# When Words Fall Short: Iterative Synergy Between Verbalized Reasoning and Hidden Features for LLM Confidence Estimation

**Anonymous ACL submission**


## Abstract

Confidence estimation is crucial for developing trustworthy large language models (LLMs), with most methods following estimator-based or verbalization-based paradigms. While recent research increasingly focuses on improving verbalized confidence, we challenge the prevailing consensus that this approach surpasses independent confidence estimators. A preliminary analysis shows that, with careful training design, a dedicated confidence estimator can extract richer representation-level information than verbalized self-reports. To leverage this insight, we propose **I**terative **Po**licy-**E**stimator **T**raining (**IPoET**), a framework that synergizes the complementary strengths of verbalized reasoning traces and informative representations. IPoET alternates policy optimization with estimator updating, integrating estimator-derived confidence feedback into policy learning and refreshing the estimator on updated policy rollouts. Experiments demonstrate that by iteratively exploiting richer hidden features and adapting to the evolving policy distribution, IPoET consistently surpasses both estimator- and verbalization-based baselines for confidence estimation across diverse datasets using Qwen and Llama backbones. Our code is available at `https://anonymous.4open.science/r/ipoet`.


## 1 Introduction

Large language models (LLMs) have become increasingly capable across a wide range of challenging tasks, such as reasoning, code generation, and planning (Jaech et al., 2024; Guo et al., 2025; Jimenez et al., 2024; Xiao et al., 2024). However, higher accuracy does not necessarily mean that models can correctly assess whether their outputs should be trusted. LLMs frequently generate hallucinated or incorrect answers with unwarranted confidence (Xiong et al., 2024; Anh-Hoang et al., 2025). This lack of reliable confidence estimation severely limits the practical deployment of LLMs in high-stakes domains such as medicine, law, and finance, where decision-making errors carry significant consequences (Li et al., 2024; Siino et al., 2025; Li et al., 2023; Kumaran et al., 2026b).

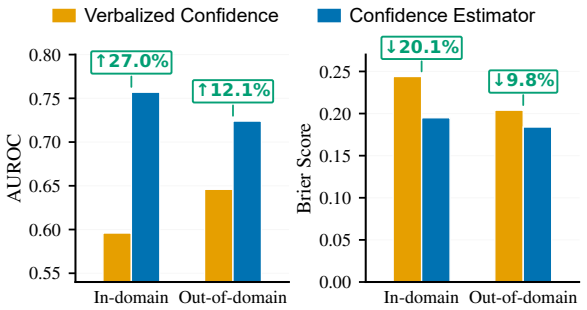


Figure 1: Confidence estimator outperforms verbalized confidence in confidence estimation. The advantage holds across in-domain and OOD evaluations, with higher AUROC and lower Brier score, indicating more reliable confidence estimates from output hidden states.

Existing research on confidence estimation can generally be categorized into two paradigms: (i) *Estimator-based methods*. These methods train independent models or lightweight probing heads to exploit features embedded within the model's hidden states (Beigi et al., 2024; Mahaut et al., 2024). (ii) *Verbalization-based methods*. This paradigm leverages the linguistic reasoning of LLMs and derives verbalized confidence scores through prompting or training (Yang et al., 2024; Li et al., 2026b). Recent work on this task has increasingly focused on RL-based verbalization methods, with several studies reporting that verbalized confidence outperforms estimator-based methods (Damani et al., 2025; Bani-Harouni et al., 2025).

However, revisiting this comparison leads us to a different finding: under carefully controlled training design, an independent confidence estimator can substantially outperform state-of-the-art verbalized confidence in both in-domain and out-of-

domain evaluations, as shown in Figure 1. Prior biased conclusions may stem from vulnerable training settings during estimator development, which easily lead to overfitting in the absence of proper checkpoint selection using held-out validation sets.

Nevertheless, these results indicate that hidden features within model representations are essential for accurate confidence estimation, yet they cannot be easily translated into natural language (Kumaran et al., 2026a). At the same time, decoupling the estimator from the main policy limits its ability to fully utilize the advanced reasoning capabilities of contemporary LLMs, which have shown strong effectiveness in confidence estimation and various other tasks. This raises a central question: *Can we harness the advantages of both paradigms to achieve more reliable confidence estimation?*

To answer this, we propose **I**terative **Po**licy-**E**stimator **T**raining (**IPoET**) for confidence estimation. It alternates between two stages: (i) optimizing the policy with an objective that incorporates estimator feedback while preserving task accuracy, and (ii) updating the estimator to leverage hidden states encoding reasoning traces from the current policy. This process enables policy reasoning to enrich the feature representations of the estimator, while keeping the estimator aligned with the evolving policy distribution. Consequently, this co-adaptation can yield stronger confidence estimation compared to isolated optimization.

We evaluate IPoET across multiple mathematical reasoning and question-answering datasets using different backbone models. Compared to both estimator- and verbalization-based approaches, our method consistently achieves superior or competitive performance on confidence estimation metrics while preserving task performance. Furthermore, these gains generalize effectively to out-of-domain tasks. We also investigate algorithmic variants, such as joint training, to provide deeper insights into the underlying effectiveness of our method.

Our contributions are summarized as follows:

- We revisit estimator- and verbalization-based paradigms, demonstrating that the former yields better confidence estimates and clarifying previous misconceptions.
- We propose IPoET, an iterative training framework that harnesses the advantages of LLM verbalized reasoning and rich internal hidden features for confidence estimation.
- Experiments across diverse datasets validate that IPoET outperforms isolated optimization in both in-domain and out-of-domain settings with distinct backbone models.

## 2 Pilot Study

In this section, we revisit two confidence estimation paradigms: estimator-based and verbalization-based. We begin by highlighting their implementations in Section 2.1, before presenting an empirical comparison of their performance in Section 2.2.

### 2.1 Preliminary

***Task Definition***: Given a prompt $x$ from a dataset $\mathcal{D}$, an LLM $\pi_\theta$ generates a response $y$ consisting of an optional reasoning trace $\tau$ and a final answer $a$. We define the correctness label as $z = \mathbb{I}[a \equiv a^*]$, where $a^*$ is the ground-truth answer. Confidence estimation aims to assign a score $c \in [0, 1]$ representing the likelihood that $a$ is correct.

**Estimator-based approaches.** These approaches utilize an independent confidence estimator $s_\phi$. While prior work typically employs either a lightweight probing head or an entire model as the estimator, we follow Malladi et al. (2023) and adopt the latter for its superior performance. Specifically, $s_\phi$ processes the input $x$ and response $y$ to predict a confidence score $c$ by applying a linear head to the hidden state of the last non-padding token, denoted as $h$: $c = s_\phi(x, y) = \text{clip}(\mathbf{w}^\top h + b, 0, 1)$, where $\mathbf{w}$ and $b$ denote the weights and bias. We train the estimator on collected rollouts $\mathcal{D}_{\text{est}} = \{(x, y, z)\}$ using a mean squared error (MSE) loss:

$$\mathcal{L}_{\text{est}} = \mathbb{E}_{(x,y,z)\sim\mathcal{D}_{\text{est}}} \left[ (s_\phi(x, y) - z)^2 \right]. \quad (1)$$

Previous studies show that critical features for confidence estimation are distributed across intermediate layers (Azaria and Mitchell, 2023; Subramani et al., 2025). Through end-to-end gradient optimization, the final hidden state learns to distill and consolidate these confidence signals scattered throughout the network.

**Verbalization-based approaches.** These approaches leverage the strong capabilities of recent LLMs to generate an answer $a$ alongside verbalized confidence $c$ via a natural language reasoning trace $\tau$. To jointly optimize both variables, this paradigm typically employs RL to search for reasoning traces $\tau$ that provide a more rigorous rationale for inferring a precise confidence $c$. Following

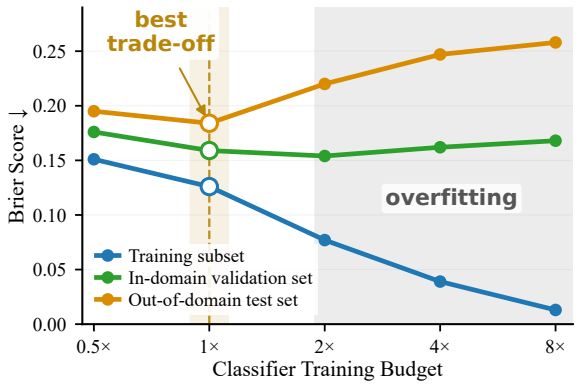


Figure 2: Training sensitivity of hidden-state confidence estimators. Excessive tuning overfits the training distribution and degrades OOD performance; validation-based checkpoint selection mitigates this issue.

RLCR (Damani et al., 2025), we augment the correctness reward in conventional RLVR (Shao et al., 2024) with a Brier-style term over the verbalized confidence:

$$\mathcal{J}_{\text{verb}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,(y,c)\sim\pi_\theta(\cdot|x)}\left[z-(c-z)^2\right]. \tag{2}$$

Damani et al. (2025) show that this approach outperforms the independent estimator trained above.

### 2.2 Estimator-Based vs. Verbalization-Based

We next revisit the comparison between estimator-based and verbalization-based confidence estimation. A fair comparison requires proper estimator training: prior evidence suggests that updating only the prediction head can underuse hidden-state information (Ni et al., 2025), while full-parameter tuning improves estimator capacity but also carries overfitting risk. For this pilot analysis, we use the HotpotQA training set from our main experiments and evaluate out-of-domain performance on the corresponding benchmarks. As shown in Figure 2, increasing the estimator training budget continuously lowers the Brier score on a 5K training subset, but the out-of-domain Brier score increases from 0.184 to 0.258 under excessive tuning. This highlights the importance of a held-out validation set, which provides a practical control signal for checkpoint selection and helps avoid the overfitting regime.

RL-based verbalized confidence methods have the advantage of optimizing confidence expression together with the generated reasoning process (Tao et al., 2024). However, with proper training control, estimator-based confidence estimation compares favorably with this paradigm. As shown in Figure 1, the confidence estimator improves AUROC by 0.161 in-domain and 0.078 out-of-domain, while reducing Brier score by 0.049 and 0.020, respectively. These results challenge the view that verbalized confidence is inherently stronger than hidden-state estimation, highlighting the value of representation-level confidence evidence.

This comparison clarifies the complementarity between estimator-based and verbalization-based confidence estimation. While estimators can extract representation-level confidence evidence, RL-based verbalized methods connect confidence expression with the reasoning process. This observation motivates a training design that moves beyond estimators on fixed outputs by updating generation and confidence estimation in alternating rounds.

## 3 Method

We propose IPoET, an iterative framework that connects reasoning-oriented policy training with hidden-state confidence estimation, as illustrated in Figure 3. To better transfer confidence-estimation signals into policy learning, we adopt reinforcement learning as the optimization mechanism, enabling estimator-derived reward to directly shape the model's generated reasoning. We describe this alternating training procedure in Section 3.1, and then define the estimator-derived reward used for policy optimization in Section 3.2.

### 3.1 Iterative Policy-Estimator Training

Let $\pi_{\theta_0}$ denote the base model. We first apply standard RLVR (Shao et al., 2024) to obtain an initial policy $\pi_{\theta_1}$. This warm-up stage provides a stable output format for answer extraction and correctness verification. Rollouts from $\pi_{\theta_1}$ are then used to train the confidence estimator $s_{\phi_1}$ with the supervised regression objective in Eq. 1.

After initialization, IPoET proceeds in alternating rounds. At iteration $t$, it performs two steps. **Step 1: Freeze estimator $s_{\phi_t}$, update policy $\pi_{\theta_t}$.** The estimator $s_{\phi_t}$ is held fixed while the policy $\pi_{\theta_t}$ is updated with GRPO (Shao et al., 2024) using a reward that combines format, accuracy, and an estimator-derived confidence term, as detailed in Section 3.2. This step incorporates representation-level confidence feedback into policy learning and produces the updated policy $\pi_{\theta_{t+1}}$. **Step 2: Freeze policy $\pi_{\theta_{t+1}}$, update estimator $s_{\phi_t}$.** Since the updated policy may induce a different response distribution, we sample fresh rollouts from $\pi_{\theta_{t+1}}$, assign

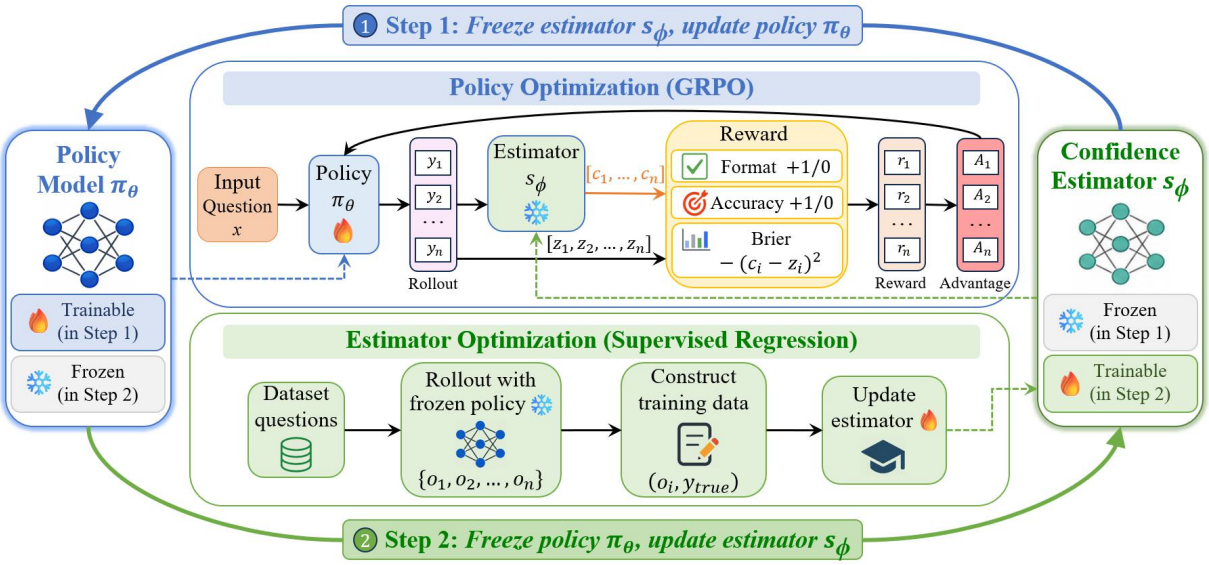


Figure 3: Overview of IPoET. The framework alternates between policy and estimator optimization. In the policy-update stage, the confidence estimator is frozen and provides confidence estimates to form an estimator-derived term, combined with format and accuracy rewards for GRPO. In the estimator-update stage, the policy is frozen to generate correctness-labeled rollouts for updating the estimator via supervised regression. Across iterations, the policy refines reasoning-oriented generation while the estimator adapts to the evolving policy distribution.

correctness labels using task-specific answer verification, and update the estimator to $s_{\phi_{t+1}}$ with the same regression objective in Eq. 1. Across estimator updates, we apply a consistent validation-based checkpoint selection protocol to reduce overfitting and maintain estimator reliability on the current policy distribution.

This training schedule separates policy improvement from estimator refinement while keeping them mutually connected across rounds. Rather than relying on a static estimator trained once, IPoET repeatedly updates the estimator on current-policy rollouts and reuses its confidence estimates in subsequent policy updates, sustaining alignment between generation behavior and hidden-state confidence estimation.

### 3.2 Reward Function

Following the notation in Section 2.1, IPoET optimizes the policy using a reward with three components:

$$R_{\text{IPoET}}(x, y, z; \phi) = R_{\text{fmt}}(y) + R_{\text{acc}}(z) + R_{\text{est}}(x, y, z; \phi). \quad (3)$$

Here, $R_{\text{fmt}}$ maintains the required output structure, and $R_{\text{acc}}(z) = z$ rewards responses whose final answers match the ground-truth answer.

The key component is the estimator-derived reward $R_{\text{est}}$, which introduces hidden-state confidence estimation into policy training. For each rollout, the current estimator produces a confidence score from its internal representation. IPoET converts this representation-level estimate into a policy reward through a negative Brier-style objective:

$$R_{\text{est}}(x, y, z; \phi) = -\left(s_\phi(x, y) - z\right)^2. \quad (4)$$

This term gives higher reward when estimated confidence matches correctness, thereby discouraging both overconfidence and underconfidence.

In this way, IPoET incorporates internal confidence information into policy training without requiring the policy to generate an explicit verbalized confidence score, moving the estimator beyond a separate evaluation-only module.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** Following the experimental setup of RLCR (Damani et al., 2025), we utilize the processed HotpotQA and Big-Math datasets as our training data, derived respectively from the modified HotpotQA distractor data (Yang et al., 2018) and the filtered Big-Math problems (Albalak et al., 2025). We use two separate training settings: one with 20,000 HotpotQA examples, and the other with 14,400 examples randomly sampled from Big-Math according to our training configuration. For evaluation, we consider benchmarks that reflect

different sources of uncertainty in confidence estimation. They fall into three categories: (i) factual question answering, including HotpotQA, TriviaQA (Joshi et al., 2017), and SimpleQA (Wei et al., 2024); (ii) commonsense and expert-level knowledge reasoning, including CommonsenseQA (Talmor et al., 2019) and GPQA (Rein et al., 2023); and (iii) mathematical reasoning, including MATH-500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and Big-Math.

**Evaluation Metrics.** We evaluate each method along two dimensions (details in Appendix C). For task performance, we report Accuracy, computed by comparing the model's final answer with the ground-truth answer. For confidence estimation, we report Area Under the Receiver Operating Characteristic Curve (AUROC) (Bradley, 1997), Brier score (Glenn et al., 1950), and Expected Calibration Error (ECE) (Guo et al., 2017). AUROC evaluates how confidence scores order correct and incorrect predictions, reflecting the ranking quality of confidence estimates. Brier score averages the squared error between the predicted confidence and the binary correctness label, capturing how accurate each confidence estimate is at the instance level. ECE measures the mismatch between predicted confidence and empirical accuracy after grouping predictions into confidence bins.

**Baselines.** We compare against the following representative baselines, including the initial model before applying additional methods. **RLVR** optimizes the policy with a binary answer reward (Shao et al., 2024) and verbalizes confidence only during evaluation. For confidence estimation baselines, **Answer-Prob** uses the generation probability of the final answer tokens as the confidence score (Jiang et al., 2021), while **P(True)** prompts the model to judge whether its own answer is correct and uses the probability of the positive label as confidence (Kadavath et al., 2022). **RLVR + Estimator** trains an additional confidence estimator on RLVR-generated outputs and correctness labels. We further compare with **RLCR**, a recent calibration-oriented RL method proposed by Damani et al. (2025), which trains models to generate both answer and verbalized confidence by optimizing a combined reward consisting of answer correctness and the Brier score of the confidence estimate. **RLCR (Step-aligned)** follows the same RLCR objective but uses the same number of training steps as our method, serving as a control for the effect of additional training. **RLCR + Estimator** applies an additional confidence estimator to RLCR-generated outputs.

**Implementation Details.** Our method is implemented on top of the verl framework (Sheng et al., 2025), with GRPO used as the RL algorithm (Shao et al., 2024). We use Qwen3-8B-Base (Yang et al., 2025) as the main backbone and additionally evaluate Llama-3.1-8B-Instruct (Grattafiori et al., 2024) to assess backbone-level generality. For a fair comparison, we also migrate the RL-based baselines to the same verl-based training pipeline while preserving their original objectives and core design. We use the same training data for all methods within each training setting. Following common practice in RL training for reasoning models (Guo et al., 2025; Hu et al., 2026), we initialize the policy from the backbone model and train it without KL regularization. Training details for our method, including the prompt template, are provided in Appendix B.

### 4.2 Main Results

**IPoET achieves the strongest overall performance among estimator- and verbalization-based baselines.** Table 1 first confirms the observation from our pilot study: estimator-based methods provide more reliable confidence estimates than verbalized confidence. RLCR trains the model to verbalize confidence, but RLVR + Estimator achieves better confidence-estimation results than RLCR in nearly all settings and metrics, indicating that confidence estimators can capture correctness-related information that is not fully reflected in self-reported confidence. The comparison between RLVR + Estimator and RLCR + Estimator further clarifies this point from another angle: although RLCR-generated outputs include additional verbalized confidence analysis, adding an estimator on top of them does not consistently improve confidence estimation and is worse than RLVR + Estimator on most metrics. One possible explanation is that explicit self-assessment may dilute the correctness-related signal in the original reasoning trace and introduce extra noise into the hidden representations. Based on this observation, IPoET keeps the policy focused on task reasoning and relies on the estimator for confidence assessment during training. As a result, IPoET goes beyond both paradigms by feeding the estimator signal back into policy optimization, outperforming both verbalized-confidence and estimator-only

| | HotpotQA | | | | OOD Averaged | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 51.6% | 0.555 | 0.363 | 0.343 | 56.5% | 0.556 | 0.359 | 0.345 |
| ∟ RLVR | 62.2% | 0.523 | 0.376 | 0.376 | **62.3%** | 0.518 | 0.357 | 0.358 |
| ∟ Answer-Prob | 62.2% | 0.661 | 0.359 | 0.360 | **62.3%** | 0.548 | 0.340 | 0.327 |
| ∟ P(True) | 62.2% | 0.543 | 0.308 | 0.260 | **62.3%** | 0.588 | 0.304 | 0.314 |
| ∟ RLVR + Estimator | 62.2% | 0.757 | 0.195 | 0.087 | **62.3%** | 0.724 | 0.184 | 0.184 |
| ∟ RLCR | 61.0% | 0.596 | 0.244 | 0.100 | 61.4% | 0.646 | 0.204 | 0.171 |
| ∟ RLCR (Step-aligned) | **65.3%** | 0.606 | 0.263 | 0.205 | 61.6% | 0.690 | 0.210 | 0.188 |
| ∟ RLCR + Estimator | 61.0% | 0.753 | 0.201 | **0.081** | 61.4% | 0.719 | 0.207 | 0.224 |
| ∟ IPoET (ours) | 63.3% | **0.800** | **0.184** | 0.107 | **62.3%** | **0.738** | **0.170** | **0.146** |
| Llama-3.1-8B-Instruct | 45.9% | 0.577 | 0.461 | 0.469 | 45.9% | 0.606 | 0.413 | 0.424 |
| ∟ RLVR + Estimator | 61.6% | 0.772 | 0.198 | 0.110 | 51.0% | **0.746** | **0.180** | 0.161 |
| ∟ RLCR | 59.1% | 0.603 | 0.248 | 0.121 | 50.2% | 0.674 | 0.217 | 0.175 |
| ∟ IPoET (ours) | **64.6%** | **0.781** | **0.188** | **0.101** | **51.5%** | 0.737 | 0.182 | **0.152** |

| | Math | | | | OOD Averaged | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 62.8% | 0.568 | 0.313 | 0.290 | 51.9% | 0.562 | 0.389 | 0.382 |
| ∟ RLVR | **74.4%** | 0.517 | 0.255 | 0.255 | **52.9%** | 0.569 | 0.417 | 0.424 |
| ∟ Answer-Prob | **74.4%** | 0.709 | 0.235 | 0.239 | **52.9%** | 0.576 | 0.420 | 0.424 |
| ∟ P(True) | **74.4%** | 0.490 | 0.366 | 0.380 | **52.9%** | 0.645 | 0.275 | 0.237 |
| ∟ RLVR + Estimator | **74.4%** | 0.862 | 0.107 | **0.037** | **52.9%** | 0.684 | 0.217 | 0.197 |
| ∟ RLCR | 73.7% | 0.625 | 0.184 | 0.115 | 52.8% | 0.600 | 0.282 | 0.272 |
| ∟ RLCR (Step-aligned) | 73.8% | 0.633 | 0.181 | 0.120 | 41.7% | 0.536 | 0.361 | 0.347 |
| ∟ RLCR + Estimator | 73.7% | 0.828 | 0.126 | 0.068 | 52.8% | 0.662 | 0.249 | 0.249 |
| ∟ IPoET (ours) | 74.2% | **0.878** | **0.100** | 0.040 | 50.9% | **0.698** | **0.206** | **0.183** |
| Llama-3.1-8B-Instruct | 42.7% | 0.591 | 0.501 | 0.512 | 43.7% | 0.611 | 0.410 | 0.423 |
| ∟ RLVR + Estimator | 53.8% | 0.789 | 0.167 | 0.085 | **45.3%** | 0.616 | 0.235 | 0.216 |
| ∟ RLCR | 50.6% | 0.570 | 0.246 | 0.165 | 44.6% | 0.616 | **0.219** | **0.179** |
| ∟ IPoET (ours) | **56.5%** | **0.824** | **0.152** | **0.061** | 45.2% | **0.622** | 0.226 | 0.191 |

Table 1: Main results for models trained on HotpotQA and Big-Math. We report accuracy and confidence-estimation metrics, including AUROC, Brier score, and ECE, for Qwen and Llama backbones. For models trained on HotpotQA, we report results on HotpotQA and a six-dataset OOD average. For models trained on Big-Math, we report a Math average over MATH-500, GSM8K, and Big-Math, together with a five-dataset OOD average. Best and second-best results for all metrics under each backbone are marked in **bold** and underlined, respectively.

methods. Overall, IPoET achieves stronger AUROC and Brier score, competitive ECE, and accuracy comparable to the strongest RL baselines. For comparison with commercial frontier models, see Appendix D.

**IPoET generalizes well to out-of-domain settings and different backbone models.** Under both HotpotQA and Big-Math training, IPoET achieves the strongest averaged OOD confidence-estimation results on Qwen, indicating that our method does not merely fit dataset-specific confidence patterns but learns transferable cues for unseen tasks. The Llama results further show that this effect is not tied to a single backbone. Compared with the initial Llama model, IPoET improves all confidence-estimation metrics on both HotpotQA and Math evaluations. Although it does not achieve the best value on every OOD metric, it remains highly competitive under distribution shifts while showing superior in-setting calibration and task accuracy. Together, these results demonstrate the strong generalization ability of our method. Results for individual datasets are provided in Appendix E.

**IPoET's gains stem from iterative policy-estimator coupling rather than additional training alone.** RLCR (Step-aligned) uses the same number of training updates as IPoET, yet it does not yield consistent improvements and can even hurt out-of-domain performance under Big-Math training. This highlights the importance of iterative coupling between policy learning and hidden-state confidence estimation. By repeatedly updating the policy with estimator-derived feedback and then refreshing the estimator on the updated policy dis-

| Setting | Strategy | AUROC↑ | Brier↓ |
|---|---|---|---|
| In-domain | Joint | 0.773 | 0.209 |
| | Iterative | **0.800** (+0.027) | **0.184** (-0.025) |
| Out-of-domain | Joint | 0.737 | 0.180 |
| | Iterative | **0.738** (+0.001) | **0.170** (-0.010) |

Table 2: Comparison between joint and iterative training on HotpotQA. Iterative training improves in-domain and out-of-domain AUROC and Brier score. Values in parentheses denote changes relative to joint training.

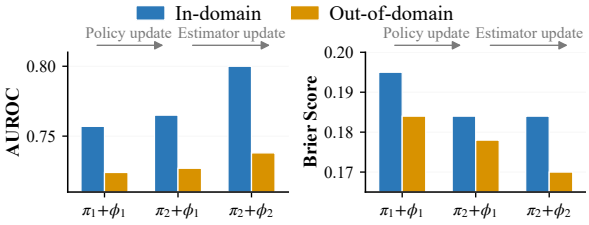


Figure 4: Step-wise effects of IPoET, showing that both policy and estimator updates progressively improve confidence calibration.

| Iteration Setting | In-domain | | Out-of-domain | |
|---|---|---|---|---|
| | AUROC↑ | Brier↓ | AUROC↑ | Brier↓ |
| 2×200 | **0.878** | **0.100** | **0.698** | 0.206 |
| 4×100 | 0.873 | 0.102 | 0.696 | **0.198** |

Table 3: Effect of alternating-update granularity under Big-Math training with the same update budget and data allocation. Best results are shown in **bold**.

tribution, IPoET makes the extra training budget more effective for confidence estimation.

### 4.3 Ablation and Analysis

**Joint vs. Iterative Training.** Table 2 compares IPoET with a joint-training variant that optimizes the policy and confidence estimator simultaneously. Across both evaluation settings, iterative training yields higher AUROC and lower Brier score than joint training, indicating gains in confidence discrimination and calibration. This suggests that simply combining the policy and estimator in one joint objective is insufficient. In joint training, the estimator must learn from representations produced by a changing policy while the policy is guided by an under-adapted estimator. This can make the confidence signal unstable and hurt confidence estimation. IPoET instead separates these roles across rounds: the estimator first adapts to current policy outputs, before being used to construct the confidence term in the next policy update, enabling more stable confidence-aware training.

**Step-wise Effects of Iterative Training.** We next examine how each stage of IPoET contributes to confidence estimation. As shown in Figure 4, both metrics consistently move in the desired direction throughout the iteration: the policy update already improves confidence quality across both in-domain and out-of-domain evaluations, and the subsequent estimator update further strengthens this trend, especially for out-of-domain calibration. These step-wise improvements suggest that the two updates play complementary roles. Policy optimization transfers the estimator signal into the model's generation behavior, whereas estimator refinement adapts confidence estimation to the updated policy distribution. This supports the iterative design of IPoET, where alternating policy and estimator updates form a more effective confidence-aware training loop than a one-shot update.

**Granularity of Alternating Updates.** A natural question is whether IPoET benefits from more frequent alternation between policy and estimator updates. To isolate this factor, we compare two schedules with the same update budget and data allocation: 2×200 uses two longer update phases, while 4×100 splits each corresponding phase into two shorter ones. Table 3 shows that more frequent alternation does not lead to better confidence estimation. The 2×200 schedule performs better on three of the four comparisons, including AUROC on both in-domain and out-of-domain evaluations and in-domain Brier score. This suggests that IPoET needs each update phase to be long enough for the policy and estimator to adapt to each other, rather than simply switching between them more frequently. When each phase becomes too short, the policy may not fully absorb the estimator feedback before the estimator is updated again, leading to less consistent gains in confidence estimation.

**Impact of Iteration Rounds.** We finally examine whether extending IPoET to additional rounds further improves confidence estimation. The trend in Figure 5 shows that the transition from Round 1 to Round 2 substantially improves both AUROC and Brier score on in-domain and out-of-domain evaluations, confirming the benefit of the first complete policy–estimator refinement cycle. The additional round brings only a marginal improvement in out-of-domain AUROC, while in-domain AUROC remains unchanged and Brier score slightly

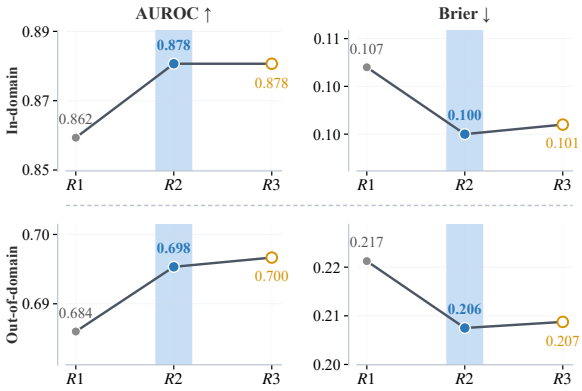


Figure 5: Round-wise comparison. Round 2 provides the best overall confidence estimation trade-off.

increases in both settings. This suggests that the policy and estimator largely stabilize after Round 2, leaving limited gains from further alternation. Therefore, we adopt Round 2 as the default setting, which offers the best overall trade-off between confidence discrimination and calibration quality.

## 5 Related Work

**Confidence Estimation from Implicit Information.** These methods estimate confidence from implicit information produced during inference or internal computation. One line of work studies model-internal representations for confidence estimation. For LLMs, hidden activations have been shown to encode factuality and truthfulness-related information (Chen et al., 2024), motivating lightweight probes or classifiers trained on internal states (Azaria and Mitchell, 2023; Servedio et al., 2025). Later work extends this view to reasoning settings, including hidden-state verification of reasoning steps and perturbation-based confidence probing (Zhang et al., 2025; Khanmohammadi et al., 2025). Another line derives implicit confidence from output probabilities or self-evaluation signals: Answer-Prob uses token likelihoods or answer probabilities as confidence proxies (Jiang et al., 2021; Gupta et al., 2024), while P(True) asks the model to judge whether its own answer is correct and uses the probability of a positive judgment (Kadavath et al., 2022). However, these signals are usually used to score fixed outputs, whereas IPoET converts internal-representation estimates into feedback for policy training.

**Verbalized Confidence Estimation.** Verbalized methods require the model to express confidence explicitly through natural language. Prompting-based approaches elicit numerical or linguistic confidence alongside generated answers or reasoning (Xiong et al., 2024; Yang et al., 2024). Training-based approaches further teach models to produce uncertainty rationales or calibrated confidence scores, such as self-reflective rationales in SaySelf (Xu et al., 2024), listener-aware confidence markers in LACIE (Stengel-Eskin et al., 2024), and calibration-aware on-policy distillation (Zhang et al., 2026). Studies of reasoning models show that chain-of-thought generation can improve confidence expression (Yoon et al., 2026), motivating recent RL-based methods that optimize verbalized confidence with calibration-oriented rewards, such as Brier-style or logarithmic scoring objectives (Damani et al., 2025; Bani-Harouni et al., 2025). However, expressed confidence remains sensitive to prompt formulation, answer-dependent self-assessment, and placement of confidence statements relative to the answer (Xia et al., 2025; Seo et al., 2025; Guo et al., 2026; Li et al., 2026a). IPoET avoids relying on such explicit confidence expression by deriving confidence from an internal-representation estimator, while still benefiting from reasoning-oriented generation.

## 6 Conclusion

This work revisits two major paradigms for LLM confidence estimation: estimator-based and verbalization-based methods. Contrary to previous findings, we demonstrate that estimator-based approaches can indeed outperform verbalization-based methods, despite the current prevalence of the latter. Nevertheless, traditional estimators fail to utilize the strong reasoning capabilities of contemporary LLMs. Based on these insights, we propose IPoET (Iterative Policy-Estimator Training), a framework that synergizes the two paradigms: the estimator distills richer features from hidden states by leveraging policy-generated reasoning traces, and the policy learns to produce more informative content for the estimator. Experiments across multiple reasoning and question-answering datasets show that IPoET outperforms both approaches for confidence estimation in both in-domain and out-of-domain settings, while maintaining competitive task performance. These results highlight the value of internal representations for reliable confidence estimation and suggest that iterative policy-estimator training is a promising direction for building more trustworthy reasoning models.

## Limitations

IPoET leverages the strengths of estimator- and verbalization-based approaches, but has several practical limitations. On one hand, it utilizes an independent estimator for confidence scoring. While this estimator ensures inference efficiency by predicting confidence in a single forward pass, deploying two distinct models incurs additional computational overhead. On the other hand, IPoET employs intensive reinforcement learning to optimize the reasoning path. This requires extended training time to achieve convergence while balancing the dual objectives of task performance and confidence estimation. Nevertheless, our validation demonstrates the promise of this paradigm. Future research should focus on developing lightweight integration strategies to enable more efficient applications.

## References


Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and 1 others. 2025. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models. *arXiv preprint arXiv:2502.17387*.

Dang Anh-Hoang, Vu Tran, and Le-Minh Nguyen. 2025. Survey and analysis of hallucinations in large language models: attribution to prompting strategies or model behavior. *Frontiers in Artificial Intelligence*, 8:1622292.

Anthropic. 2025. The claude 4 model family: Opus, sonnet, and haiku. https://www.anthropic.com/research/claude-haiku-4-5.

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

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

Mohammad Beigi, Ying Shen, Runing Yang, Zihao Lin, Qifan Wang, Ankith Mohan, Jianfeng He, Ming Jin, Chang-Tien Lu, and Lifu Huang. 2024. Internalinspector i2: Robust confidence estimation in llms through internal states. In *Findings of the association for computational linguistics: EMNLP 2024*, pages 12847–12865.

Michael Bereket and Jure Leskovec. 2025. Uncalibrated reasoning: Grpo induces overconfidence for stochastic outcomes. *arXiv preprint arXiv:2508.11800*.

Andrew P Bradley. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. 2025. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint arXiv:2507.16806*.

DeepSeek. 2026. Deepseek v4 preview release. https://api-docs.deepseek.com/news/news260424.

W Brier Glenn and 1 others. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Google DeepMind. 2025. Gemini 3 flash. https://deepmind.google/models/gemini/flash/. Model description page.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

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

Junyu Guo, Shangding Gu, Ming Jin, Costas Spanos, and Javad Lavaei. 2026. Llms should express uncertainty explicitly. *arXiv preprint arXiv:2604.05306*.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language model cascades: Token-level uncertainty and beyond. *arXiv preprint arXiv:2404.10136*.

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

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2026. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Advances in Neural Information Processing Systems*, 38:162239–162262.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, volume 2024, pages 54107–54157.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Reza Khanmohammadi, Erfan Miahi, Mehrsa Mardikoraem, Simerjot Kaur, Ivan Brugere, Charese Smiley, Kundan S Thind, and Mohammad M Ghassemi. 2025. Calibrating llm confidence by probing perturbed representation stability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10459–10525.

Dharshan Kumaran, Arthur Conmy, Federico Barbero, Simon Osindero, Viorica Patraucean, and Petar Velickovic. 2026a. How do llms compute verbal confidence. *arXiv preprint arXiv:2603.17839*.

Dharshan Kumaran, Stephen M Fleming, Larisa Markeeva, Joe Heyward, Andrea Banino, Mrinal Mathur, Razvan Pascanu, Simon Osindero, Benedetto De Martino, Petar Veličković, and 1 others. 2026b. Competing biases underlie overconfidence and underconfidence in llms. *Nature Machine Intelligence*, pages 1–14.

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

Junkai Li, Yunghwei Lai, Weitao Li, Jingyi Ren, Meng Zhang, Xinhui Kang, Siyu Wang, Peng Li, Ya-Qin Zhang, Weizhi Ma, and 1 others. 2024. Agent hospital: A simulacrum of hospital with evolvable medical agents. *arXiv preprint arXiv:2405.02957*.

Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. 2026b. Conftuner: Training large language models to express their confidence verbally. *Advances in Neural Information Processing Systems*, 38:53484–53513.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 374–382.

Matéo Mahaut, Laura Aina, Paula Czarnowska, Momchil Hardalov, Thomas Müller, and Lluís Màrquez. 2024. Factual confidence of llms: on reliability and robustness of current estimators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4554–4570.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. 2023. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075.

Shiyu Ni, Keping Bi, Jiafeng Guo, Minghao Tang, Jingtong Wu, Zengxin Han, and Xueqi Cheng. 2025. Annotation-efficient universal honesty alignment. *arXiv preprint arXiv:2510.17509*.

OpenAI. 2025. Gpt-5 system card. https://openai.com/index/gpt-5-system-card.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Ki Jung Seo, Sehun Lim, and Taeuk Kim. 2025. Advice: Answer-dependent verbalized confidence estimation. *arXiv preprint arXiv:2510.10913*.

Giovanni Servedio, Alessandro De Bellis, Dario Di Palma, Vito Walter Anelli, and Tommaso Di Noia. 2025. Are the hidden states hiding something? testing the limits of factuality-encoding capabilities in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6089–6104.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical

reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

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

Marco Siino, Mariana Falco, Daniele Croce, and Paolo Rosso. 2025. Exploring llms applications in law: A literature review on current legal nlp approaches. *IEEE Access*, 13:18253–18276.

Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. 2024. Lacie: Listener-aware finetuning for calibration in large language models. *Advances in Neural Information Processing Systems*, 37:43080–43106.

Nishant Subramani, Jason Eisner, Justin Svegliato, Benjamin Van Durme, Yu Su, and Sam Thomson. 2025. Mice for cats: Model-internal confidence estimation for calibrating agents with tools. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12362–12375.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Shuchang Tao, Liuyi Yao, Hanxing Ding, Yuexiang Xie, Qi Cao, Fei Sun, Jinyang Gao, Huawei Shen, and Bolin Ding. 2024. When to trust llms: Aligning confidence with response quality. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5984–5996.

Benjamin Turtel, Danny Franklin, Kris Skotheim, Luke Hewitt, and Philipp Schoenegger. 2025. Outcome-based reinforcement learning to predict the future. *arXiv preprint arXiv:2505.17989*.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Yuxi Xia, Pedro Henrique Luz De Araujo, Klim Zaporojets, and Benjamin Roth. 2025. Influences on llm calibration: A study of response agreement, loss functions, and prompt styles. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3740–3761.

Ruixuan Xiao, Wentao Ma, Ke Wang, Yuchuan Wu, Junbo Zhao, Haobo Wang, Fei Huang, and Yongbin Li. 2024. Flowbench: Revisiting and benchmarking workflow-guided planning for llm-based agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10883–10900.

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

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

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

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Dongkeun Yoon, Seungone Kim, Sohee Yang, Sunkyoung Kim, Soyeon Kim, Yongil Kim, Eunbi Choi, Yireun Kim, and Minjoon Seo. 2026. Reasoning models better express their confidence. *Advances in Neural Information Processing Systems*, 38:103869–103896.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*.

Jiaxin Zhang, Xiangyu Peng, Qinglin Chen, Qinyuan Ye, Caiming Xiong, and Chien-Sheng Wu. 2026. The illusion of certainty: Decoupling capability and calibration in on-policy distillation. *arXiv preprint arXiv:2604.16830*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

## A Use of Large Language Models

We use Large Language Models to aid or polish writing.

## B Training Details

All experiments are conducted on a server equipped with 8 NVIDIA A100 GPUs, each with 80GB memory. For both HotpotQA and Big-Math, the overall training time is approximately 15 hours.

For policy training, we use the same hyperparameters in both rounds. Specifically, we use GRPO with 8 sampled responses per prompt and a sampling temperature of 0.7. The policy learning rate is set to 1e-6 for both HotpotQA and Big-Math. In line with Damani et al. (2025), we use a training batch size of 64 for HotpotQA and 72 for Big-Math. The maximum prompt and response lengths are set to 3072 and 1536 for HotpotQA, and 1024 and 4096 for Big-Math, respectively. We do not normalize GRPO advantages by their group standard deviation, following prior practice (Bereket and Leskovec, 2025; Turtel et al., 2025), thereby preserving reward-scale information for examples with large calibration errors.

For estimator training, we use a per-device batch size of 1 with gradient accumulation over 8 steps. We reduce the second-round learning rate to support stable refinement and avoid overfitting or excessive drift from the first-round estimator. The estimator learning rates are 5e-6 and 1e-6 for HotpotQA, and 5e-6 and 5e-7 for Big-Math, in the first and second rounds, respectively.

The prompt template used for policy training is shown in Table 4.

| |
|---|
| 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 then provides the user with the answer. The reasoning process and answer are enclosed within <reason> </reason> and <answer> </answer> tags, respectively, i.e., <reason> reasoning process here </reason><answer> answer here </answer>. |

Table 4: Policy training prompt template.

## C Details of Evaluation Metrics

We follow the notation defined in the main text.

**Accuracy.** We compute accuracy using dataset-specific correctness criteria. For HotpotQA and HotpotQA-Modified, we use exact match. For GSM8K, MATH-500, and Big-Math, we use the *math_verify* library. For CommonsenseQA, GPQA, SimpleQA, and TriviaQA, we use an LLM-as-a-judge (Zheng et al., 2023) evaluation. To keep the judge model family separate from the evaluated backbone, we use Llama-3.1-8B-Instruct for Qwen-based methods and Qwen3-8B for Llama-based methods. Following Damani et al. (2025), we run the judge with temperature set to 0 and provide it with the question, the ground-truth answer, and the extracted answer. The judge is instructed to output only "YES" or "NO" according to whether the answer is correct. Since these datasets contain short and objective answers, we do not condition the judge on thinking traces, avoiding potential bias from generated rationales.

**AUROC.** We compute AUROC by sweeping the confidence threshold and integrating the resulting ROC curve:

$$\mathrm{AUROC} = \int_0^1 \mathrm{TPR}\left(\mathrm{FPR}^{-1}(x)\right)\, dx,$$

where TPR and FPR denote the true positive rate and false positive rate, respectively.

**Brier score.** Given confidence scores $c_i$ and correctness labels $z_i$, we compute

$$\text{Brier score} = \frac{1}{N}\sum_{i=1}^{N}(c_i - z_i)^2.$$

**ECE.** We partition predictions into $M = 10$ confidence bins $\{B_m\}_{m=1}^{M}$. For each bin, we compute the average correctness and average confidence:

$$\bar{z}_m = \frac{1}{|B_m|}\sum_{i\in B_m} z_i.$$

$$\bar{c}_m = \frac{1}{|B_m|}\sum_{i\in B_m} c_i.$$

The ECE is then computed as

$$\mathrm{ECE} = \sum_{m=1}^{M}\frac{|B_m|}{N}\left|\bar{z}_m - \bar{c}_m\right|.$$

| | HotpotQA | | | | HotpotQA-Modified | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 51.6% | 0.555 | 0.363 | 0.343 | 34.6% | 0.765 | 0.347 | 0.376 |
| ∟ RLVR | 62.2% | 0.523 | 0.376 | 0.376 | 46.8% | 0.523 | 0.527 | 0.528 |
| ∟ Answer-Prob | 62.2% | 0.661 | 0.359 | 0.360 | 46.8% | 0.600 | 0.502 | 0.507 |
| ∟ P(True) | 62.2% | 0.543 | 0.308 | 0.260 | 46.8% | 0.677 | 0.330 | 0.305 |
| ∟ RLVR + Estimator | 62.2% | 0.757 | 0.195 | 0.087 | 46.8% | **0.853** | **0.159** | **0.059** |
| ∟ RLCR | 61.0% | 0.596 | 0.244 | 0.100 | 44.2% | 0.739 | 0.207 | 0.067 |
| ∟ RLCR (Step-aligned) | **65.3%** | 0.606 | 0.263 | 0.205 | **47.2%** | 0.788 | 0.223 | 0.173 |
| ∟ RLCR + Estimator | 61.0% | 0.753 | 0.201 | **0.081** | 44.2% | 0.850 | 0.163 | 0.076 |
| ∟ IPoET (ours) | 63.3% | **0.800** | **0.184** | 0.107 | 45.2% | 0.852 | 0.174 | 0.118 |
| Llama-3.1-8B-Instruct | 45.9% | 0.577 | 0.461 | 0.469 | 33.2% | 0.772 | 0.429 | 0.473 |
| ∟ RLVR + Estimator | 61.6% | 0.772 | 0.198 | 0.110 | 48.2% | **0.844** | **0.169** | **0.079** |
| ∟ RLCR | 59.1% | 0.603 | 0.248 | 0.121 | 45.0% | 0.770 | 0.205 | 0.116 |
| ∟ IPoET (ours) | **64.6%** | **0.781** | **0.188** | **0.101** | **51.2%** | 0.831 | 0.177 | 0.083 |

Table 5: Performance on HotpotQA and HotpotQA-Modified for models trained on HotpotQA. Best and second-best values are marked in **bold** and underlined, respectively.

**(a) HotpotQA**

| Model | AUROC↑ | Brier↓ | ECE↓ |
|---|---|---|---|
| GPT-5 mini | 0.603 | 0.306 | 0.298 |
| Claude Haiku 4.5 | 0.671 | 0.279 | 0.266 |
| DeepSeek-V4-Flash | 0.537 | 0.332 | 0.328 |
| Gemini 3 Flash | 0.509 | 0.357 | 0.357 |
| IPoET (ours) | **0.800** | **0.184** | **0.107** |

**(b) Math Average**

| Model | AUROC↑ | Brier↓ | ECE↓ |
|---|---|---|---|
| GPT-5 mini | 0.638 | 0.189 | 0.181 |
| Claude Haiku 4.5 | 0.584 | 0.309 | 0.300 |
| DeepSeek-V4-Flash | 0.686 | 0.137 | 0.136 |
| Gemini 3 Flash | 0.557 | 0.150 | 0.150 |
| IPoET (ours) | **0.878** | **0.100** | **0.040** |

Table 6: Comparison between IPoET and commercial models. IPoET is trained on HotpotQA for the HotpotQA results and on Big-Math for the Math average over MATH-500, GSM8K, and Big-Math. Best and second-best results are marked in **bold** and underlined, respectively.

# D Commercial Model Comparison

We further compare IPoET with commercial models in Table 6. We evaluate GPT-5 mini (OpenAI, 2025), Claude Haiku 4.5 (Anthropic, 2025), DeepSeek-V4-Flash (DeepSeek, 2026), and Gemini 3 Flash (Google DeepMind, 2025).[1] For the HotpotQA results, IPoET is trained on HotpotQA; for the math results, IPoET is trained on Big-Math and evaluated using the Math average over MATH-500, GSM8K, and Big-Math. Across both settings, IPoET obtains the best results on AUROC, Brier score, and ECE among the evaluated models. Commercial models demonstrate reasonable ability to estimate confidence, but their verbalized confidence remains less reliable in these settings. These results indicate that strong general-purpose models can still struggle to align expressed confidence with correctness on reasoning tasks, highlighting the need for task-oriented training objectives that explicitly target confidence reliability.

[1] Model versions used: `gpt-5-mini-2025-08-07`, `claude-haiku-4-5-20251001`, `deepseek-v4-flash`, and `gemini-3-flash-preview-nothinking`.

# E Results

## E.1 Models Trained on HotpotQA

We report detailed results for models trained on HotpotQA across individual evaluation datasets. Results on HotpotQA and HotpotQA-Modified are shown in Table 5, CommonsenseQA and GPQA in Table 7, GSM8K and MATH-500 in Table 8, and SimpleQA and TriviaQA in Table 9.

## E.2 Models Trained on Big-Math

We report detailed results for models trained on Big-Math across individual evaluation datasets. Results on Big-Math and HotpotQA are shown in Table 10, CommonsenseQA and GPQA in Table 11, GSM8K and MATH-500 in Table 12, and SimpleQA and TriviaQA in Table 13.

| | CommonsenseQA | | | | GPQA | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 89.5% | 0.612 | 0.090 | **0.016** | 47.1% | 0.533 | 0.418 | 0.413 |
| ∟ RLVR | 92.5% | 0.502 | 0.074 | 0.060 | **52.0%** | 0.451 | 0.454 | 0.450 |
| ∟ Answer-Prob | 92.5% | 0.551 | **0.072** | 0.028 | **52.0%** | 0.488 | 0.422 | 0.408 |
| ∟ P(True) | 92.5% | 0.727 | 0.352 | 0.486 | **52.0%** | 0.532 | 0.276 | 0.152 |
| ∟ RLVR + Estimator | 92.5% | **0.809** | 0.185 | 0.346 | **52.0%** | 0.542 | 0.275 | 0.125 |
| ∟ RLCR | 89.0% | 0.671 | 0.144 | 0.223 | 47.3% | 0.617 | 0.254 | 0.130 |
| ∟ RLCR (Step-aligned) | 91.4% | 0.759 | 0.098 | 0.152 | 45.3% | 0.600 | 0.293 | 0.220 |
| ∟ RLCR + Estimator | 89.0% | 0.775 | 0.268 | 0.419 | 47.3% | **0.629** | **0.253** | **0.124** |
| ∟ IPoET (ours) | **92.9%** | 0.797 | 0.126 | 0.234 | 48.9% | 0.579 | 0.269 | 0.135 |
| Llama-3.1-8B-Instruct | 73.0% | 0.651 | 0.212 | 0.161 | 27.7% | **0.587** | 0.478 | 0.510 |
| ∟ RLVR + Estimator | **75.4%** | **0.754** | 0.216 | 0.238 | 31.9% | 0.560 | 0.237 | 0.135 |
| ∟ RLCR | 74.1% | 0.660 | **0.186** | **0.079** | **35.0%** | 0.538 | 0.279 | 0.206 |
| ∟ IPoET (ours) | 74.9% | 0.753 | 0.212 | 0.228 | 31.0% | 0.576 | **0.231** | **0.122** |

Table 7: Performance on CommonsenseQA and GPQA for models trained on HotpotQA. Best and second-best values are marked in **bold** and underlined, respectively.

| | GSM8K | | | | MATH-500 | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 84.4% | 0.494 | 0.137 | 0.072 | 48.2% | 0.611 | 0.435 | 0.440 |
| ∟ RLVR | 88.8% | 0.542 | 0.111 | 0.110 | 64.8% | 0.514 | 0.351 | 0.351 |
| ∟ Answer-Prob | 88.8% | 0.636 | 0.105 | 0.082 | 64.8% | 0.742 | 0.304 | 0.302 |
| ∟ P(True) | 88.8% | 0.470 | 0.433 | 0.545 | 64.8% | 0.385 | 0.319 | 0.271 |
| ∟ RLVR + Estimator | 88.8% | 0.739 | 0.154 | 0.257 | 64.8% | **0.791** | 0.184 | 0.113 |
| ∟ RLCR | 87.9% | 0.585 | 0.120 | 0.126 | **67.2%** | 0.679 | 0.203 | 0.083 |
| ∟ RLCR (Step-aligned) | 87.6% | 0.674 | **0.098** | **0.024** | 67.0% | 0.723 | 0.209 | 0.153 |
| ∟ RLCR + Estimator | 87.9% | 0.696 | 0.190 | 0.307 | **67.2%** | 0.767 | 0.203 | 0.201 |
| ∟ IPoET (ours) | **89.2%** | **0.788** | 0.155 | 0.265 | 65.6% | 0.772 | **0.175** | **0.056** |
| Llama-3.1-8B-Instruct | 69.4% | 0.569 | 0.285 | 0.283 | 33.0% | 0.590 | 0.602 | 0.614 |
| ∟ RLVR + Estimator | **82.3%** | **0.783** | 0.183 | 0.251 | 42.0% | **0.788** | **0.196** | 0.087 |
| ∟ RLCR | 81.0% | 0.668 | **0.144** | **0.066** | 38.8% | 0.683 | 0.330 | 0.336 |
| ∟ IPoET (ours) | 82.0% | 0.730 | 0.205 | 0.272 | **44.4%** | 0.763 | 0.206 | **0.085** |

Table 8: Performance on GSM8K and MATH-500 for models trained on HotpotQA. Best and second-best values are marked in **bold** and underlined, respectively.

| | SimpleQA | | | | TriviaQA | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 11.2% | 0.501 | 0.715 | 0.781 | 58.6% | 0.583 | 0.358 | 0.345 |
| ∟ RLVR | 12.9% | 0.519 | 0.804 | 0.832 | 63.0% | 0.577 | 0.350 | 0.346 |
| ∟ Answer-Prob | 12.9% | 0.432 | 0.794 | 0.819 | 63.0% | 0.438 | 0.342 | 0.321 |
| ∟ P(True) | 12.9% | 0.623 | 0.255 | 0.318 | 63.0% | 0.793 | 0.191 | 0.113 |
| ∟ RLVR + Estimator | 12.9% | 0.616 | 0.129 | 0.120 | 63.0% | 0.846 | 0.178 | 0.145 |
| ∟ RLCR | 12.4% | 0.602 | 0.304 | 0.388 | 64.4% | 0.721 | 0.199 | **0.073** |
| ∟ RLCR (Step-aligned) | 12.0% | 0.624 | 0.350 | 0.429 | **66.5%** | 0.758 | 0.209 | 0.151 |
| ∟ RLCR + Estimator | 12.4% | 0.635 | **0.122** | 0.088 | 64.4% | 0.810 | 0.208 | 0.205 |
| ∟ IPoET (ours) | **14.2%** | **0.640** | 0.132 | **0.087** | 62.7% | **0.849** | **0.162** | 0.101 |
| Llama-3.1-8B-Instruct | 6.6% | 0.590 | 0.634 | 0.742 | 65.4% | 0.648 | 0.264 | 0.232 |
| ∟ RLVR + Estimator | 7.0% | 0.736 | 0.091 | 0.130 | 67.1% | **0.853** | 0.156 | 0.122 |
| ∟ RLCR | 5.6% | 0.718 | 0.193 | 0.331 | 66.7% | 0.779 | 0.170 | **0.032** |
| ∟ IPoET (ours) | **7.7%** | **0.748** | **0.086** | **0.099** | **68.8%** | 0.850 | **0.152** | 0.107 |

Table 9: Performance on SimpleQA and TriviaQA for models trained on HotpotQA. Best and second-best values are marked in **bold** and underlined, respectively.

| | Big-Math | | | | HotpotQA | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 55.7% | 0.599 | 0.367 | 0.358 | 53.0% | 0.582 | 0.363 | 0.356 |
| ∟ RLVR | 70.2% | 0.519 | 0.294 | 0.294 | **54.3%** | 0.551 | 0.437 | 0.440 |
| ∟ Answer-Prob | 70.2% | 0.633 | 0.281 | 0.284 | **54.3%** | 0.604 | 0.415 | 0.416 |
| ∟ P(True) | 70.2% | 0.556 | 0.293 | 0.257 | **54.3%** | 0.546 | 0.318 | 0.240 |
| ∟ RLVR + Estimator | 70.2% | 0.857 | 0.135 | 0.026 | **54.3%** | 0.677 | 0.252 | 0.173 |
| ∟ RLCR | 69.7% | 0.660 | 0.200 | 0.059 | 54.2% | 0.564 | 0.288 | 0.211 |
| ∟ RLCR (Step-aligned) | 70.3% | 0.640 | 0.199 | 0.061 | 49.5% | 0.544 | 0.325 | 0.277 |
| ∟ RLCR + Estimator | 69.7% | 0.858 | 0.139 | 0.035 | 54.2% | 0.642 | 0.277 | 0.215 |
| ∟ IPoET (ours) | **72.2%** | **0.865** | **0.129** | **0.020** | 52.4% | **0.703** | **0.245** | **0.165** |
| Llama-3.1-8B-Instruct | 25.6% | 0.615 | 0.615 | 0.640 | 45.9% | **0.577** | 0.461 | 0.469 |
| ∟ RLVR + Estimator | 36.5% | 0.769 | 0.192 | **0.053** | **50.1%** | 0.570 | 0.267 | 0.153 |
| ∟ RLCR | 33.3% | 0.556 | 0.224 | 0.070 | 49.5% | 0.554 | **0.252** | **0.059** |
| ∟ IPoET (ours) | **41.1%** | **0.803** | **0.180** | 0.055 | 48.8% | 0.570 | 0.255 | 0.098 |

Table 10: Performance on Big-Math and HotpotQA for models trained on Big-Math. Best and second-best values are marked in **bold** and underlined, respectively.

| | CommonsenseQA | | | | GPQA | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 89.5% | 0.612 | **0.090** | **0.016** | **47.1%** | 0.533 | 0.418 | 0.413 |
| ∟ RLVR | 89.8% | 0.578 | 0.094 | 0.068 | 44.9% | 0.518 | 0.518 | 0.521 |
| ∟ Answer-Prob | 89.8% | 0.617 | 0.096 | 0.081 | 44.9% | 0.520 | 0.501 | 0.502 |
| ∟ P(True) | 89.8% | 0.672 | 0.446 | 0.570 | 44.9% | 0.526 | 0.270 | 0.121 |
| ∟ RLVR + Estimator | 89.8% | **0.805** | 0.177 | 0.313 | 44.9% | 0.592 | 0.256 | 0.100 |
| ∟ RLCR | 89.9% | 0.639 | 0.119 | 0.175 | 44.4% | **0.608** | 0.297 | 0.245 |
| ∟ RLCR (Step-aligned) | 75.1% | 0.545 | 0.187 | 0.030 | 31.5% | 0.575 | 0.373 | 0.399 |
| ∟ RLCR + Estimator | 89.9% | 0.765 | 0.288 | 0.453 | 44.4% | 0.607 | 0.261 | 0.145 |
| ∟ IPoET (ours) | **90.3%** | 0.772 | 0.168 | 0.297 | 43.8% | 0.605 | **0.248** | **0.097** |
| Llama-3.1-8B-Instruct | 73.0% | **0.651** | **0.212** | **0.161** | 27.7% | 0.587 | 0.478 | 0.510 |
| ∟ RLVR + Estimator | 75.1% | 0.601 | 0.250 | 0.261 | 29.5% | **0.591** | 0.224 | 0.107 |
| ∟ RLCR | 74.9% | 0.586 | 0.226 | 0.207 | 26.6% | 0.584 | **0.223** | 0.167 |
| ∟ IPoET (ours) | **75.3%** | 0.612 | 0.227 | 0.222 | **31.7%** | 0.568 | 0.235 | **0.094** |

Table 11: Performance on CommonsenseQA and GPQA for models trained on Big-Math. Best and second-best values are marked in **bold** and underlined, respectively.

| | GSM8K | | | | MATH-500 | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 84.4% | 0.494 | 0.137 | 0.072 | 48.2% | 0.611 | 0.435 | 0.440 |
| ∟ RLVR | 92.3% | 0.532 | 0.076 | 0.076 | 60.6% | 0.500 | 0.394 | 0.394 |
| ∟ Answer-Prob | 92.3% | 0.657 | 0.075 | 0.073 | 60.6% | 0.838 | 0.349 | 0.361 |
| ∟ P(True) | 92.3% | 0.515 | 0.489 | 0.631 | 60.6% | 0.400 | 0.317 | 0.252 |
| ∟ RLVR + Estimator | 92.3% | 0.834 | 0.058 | **0.030** | 60.6% | 0.896 | 0.129 | **0.055** |
| ∟ RLCR | 90.3% | 0.564 | 0.102 | 0.121 | **61.0%** | 0.652 | 0.249 | 0.164 |
| ∟ RLCR (Step-aligned) | 92.0% | 0.613 | 0.085 | 0.115 | 59.2% | 0.646 | 0.260 | 0.183 |
| ∟ RLCR + Estimator | 90.3% | 0.768 | 0.074 | 0.064 | **61.0%** | 0.859 | 0.165 | 0.105 |
| ∟ IPoET (ours) | **92.9%** | **0.849** | **0.053** | 0.031 | 57.4% | **0.919** | **0.119** | 0.070 |
| Llama-3.1-8B-Instruct | 69.4% | 0.569 | 0.285 | 0.283 | 33.0% | 0.590 | 0.602 | 0.614 |
| ∟ RLVR + Estimator | 82.8% | 0.807 | 0.124 | 0.121 | 42.2% | 0.790 | 0.185 | 0.081 |
| ∟ RLCR | 81.4% | 0.551 | 0.288 | 0.369 | 37.2% | 0.604 | 0.225 | 0.056 |
| ∟ IPoET (ours) | **83.3%** | **0.844** | **0.108** | **0.081** | **45.2%** | **0.826** | **0.169** | **0.047** |

Table 12: Performance on GSM8K and MATH-500 for models trained on Big-Math. Best and second-best values are marked in **bold** and underlined, respectively.

| | SimpleQA | | | | TriviaQA | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ | Acc.↑ | AUROC↑ | Brier↓ | ECE↓ |
| Qwen3-8B-Base | 11.2% | 0.501 | 0.715 | 0.781 | 58.6% | 0.583 | 0.358 | 0.345 |
| ∟ RLVR | 12.3% | 0.605 | 0.698 | 0.756 | **63.1%** | 0.591 | 0.338 | 0.333 |
| ∟ Answer-Prob | 12.3% | 0.555 | 0.759 | 0.804 | **63.1%** | 0.584 | 0.327 | 0.317 |
| ∟ P(True) | 12.3% | **0.662** | **0.164** | **0.183** | **63.1%** | **0.817** | **0.176** | **0.071** |
| ∟ RLVR + Estimator | 12.3% | 0.580 | 0.195 | 0.266 | **63.1%** | 0.766 | 0.207 | 0.134 |
| ∟ RLCR | **12.6%** | 0.556 | 0.465 | 0.583 | 63.0% | 0.635 | 0.239 | 0.147 |
| ∟ RLCR (Step-aligned) | 9.3% | 0.538 | 0.512 | 0.634 | 43.0% | 0.476 | 0.408 | 0.393 |
| ∟ RLCR + Estimator | **12.6%** | 0.547 | 0.206 | 0.289 | 63.0% | 0.748 | 0.213 | 0.141 |
| ∟ IPoET (ours) | 10.3% | 0.618 | 0.171 | 0.251 | 57.6% | 0.790 | 0.199 | 0.107 |
| Llama-3.1-8B-Instruct | 6.6% | 0.590 | 0.634 | 0.742 | 65.4% | 0.648 | 0.264 | 0.232 |
| ∟ RLVR + Estimator | **6.7%** | 0.596 | 0.213 | 0.368 | 64.9% | 0.724 | 0.221 | 0.190 |
| ∟ RLCR | 6.4% | **0.671** | **0.190** | **0.339** | **65.5%** | 0.684 | **0.205** | **0.125** |
| ∟ IPoET (ours) | 6.2% | 0.612 | **0.190** | 0.344 | 64.2% | **0.749** | 0.224 | 0.198 |

Table 13: Performance on SimpleQA and TriviaQA for models trained on Big-Math. Best and second-best values are marked in **bold** and underlined, respectively.