate of 2×10^{-6}. Table 11 reports the proportion of training targets in each of ten equal-width confidence intervals across the five methods.

Proper-scoring targets preserve both calibration and discrimination. Table 12 reports the AUROC and Brier score of the constructed targets before SFT (Pre-SFT targets), alongside the resulting models' evaluation performance on in-distribution mathematical benchmarks after SFT (Post-SFT predictions). In the pre-SFT phase, the metrics evaluate the theoretical properties and supervision quality of the training targets themselves. Prior methods struggle to balance calibration and discrimination during target construction. CSFT estimates question-level accuracy q_x from verification signals but assigns the same target to all responses of a question; since each question contributes one correct and one incorrect response, its targets yield an AUROC of 50.00 and a high Brier score of 31.67. Heuristic

Table 12: AUROC and Brier score of five target construction methods on ID mathematical benchmarks. Pre-SFT metrics characterize the constructed targets, whereas Post-SFT metrics evaluate the fine-tuned models.

	Pre-SFT targets		Post-SFT predictions	
Target	AUROC ↑	Brier ↓	AUROC ↑	Brier ↓
CSFT	50.00	31.67	73.46	19.15
UaIT	52.05	33.17	74.72	24.24
SaySelf	–	41.33	57.69	39.38
LoVeC	75.40	24.71	71.11	32.32
CalibSFT	**100.00**	**7.92**	**79.68**	**17.74**

Table 13: AUROC and Brier score on eight ID and eight OOD benchmarks across mixing weights. $\lambda = 0.5$ achieves the highest AUROC on both splits and the lowest Brier score on ID benchmarks.

	ID		OOD	
λ	AUROC ↑	Brier ↓	AUROC ↑	Brier ↓
0	68.10	32.49	59.20	38.92
0.25	78.02	18.83	66.88	29.94
0.5	**79.68**	**17.74**	**67.81**	30.82
0.75	75.93	20.00	65.82	**28.37**
1	73.29	19.90	57.22	35.72

targets derived from token probabilities (UaIT), semantic consistency (SaySelf), or LLM judgments (LoVeC) lack grounding in strictly proper scoring rules, which leads to miscalibrated targets and high Brier scores. In contrast, by combining group success rates with individual response correctness z_i, CalibSFT separates correct from incorrect responses by construction at $\lambda = 0.5$ (an AUROC of 100.00), and its Brier score equals λ^2 times that of question-level targets (7.92 vs. 31.67).

Target-level advantages translate to superior post-SFT performance. These advantages at target construction transfer to the post-SFT models. Averaged over eight mathematical benchmarks, the policy fine-tuned with CalibSFT achieves the strongest performance on both metrics. Specifically, CalibSFT obtains the highest AUROC of 79.68 and the lowest Brier score of 17.74, outperforming models trained on alternative targets. These results confirm that grounding target construction in proper scoring rules yields supervision that improves both calibration and discrimination of the trained model.

D.5.2 Target-Balanced Sampling

Our CalibSFT pipeline balances training responses across confidence targets. This step prevents frequent targets (especially 0 and 1) from dominating the loss and pushing predicted confidences toward the endpoints. To verify this effect, we compare our balanced model against a baseline trained on the raw, unbalanced responses from the same $n = 50$ candidate pool, evaluating both on DeepScaleR-Eval.

Target balancing mitigates concentration at extremes. As shown in Figure 8 and Table 14, balancing has little impact on standard calibration error, but substantially improves confidence diversity. Specifically, it drops the mass at the extremes from 75.56% to 14.49% and expands the effective support from 3.33 to 29.86. This broader distribution is critical for downstream RL: instead of collapsing into near-binary actions (0 or 1), the policy gains access to a continuous, fine-grained action space for optimization, consistent with our diversity analysis in Table 10 and format comparisons in Table 3.

D.5.3 Target Mixing Weight

We construct confidence targets as $c_i^* = \lambda q_x + (1 - \lambda) z_i$, where q_x is the group success rate and z_i is the correctness of an individual response. To evaluate the effect of the mixing weight, we compare $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$ using the same subset as above and keeping the SFT settings fixed. At $\lambda = 0$, targets are restricted to 0 and 1; this setting yields the highest Brier score in Table 13. At $\lambda = 1$, correct and incorrect responses to the same question receive identical targets, providing no within-question discrimination signal; post-SFT AUROC is also lower than at $\lambda = 0.5$. For $\lambda > 0.5$, a correct response to a hard question can receive a lower target than an incorrect response to an easy question, since $\lambda q_h + (1 - \lambda) < \lambda q_e$ whenever $q_e - q_h > (1 - \lambda)/\lambda$; post-SFT AUROC accordingly drops from 79.68 at $\lambda = 0.5$ to 75.93 at $\lambda = 0.75$. In contrast, any $\lambda \leq 0.5$ keeps every correct target above every incorrect one, because $\lambda(q_e - q_h) < \lambda \leq 1 - \lambda$. Equal weighting is thus the largest weight on the success rate that preserves this ordering, and it achieves the highest AUROC and lowest Brier score on ID benchmarks among the tested settings, so we use $\lambda = 0.5$ as the default.

Table 14: Ablation of target-balanced sampling on DeepScaleR-Eval. Balancing substantially broadens confidence support and prevents concentration at extremes, while maintaining comparable task performance (Pass@1) and calibration.

	Accuracy	**Calibration Metrics**			**Confidence Diversity**			
Sampling	Pass@1 ↑	ECE ↓	Brier ↓	AUROC ↑	Support ↑	Eff. Supp. ↑	Top-1 ↓	Top-3 ↓
Balanced	**58.50**	**14.23**	17.30	79.92	**85.62**	**29.86**	**28.89**	**47.05**
Unbalanced	58.03	16.41	**16.64**	**81.74**	21.75	3.33	61.57	92.25

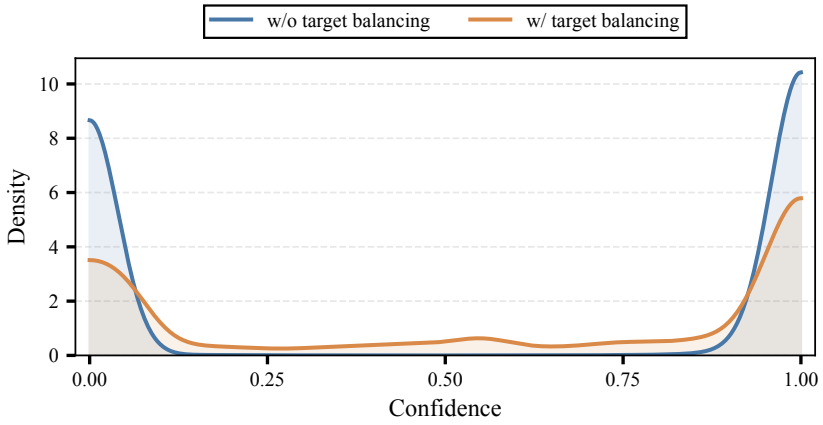

Figure 8: Confidence distributions on DeepScaleR-Eval after CalibSFT. Target-balanced sampling prevents probability mass from collapsing to the extremes (0 and 1), fostering a broader distribution.

Table 15: Ablation of the CalibSFT supervision design on eight ID benchmarks. The variants differ in the supervised responses for each segment and in loss balancing. CalibSFT achieves the best calibration while keeping Pass@1 close to the best.

	Reasoning/answer		Confidence		Segment	ID performance			
Variant	Correct	Incorrect	Correct	Incorrect	balanced	Pass@1 ↑	AUROC ↑	Brier ↓	ECE ↓
Confidence-only	–	–	✓	✓	–	43.56	74.62	20.97	17.43
Full-CE	✓	✓	✓	✓	–	58.49	61.66	29.29	27.65
Full-Balanced	✓	✓	✓	✓	✓	58.46	76.67	19.04	16.87
Correct-only	✓	–	✓	–	✓	**59.89**	77.58	18.95	16.98
CalibSFT	✓	–	✓	✓	✓	58.51	**79.93**	**17.30**	**14.23**

D.5.4 Correctness-Conditional Supervision

To verify the effectiveness of the CalibSFT design, we ablate three choices in the SFT loss: restricting reasoning and answer supervision to correct responses, supervising confidence on all responses, and balancing the segment losses. Table 15 reports the results.

Correctness-conditional supervision improves calibration and discrimination. Confidence-only supervision leaves Pass@1 far behind the other variants, showing that reasoning and answer supervision is necessary for accuracy. Without segment balancing, calibration is markedly worse: Full-CE trails Full-Balanced on both AUROC and ECE. Restricting reasoning and answer supervision to correct responses slightly raises accuracy, consistent with incorrect reasoning pulling the model toward incorrect solutions, but restricting confidence supervision to correct responses as well narrows its coverage and weakens calibration. Supervising confidence on all responses while keeping reasoning and answer supervision correct-only recovers this coverage: CalibSFT achieves the best AUROC, Brier score, and ECE among all variants, with Pass@1 close to the best. This combination best balances accuracy, calibration, and confidence diversity, giving a well-calibrated starting point for downstream confidence-aware RL.

Table 16: Confidence estimators for RLCR with and without CalibSFT initialization, averaged over the 8 in-distribution and 8 out-of-distribution benchmarks. Self-consistency aggregates the sampled responses by majority vote, so its Pass@1 column reports majority-vote accuracy. CalibSFT initialization improves calibration and discrimination under all three estimators.

		In-distribution				Out-of-distribution			
Estimator	Method	Pass@1 ↑	AUROC ↑	Brier ↓	ECE ↓	Pass@1 ↑	AUROC ↑	Brier ↓	ECE ↓
Verbalized	RLCR	65.33	82.36	14.43	14.46	43.96	69.38	31.14	31.17
	+ CalibSFT	**67.20**	**86.32**	**12.77**	**8.59**	**45.71**	**71.31**	**23.24**	**18.20**
Token probability	RLCR	65.33	83.60	14.35	15.03	43.96	69.70	31.22	30.54
	+ CalibSFT	**67.20**	**87.66**	**12.75**	**10.13**	**45.71**	**71.51**	**22.80**	**19.29**
Self-consistency	RLCR	70.86	85.36	13.09	17.62	44.83	70.83	30.61	30.53
	+ CalibSFT	**72.31**	**87.64**	**11.35**	**9.67**	**47.24**	**72.51**	**22.77**	**18.19**

D.6 Integration with Other Confidence Estimators

While this work primarily studies verbalized confidence, confidence can also be derived from internal token probabilities or sample consistency. To evaluate whether the benefits of CalibSFT generalize across different confidence estimation paradigms, we examine two widely adopted alternatives using the exact response samples and evaluation protocol from the main experiment (Table 1):

- **Token probability.** We extract model confidence directly from the output logits over confidence tokens. Specifically, following the prefix `<confidence>0.`, we apply a restricted softmax over the ten digit tokens $\{0, \ldots, 9\}$ to obtain normalized probabilities $P(d)$, computing the expected confidence as:

$$c = \sum_{d=0}^{9} P(d) \left(\frac{d}{10} + 0.05 \right),$$

  where each bin is represented by its midpoint.
- **Self-consistency.** Following majority voting (Wang et al., 2023), we aggregate multiple rollouts per question (4 by default; 32 on AIME) by answer equivalence. The dominant consensus answer determines final correctness, and its corresponding confidence is defined as the mean verbalized score across all rollouts in this consensus group.

CalibSFT transfers across confidence estimators. Table 16 compares these estimators alongside verbalized confidence. Crucially, CalibSFT initialization consistently improves all metrics across every estimator on both in-distribution and out-of-distribution benchmarks. For instance, under token probability, CalibSFT boosts in-distribution AUROC from 83.60 to 87.66 while cutting out-of-distribution Brier score from 31.22 to 22.80. Under self-consistency, it reduces ECE substantially on both splits (from 17.62 to 9.67 ID and from 30.53 to 18.19 OOD).

D.7 Confidence Visualization

To intuitively illustrate how CalibSFT alleviates calibration errors across tasks of varying difficulty, we examine the reliability diagrams of RLCR with and without CalibSFT initialization. Specifically, we evaluate on three in-distribution mathematical benchmarks of distinct difficulty (MATH-500, AIME 2026, and MinervaMath; Figure 9) and three out-of-distribution reasoning benchmarks (LiveBenchReasoning, TriviaQA, and NQOpen; Figure 10) across ten equal-width confidence intervals.

CalibSFT improves calibration across task difficulties. As shown in Figures 9 and 10, CalibSFT effectively mitigates both overconfidence and underconfidence across tasks with different difficulty levels. On harder benchmarks where standard RLCR is heavily overconfident, its average confidence exceeds empirical accuracy by 23.0 points on MinervaMath, 52.4 points on NQOpen, and 29.1 points on LiveBenchReasoning. Initializing with CalibSFT reduces the gap between confidence and empirical accuracy, lowering ECE from 23.05% to 12.00% on MinervaMath, from 52.43% to 30.89%

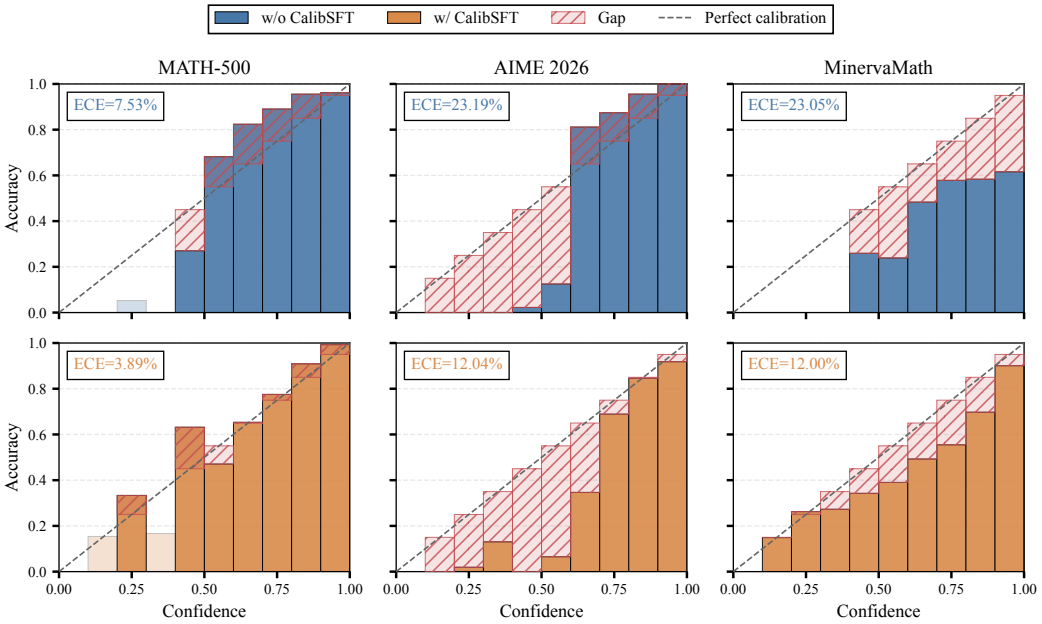

Figure 9: Reliability diagrams of RLCR with and without CalibSFT across three in-distribution mathematical benchmarks. CalibSFT effectively reduces both underconfidence on easier tasks (MATH-500) and overconfidence on harder tasks (MinervaMath and AIME 2026), aligning confidence closer to actual accuracy.

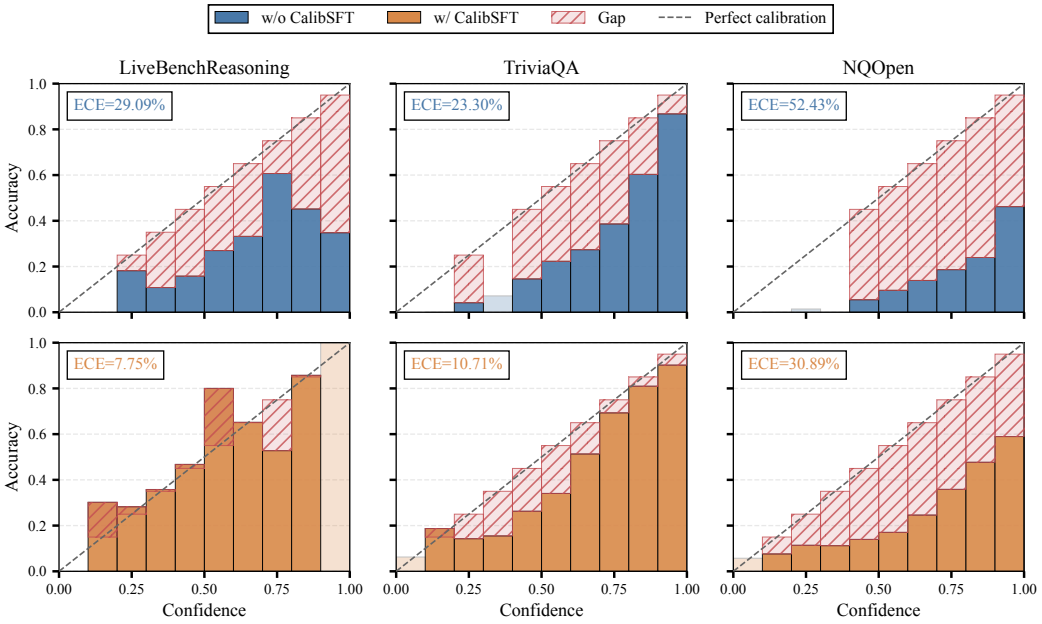

Figure 10: Reliability diagrams of RLCR with and without CalibSFT on out-of-distribution reasoning benchmarks. CalibSFT substantially alleviates severe overconfidence across all three tasks and prevents confidence from collapsing onto a few extreme values.

on NQOpen, and from 29.09% to 7.75% on LiveBenchReasoning. Meanwhile, on easier datasets such as MATH-500 where RLCR exhibits underconfidence, CalibSFT properly raises the confidence scores and reduces ECE from 7.53% to 3.89%.

Table 17: Training cost comparison between CalibSFT and confidence-aware RL under a 200-step training budget on eight NVIDIA B20Z GPUs. CalibSFT takes substantially less training time and consumes only 6.9% of the GPU-hours of RLCR.

Stage	Minutes per step	Wall-clock (h)	GPU-hours
CalibSFT	0.33	1.1	8.8
RLCR	4.77	15.9	127.3
RLCR + CalibSFT	4.58	15.3	122.2

D.8 Running Time Analysis

To evaluate the computational efficiency of CalibSFT, we compare its training overhead against standard confidence-aware RL. CalibSFT requires no manual annotation, as its targets are derived purely from self-sampled rollouts verified by the same automated verifier used in downstream RL. Crucially, these rollout responses are collected only once before fine-tuning, whereas RL needs to sample new responses at every optimization step.

CalibSFT relies on a one-time offline rollout stage prior to policy fine-tuning. For $n = 50$, we generate 50 responses per training question, requiring approximately 147 GPU-hours in total (roughly 16.4 wall-clock hours on an 8$\times$ NVIDIA B20Z node). For a given base model and prompt format, this curated dataset can be reused across all downstream confidence-aware RL methods, avoiding repeated rollout collection. Reducing the rollout budget to $n = 10$ lowers the collection cost to approximately 32 GPU-hours (around 4 wall-clock hours), providing a compute-efficient alternative. This setting yields probability targets on the grid $\{0.00, 0.05, \ldots, 1.00\}$ and remains effective: the corresponding CalibSFT model achieves an AUROC of 79.68 and a Brier score of 17.74 in the target-construction experiment (Table 12).

CalibSFT requires little fine-tuning time. As shown in Table 17, CalibSFT introduces minimal overhead compared to downstream RL. On a single node with 8 NVIDIA B20Z GPUs, each CalibSFT step takes only 0.33 minutes, compared to 4.77 minutes per step for RLCR. Consequently, the CalibSFT fine-tuning stage accounts for merely 6.9% of the GPU-hours required by RLCR. The dominant cost is the one-time rollout collection, which is shared across all downstream confidence-aware RL methods.

E Downstream Applications

Beyond standard calibration and discrimination metrics, we evaluate whether CalibSFT's gains transfer into practical benefits for downstream decision-making under finite-sample statistical guarantees. We investigate two deployment scenarios: (1) **selective risk control** (§E.1), which maximizes response coverage while strictly controlling the selective risk (error rate) among accepted answers, and (2) **confidence-guided model routing** (§E.2), which minimizes calls to a stronger and more expensive model while provably bounding accuracy loss relative to a full-deferral baseline.

E.1 Selective Risk Control

Selective prediction allows a reasoning model to answer a question only when its confidence is high enough, and abstain otherwise to avoid incorrect answers. Specifically, given a threshold τ, the model outputs an answer only if its confidence satisfies $c \geq \tau$. Under this setting, *coverage* is the fraction of answered questions, while *selective risk* is the error rate among them. The goal is to maximize coverage while keeping selective risk below a given budget $r_{\max}$ (Huang et al., 2025). We test whether CalibSFT yields higher certified coverage under the same risk budget.

To select a valid threshold with finite-sample guarantees, we adopt the Learn-then-Test framework (Angelopoulos et al., 2025) following Shen (2026). Specifically, over a grid of 101 candidate thresholds in $\{0, 0.01, \ldots, 1\}$, we compute a one-sided Clopper–Pearson upper bound on selective risk (Clopper & Pearson, 1934) at confidence level $1 - \delta/101$, with $\delta = 0.05$. By a union bound, these risk bounds hold simultaneously across all candidates with probability at least $1 - \delta$. We

Table 18: Certified coverage (%) at fixed risk budgets for confidence-aware RL with and without CalibSFT initialization. Values report the mean and standard deviation on held-out questions over ten random splits ($\delta = 0.05$). Perfect ranking serves as an oracle reference that ranks all correct responses above incorrect ones. CalibSFT initialization consistently increases certified coverage across all methods and risk budgets.

	In-distribution		Out-of-distribution
Method	$r_{\max} = 10\%$	$r_{\max} = 20\%$	$r_{\max} = 20\%$
RLCR	$31.4_{\pm 0.6}$	$86.4_{\pm 1.1}$	$0.0_{\pm 0.0}$
+ CalibSFT	$\mathbf{62.6}_{\pm 2.5}$	$\mathbf{88.6}_{\pm 1.3}$	$\mathbf{20.3}_{\pm 0.8}$
ReDoubt	$28.8_{\pm 0.5}$	$70.2_{\pm 12.3}$	$0.0_{\pm 0.0}$
+ CalibSFT	$\mathbf{66.1}_{\pm 0.9}$	$\mathbf{88.6}_{\pm 1.2}$	$\mathbf{18.0}_{\pm 2.5}$
CoCA	$0.0_{\pm 0.0}$	$82.3_{\pm 1.4}$	$0.0_{\pm 0.0}$
+ CalibSFT	$\mathbf{52.3}_{\pm 10.5}$	$\mathbf{87.2}_{\pm 2.5}$	$\mathbf{20.5}_{\pm 0.2}$
DCPO	$1.6_{\pm 4.7}$	$85.5_{\pm 2.2}$	$0.0_{\pm 0.0}$
+ CalibSFT	$\mathbf{33.3}_{\pm 21.4}$	$\mathbf{87.3}_{\pm 1.4}$	$\mathbf{22.8}_{\pm 1.5}$
C3RL	$0.0_{\pm 0.0}$	$17.7_{\pm 35.3}$	$0.0_{\pm 0.0}$
+ CalibSFT	$\mathbf{49.2}_{\pm 16.4}$	$\mathbf{84.8}_{\pm 0.5}$	$\mathbf{16.8}_{\pm 0.2}$
Perfect ranking	82.0	92.3	54.5

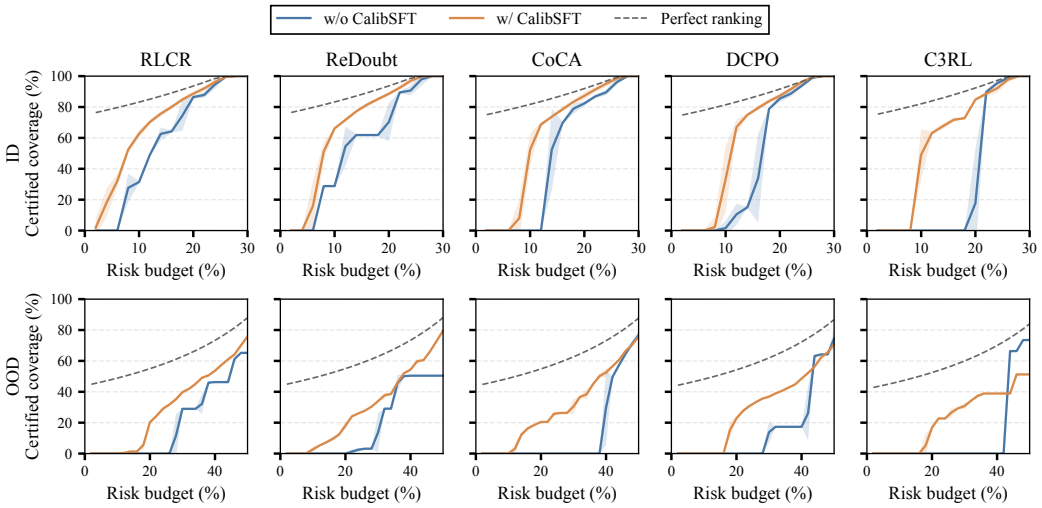

Figure 11: Certified coverage across risk budgets on ID (top) and OOD (bottom) benchmarks. The results are averaged over ten random splits. Initializing with CalibSFT consistently yields higher certified coverage across budgets and tracks the perfect-ranking reference curve more closely, particularly under strict risk tolerances.

then choose the smallest threshold whose upper bound is at most $r_{\max}$, maximizing coverage; if no candidate qualifies, the model abstains on all questions. For both ID and OOD benchmarks, we randomly split the questions into equal calibration and test sets ($50\%/50\%$). We use one response per calibration question to determine the threshold, and evaluate coverage and selective risk on all responses in the test set. We repeat this procedure over ten random splits and report the mean and standard deviation of coverage.

CalibSFT increases coverage under a fixed error-rate guarantee. Table 18 shows that CalibSFT initialization substantially boosts certified coverage across all evaluated RL algorithms under identical risk budgets. On ID benchmarks at a 10% risk budget, CalibSFT raises RLCR's coverage from

Table 19: Confidence-guided model routing to Qwen3.7-Flash across two accuracy-loss tolerances ε, where local policies are trained on Qwen3-8B. Values are averaged over ten random splits. CalibSFT initialization substantially reduces deferral rates and increases token savings while maintaining task accuracy close to the full-deferral baseline (82.58%).

	Tolerance $\varepsilon = 1\%$			Tolerance $\varepsilon = 2\%$		
Method	Deferred (%) ↓	Tokens saved (%) ↑	Accuracy	Deferred (%) ↓	Tokens saved (%) ↑	Accuracy
RLCR	$65.0_{\pm 1.1}$	10.1	82.97	$45.9_{\pm 1.0}$	23.3	83.33
+ CalibSFT	$\mathbf{60.3}_{\pm 3.6}$	**17.9**	82.66	$\mathbf{39.2}_{\pm 1.0}$	**36.1**	82.47
ReDoubt	$67.4_{\pm 0.8}$	10.1	82.99	$43.1_{\pm 1.0}$	27.1	83.24
+ CalibSFT	$\mathbf{55.3}_{\pm 2.7}$	**23.8**	82.81	$\mathbf{36.3}_{\pm 3.7}$	**42.1**	82.60
CoCA	$98.7_{\pm 0.1}$	0.6	82.61	$98.7_{\pm 0.1}$	0.6	82.61
+ CalibSFT	$\mathbf{71.7}_{\pm 4.9}$	**17.3**	82.51	$\mathbf{51.2}_{\pm 5.5}$	**33.1**	82.42
DCPO	$82.9_{\pm 0.7}$	5.6	82.64	$82.9_{\pm 0.7}$	5.6	82.64
+ CalibSFT	$\mathbf{77.4}_{\pm 0.3}$	**12.2**	82.58	$\mathbf{33.3}_{\pm 3.8}$	**43.5**	82.80

31.4% to 62.6%, and enables CoCA and C3RL to reach 52.3% and 49.2% coverage, respectively, compared with zero coverage for both baselines. On OOD benchmarks at a 20% risk budget, all five baselines yield zero certified coverage, whereas CalibSFT achieves coverage ranging from 16.8% to 22.8%. Figure 11 further illustrates that CalibSFT consistently brings certified coverage closer to the perfect-ranking reference across risk budgets, especially under stricter risk constraints. These results indicate that mitigating confidence concentration is important for risk-controlled deployment, allowing the model to certify more responses rather than conservatively abstaining.

E.2 Confidence-Guided Model Routing

Confidence-guided model routing uses a local model's confidence to decide whether to retain its answer or defer the question to a stronger, more expensive model. The local model first generates an answer and a confidence score for every question. Its answer is retained when the confidence is at least a threshold τ; otherwise, the question is sent to Qwen3.7-Flash. Following work on efficient reasoning and risk-aware routing (Yu et al., 2026a; Zeng et al., 2026; Hao et al., 2026), we aim to reduce reliance on the stronger model while keeping the accuracy loss relative to deferring all questions below a tolerance ε. We measure cost using the fraction of questions deferred and the percentage of Qwen3.7-Flash tokens saved relative to this baseline.

To certify the routing threshold τ, we bound the probability that a question is answered locally and incorrectly while the stronger model would answer it correctly, which limits the accuracy loss relative to deferring all questions to the stronger model. Following §E.1, we compute one-sided Clopper–Pearson upper bounds (Clopper & Pearson, 1934) with a union-bound correction across 102 candidate thresholds at significance level $\delta = 0.05$, selecting the smallest threshold whose upper bound does not exceed ε. In our experiments, the local Qwen3-8B is instantiated by each RL policy, while the stronger model is Qwen3.7-Flash. We evaluate on 4,828 questions from eight ID mathematics benchmarks using one rollout per question from each model, randomly splitting the questions into equal calibration and test sets across ten independent trials to report the mean results along with the standard deviation of the deferral rate.

CalibSFT improves routing efficiency under an accuracy-loss guarantee. Table 19 shows that CalibSFT initialization consistently lowers deferral rates across all evaluated RL methods under both accuracy-loss tolerances. Across these baselines, CalibSFT achieves an average deferral reduction of 12.3 and 27.7 percentage points at $\varepsilon = 1\%$ and $\varepsilon = 2\%$, respectively. In particular, at $\varepsilon = 2\%$, CalibSFT cuts DCPO's deferral rate from 82.9% to 33.3%, thereby boosting token savings from 5.6% to 43.5%. Across both tolerances, routing guided by CalibSFT maintains overall accuracies between 82.42% and 82.81%, which closely match the full-deferral baseline (82.58%). By lowering confidence on incorrect responses, CalibSFT enables the local model to reliably retain valid solutions, substantially reducing computational overhead in cascaded reasoning without sacrificing accuracy.