EPICAR: Knowing What You Don't Know Matters for Better Reasoning in LLMs

Jewon Yeom[1] **Jaewon Sok**[2] **Seonghyeon Park**[3] **Jeongjae Park**[1] **Taesup Kim**[1,*]
[1]Graduate School of Data Science, Seoul National University
[2]Department of Rural Systems Engineering, Seoul National University
[3]Department of Aerospace Engineering, Seoul National University

Abstract

Improving the reasoning abilities of large language models (LLMs) has largely relied on iterative self-training with model-generated data. While effective at boosting accuracy, existing approaches primarily reinforce successful reasoning paths, incurring a substantial *calibration cost*: models become overconfident and lose the ability to represent uncertainty. This failure has been characterized as a form of *model collapse* in alignment, where predictive distributions degenerate toward low-variance point estimates. We address this issue by reframing open-ended reasoning training as an epistemic learning problem, in which models must learn not only how to reason, but also when their reasoning should be trusted. We propose *epistemically-calibrated reasoning* (EPICAR) as a training objective that jointly optimizes reasoning performance and calibration, and instantiate it within an iterative supervised fine-tuning framework using explicitly extracted meta-cognitive self-evaluation signals. Experiments on Llama-3 and Qwen-3 families demonstrate that our approach achieves Pareto-superiority over standard baselines in both accuracy and calibration, particularly in models with sufficient reasoning capacity (e.g., 3B+). This framework generalizes effectively to OOD mathematical reasoning (GSM8K) and code generation (MBPP). Ultimately, our approach enables a 3× reduction in the overall inference compute budget, matching the $K = 30$ majority-vote performance of STaR with only $K = 10$ confidence-weighted samples, entirely without the multi-model overhead of external verifiers.

1 Introduction

The advent of Large Language Models (LLMs) has revolutionized complex, open-ended reasoning tasks such as mathematics and logic. Techniques like Chain-of-Thought (CoT) prompting

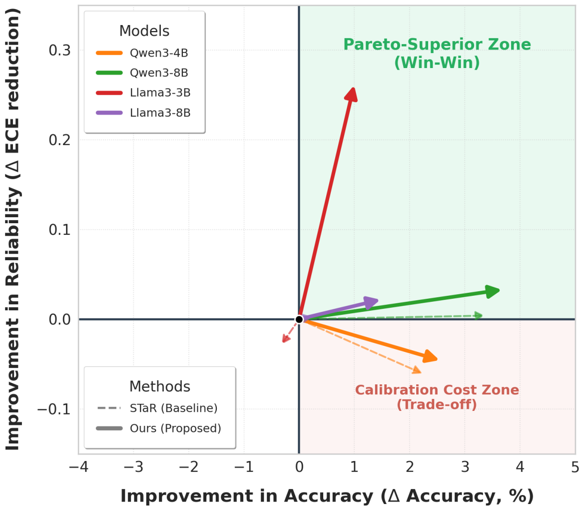

Figure 1: **Pareto-Superior Improvement in Reasoning and Reliability.** We visualize the relative improvement in reasoning accuracy (Δ Accuracy, %) and reliability (Δ ECE reduction) compared to the base model (at the origin). Solid and dashed arrows represent the trajectories of our proposed EPICAR and the STaR baseline, respectively. While standard iterative SFT often incurs a trade-off (Calibration Cost Zone), our method consistently drives diverse model families (Llama-3 and Qwen-3) into the *Pareto-Superior Zone*, achieving simultaneous gains in both task performance and uncertainty calibration.

(Wei et al., 2022) have significantly boosted performance by decomposing problems into intermediate steps. However, a critical challenge remains: the discrepancy between a model's accuracy and its self-knowledge, or *calibration* (Kadavath et al., 2022; Guo et al., 2017). Current models frequently exhibit confident misalignment, hallucinating plausible-sounding but incorrect answers with high certainty (OpenAI, 2023; Lin et al., 2022). For example, a model may fail at a final arithmetic step while assigning near-certainty to its erroneous conclusion. Such behavior highlights a systematic failure to represent uncertainty about reasoning outcomes, raising critical concerns for high-stakes ap-

* Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22414–22443

plications where knowing when a model should not be trusted is as important as producing a correct answer.

In the pursuit of higher reasoning accuracy, iterative self-training methods such as STaR (Zelikman et al., 2022) and ReST (Gulcehre et al., 2023) have become a dominant paradigm, reinforcing successful reasoning paths through a ground-truth verifier. While effective for boosting raw accuracy, we argue that this positive-only feedback loop imposes a significant *calibration cost* (Hu et al., 2025) that is often overlooked. The outcome of exclusively reinforcing correct paths is a manifestation of *Model Collapse* (Shumailov et al., 2023), where the model's predictive behavior converges toward low-variance point estimates, reinforcing its own biased beliefs while discarding the distributional tails necessary for reliable uncertainty estimation.

Recent advancements in reasoning models like DeepSeek-R1 (DeepSeek-AI et al., 2025) have addressed reasoning performance through reinforcement learning (RL) frameworks like GRPO. Evidence suggests that such models can express their confidence more accurately by engaging in *Slow Thinking* behaviors such as self-verification and backtracking within an extended CoT (Yoon et al., 2025). Furthermore, Zhang et al. (2025) found that models encode correctness signals within their hidden states, yet they often suffer from *overthinking* and fail to exploit this information during standard generation. While effective, the reliability of these models is strictly tied to the computational overhead of increased inference-time compute.

While inference-time scaling offers a partial solution, it does not fundamentally resolve the underlying miscalibration of the base policy. In this work, we address this issue by reframing reasoning training as an epistemic learning problem, where models must learn to reason while simultaneously discerning the reliability of their own outputs. We propose *epistemically-calibrated reasoning* (EPICAR), a training objective that jointly optimizes reasoning performance and calibration. By instantiating EPICAR within an iterative SFT framework using explicit meta-cognitive self-evaluation signals, we enable models to navigate the trade-off between accuracy and overconfidence. As shown in Figure 1, our framework drives models into the *Pareto-Superior Zone*. Crucially, by leveraging this internalized confidence for weighted ensembling, EPICAR matches high-sample-count ($K = 30$) performance with significantly fewer samples ($K = 10$), yielding an effective $3\times$ reduction in the total inference compute budget without relying on the multi-model architectural overhead of external verifiers.

2 Related Work

Iterative Reasoning & Self-Improvement Self-training techniques like STaR (Zelikman et al., 2022) and ReST (Gulcehre et al., 2023) bootstrap reasoning capabilities by fine-tuning on self-generated correct paths. To address the data-inefficiency of discarding incorrect attempts, V-STaR (Hosseini et al., 2024) utilizes both correct and incorrect solutions to train a verifier that judges the correctness of generated solutions. Recent approaches explicitly target the model's metacognition; for instance, MASA (Kim et al., 2025) aligns the model's predictions of solution attributes (e.g., difficulty, length) with actual rollout statistics to enhance training efficiency. While V-STaR requires training and deploying a separate verifier model—incurring massive multi-model inference overhead (see Table 7)—and MASA focuses on metadata for efficient gating, our method addresses the underlying *calibration cost* (Hu et al., 2025) and *Model Collapse* (Shumailov et al., 2023) by internalizing the evaluation process directly into the generator's objective.

Calibration Cost & Alignment Tax Conventional studies on the *alignment tax* have almost exclusively focused on the degradation of general task performance and accuracy during the alignment process (Lu et al., 2024; Lin et al., 2024; Fu et al., 2024). However, Hu et al. (2025) identify a more pervasive side effect: the *calibration cost*. They argue that while the alignment tax on capability often yields mixed or inconsistent results across different benchmarks, the cost to calibration is universal, manifesting as a significant rise in overconfidence that undermines a model's reliability. Unlike the alignment tax, which concerns *what* a model can do, the calibration cost represents a fundamental loss in *knowing* what it knows. While Hu et al. (2025) propose post-hoc *Model Merging* to navigate this trade-off, our approach, EPICAR, mitigates this cost intrinsically during the learning process. By employing a dual-task objective that balances reasoning performance with reliability, we enable models to achieve Pareto-superiority without the need for extrinsic post-hoc interventions.

Calibration & Uncertainty Estimation Calibration approaches have evolved from logit scaling such as *Temperature Scaling* (Guo et al., 2017) to LLM-specific strategies like *Verbalized Confidence* (Kadavath et al., 2022; Lin et al., 2022), though verbalized outputs remains susceptible to prompt variations (Xia et al., 2025) and struggle with long-form reasoning (Yang et al., 2025b). To enhance reliability, inference-time ensembles like *Self-Consistency* (Wang et al., 2024) utilize response consistency but suffer from prohibitive computational costs. Recent literature, notably Yoon et al. (2025), identifies that reasoning models better express their confidence by leveraging *Slow Thinking* behaviors such as self-verification within an extended CoT. Furthermore, while models encode correctness in hidden states (Zhang et al., 2025), and auxiliary interventions like LitCab (Liu et al., 2024) or post-hoc calibration models like Thermometer (Shen et al., 2024) have been proposed for efficiency, our approach internalizes calibration directly into the generator's core objective. Crucially, as we demonstrate in Table 3, EPICAR provides a superior foundational representation that is highly complementary to such post-hoc methods. By balancing reasoning and self-evaluation within a unified SFT task, we achieve robust uncertainty estimation without the need for extended inference compute.

3 Preliminaries

We formally define the problem of calibrated reasoning, the mechanism for verbalized confidence estimation, and the theoretical limitations of iterative frameworks.

3.1 Problem Formulation and Verbalized Confidence

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a dataset of reasoning problems x_i and ground truth answers y_i. An LLM P_θ generates a reasoning path r and a predicted answer $\hat{y}$. Our objective is to maximize $P_\theta(y|x, r)$ while ensuring a reliable confidence score $c \in [0, 1]$.

Following Kapoor et al. (2024), we adopt *verbalized confidence estimation*. Instead of raw token probabilities, we prompt the model to assess its own correctness via a binary query (e.g., "Is the answer correct? `yes/no`"). The confidence c is the normalized probability of the affirmative token:

$$c = \frac{P_\theta(\texttt{yes} \mid x, r, \hat{y})}{P_\theta(\texttt{yes} \mid x, r, \hat{y}) + P_\theta(\texttt{no} \mid x, r, \hat{y})} \quad (1)$$

This semantic approach captures high-level certainty more effectively than sequence perplexity for reasoning tasks (Kuhn et al., 2023).

3.2 Evaluation Metrics: Calibration and Discrimination

To quantify calibration, we employ *Expected Calibration Error (ECE)* (Guo et al., 2017), which measures the weighted average discrepancy between accuracy and confidence. Let N be the total number of evaluation samples. We partition the predictions into M equally-spaced intervals (bins) based on their predicted confidence scores, where B_m denotes the set of indices of samples whose confidence falls into the m-th interval. ECE is defined as:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \quad (2)$$

where $\text{acc}(B_m)$ and $\text{conf}(B_m)$ represent the average accuracy and the average confidence within bin B_m, respectively. A perfectly calibrated model achieves an $\text{ECE} = 0$.

Furthermore, we report the *Brier Score* (Brier, 1950) as a comprehensive measure of the quality of predicted probabilities. For a set of binary outcomes (correct or incorrect), the Brier Score is defined as the mean squared error between the predicted confidence f_i and the actual outcome $o_i \in \{0, 1\}$:

$$\text{BS} = \frac{1}{N} \sum_{i=1}^{N} (f_i - o_i)^2 \quad (3)$$

The Brier Score is a proper scoring rule that can be decomposed into calibration, resolution, and uncertainty, providing a holistic view of the model's predictive performance. A lower Brier Score indicates a better-calibrated and more accurate model.

In addition to these metrics, we report the *Area Under the Receiver Operating Characteristic curve (AUROC)* to evaluate the model's discriminative ability. While ECE measures absolute alignment, AUROC assesses the model's capacity to assign higher confidence scores to correct answers than to incorrect ones, providing a measure of reliability that is invariant to logit scaling or temperature shifts.

3.3 The Epistemic Cost of Iterative Training

Standard iterative methods like STaR (Zelikman et al., 2022) utilize a "positive-only" feedback loop, fine-tuning exclusively on successful reasoning paths. We characterize the failure of this approach through the lens of the *epistemic learning problem*.

While *aleatoric uncertainty* stems from inherent stochasticity, *epistemic uncertainty* reflects a lack of knowledge regarding specific reasoning patterns (Hüllermeier and Waegeman, 2021). By training only on correct samples, the model suffers from *epistemic signal truncation*: it learns the distribution $P(r|x, y = 1)$ but never encounters the decision boundary between correct and incorrect paths. This leads to a form of *Model Collapse* (Shumailov et al., 2023), where the model converges toward low-variance point estimates and discards the distributional tails necessary for representing uncertainty. Consequently, the model incurs a severe *calibration cost* (Hu et al., 2025), manifesting as pathologically high confidence in logically flawed generations.

4 Methodology

We propose EPICAR, a framework that internalizes epistemic calibration into the iterative self-training loop without the need for auxiliary models or inference-time compute scaling.

4.1 Iterative Dual-Loop Framework

Our framework alternates between generating reasoning traces and optimizing a unified objective that balances problem-solving and self-evaluation. The detailed procedure is outlined in Algorithm 1, and training objective is presented in Section I.

Internalizing Evaluation Task Unlike STaR, which discards incorrect generations, we utilize them as negative signals for the *self-evaluation task*. For every generated path $(x, \hat{y})$, if the generation is correct, the instance is added to the *reasoning task* for accuracy reinforcement and labeled as "yes" for the self-evaluation task. Conversely, if the generation is incorrect, the instance is excluded from reasoning reinforcement but labeled as "no" for the self-evaluation task. To validate this unified objective, an ablation study confirming that the standard 1:1 unweighted SFT loss natively achieves near-optimal calibration without explicit hyperparameter tuning is provided in Appendix M.2.

This dual-task structure is motivated by recent findings from Zhang et al. (2025), which observed that while reasoning models encode correctness in their hidden states, standard non-reasoning models exhibit significantly degraded signals. They hypothesize that this latent capacity is acquired through exposure to reasoning patterns containing both correct and incorrect steps. By explicitly training on "no" labels for erroneous paths, our framework provides the necessary exposure to elicit and strengthen these internal signals in standard LLMs. This forces the model to exploit latent features of logical failure, effectively mitigating the alignment-induced *Model Collapse* and the resulting *calibration cost* described by Shumailov et al. (2023), and Hu et al. (2025).

Algorithm 1 Epistemically-Calibrated Reasoning

Require: Base Model $\mathcal{M}_0$, Dataset $\mathcal{D}$, Sampling size K

for iteration $t = 1 \dots T$ **do**
  $\mathcal{D}_{\text{reason}} \leftarrow \emptyset, \mathcal{D}_{\text{eval}} \leftarrow \emptyset$
  ▷ **Step 1: Generation Phase**
  for $x \in \mathcal{D}$ **do**
    $\hat{y}_1, \dots, \hat{y}_K \sim \text{Sample}(\mathcal{M}_{t-1}, x)$
    for $k = 1 \dots K$ **do**
      if IsCorrect($\hat{y}_k$) **then**
        $\mathcal{D}_{\text{reason}}$.add$(x, \hat{y}_k)$
        $\mathcal{D}_{\text{eval}}$.add$((x, \hat{y}_k), \texttt{yes})$
      else
        $\mathcal{D}_{\text{eval}}$.add$((x, \hat{y}_k), \texttt{no})$
      end if
    end for
  end for
  ▷ **Step 2: Mixing Phase**
  $\mathcal{D}_{\text{total}} \leftarrow \text{Shuffle}(\mathcal{D}_{\text{reason}} \cup \mathcal{D}_{\text{eval}})$
  ▷**Step 3: Dual-Objective Training**
  $\mathcal{M}_t \leftarrow \text{SFT}(\mathcal{M}_{t-1}, \mathcal{D}_{\text{total}})$
end for

4.2 Adaptive Injection Decoding (AID)

To eliminate noise caused by formatting failures, we adapt the method from Jin et al. (2025). When generating reasoning paths, we enforce format compliance (e.g., `\boxed{}`) by injecting rigid completion strings during decoding. This ensures that valid reasoning paths are not mislabeled as "no" due to parsing errors, which would otherwise provide a confusing training signal for the self-evaluation task. We provide a stateful implementation of AID to handle edge cases such as premature termination and unclosed formatting tags; see Appendix C for the detailed algorithmic logic, and Appendix J for the ablation study.

Model	Method	Perf.	Reliability & Calibration			
		Acc (↑)	AUROC (↑)	ECE (↓)	ECE (+TS) (↓)	Brier (↓)
Llama-3-1B	Base Model	3.30%	0.525	0.841	0.516	0.740
	+ Slow Thinking (ICL)	3.00%	0.507	0.827	0.517	**0.716**
	STaR	3.46%	0.491	0.838	0.515	0.737
	+ Model Merging	**3.54%**	0.518	0.838	0.515	0.737
	+ Slow Thinking (ICL)	3.12%	0.469	**0.826**	0.516	**0.716**
	Ours (EPICAR)	3.30%	0.573	0.871	0.532	0.800
	+ Model Merging	3.53%	0.555	0.848	**0.514**	0.754
	+ Slow Thinking (ICL)	3.14%	**0.576**	0.885	0.534	0.817
Llama-3-3B	Base Model	7.56%	0.555	0.376	0.315	0.216
	+ Slow Thinking (ICL)	5.84%	0.529	0.605	0.460	0.424
	STaR	7.38%	0.562	0.382	0.344	0.219
	+ Model Merging	7.60%	0.562	0.377	0.318	0.217
	+ Slow Thinking (ICL)	6.24%	0.548	0.609	0.456	0.432
	Ours (EPICAR)	**8.58%**	0.568	**0.108**	0.053	**0.097**
	+ Model Merging	7.86%	**0.593**	0.167	**0.050**	0.106
	+ Slow Thinking (ICL)	6.46%	0.571	0.440	0.436	0.257
Llama-3-8B	Base Model	13.30%	0.544	0.496	0.384	0.368
	+ Slow Thinking (ICL)	12.22%	0.387	0.448	0.384	0.316
	STaR	13.46%	0.570	0.494	0.381	0.365
	+ Model Merging	13.72%	0.555	0.492	0.381	0.365
	+ Slow Thinking (ICL)	13.06%	0.415	0.436	0.375	0.312
	Ours (EPICAR)	14.42%	**0.595**	0.415	0.362	**0.298**
	+ Model Merging	**15.02%**	0.571	0.443	**0.361**	0.328
	+ Slow Thinking (ICL)	14.38%	0.435	**0.390**	**0.361**	0.282

Table 1: **Llama-3 Family Results.** EPICAR generally outperforms STaR (Zelikman et al., 2022) in both accuracy and calibration, with significant gains observed in the 3B and 8B variants. While the 1B model improves discriminative power (AUROC) over the baseline, it exhibits trade-offs in reasoning accuracy due to limited capacity. We compare our results with weight-space interventions (Hu et al., 2025) and inference-time scaling (Yoon et al., 2025).

5 Experimental Setup

Datasets We evaluate our framework using the **MATH**[1] dataset (Hendrycks et al., 2021) for the main iterative training loop ($T = 3$). We test out-of-distribution (OOD) generalization on **GSM8K**[2] (Cobbe et al., 2021) and cross-domain robustness on **MBPP**[3] (Austin et al., 2021). For calibration tuning, we use a 500-instance split-validation for MATH, and use the official validation set for MBPP. Specifically, for the scaling analysis in Section 6.4, we evaluate on the **MATH-500**[4] subset to ensure comparability with prior literature. For MBPP, we adopt a 3-shot prompting setup and evaluate functional correctness via a sandboxed execution environment; see Appendix D for details on code extraction and the verification pipeline.

Models and Baselines We employ the **Llama-3** (Dubey et al., 2024) and **Qwen-3** (Yang et al., 2025a) families, focusing on 8B variants for MBPP and ablation studies. We compare our method against three primary baselines: (1) the **Base Model**, (2) **STaR** (Zelikman et al., 2022) for iterative self-improvement, and (3) **Slow Thinking (ICL)** (Yoon et al., 2025) for reliability. Additionally, we evaluate post-hoc **Model Merging** (Hu et al., 2025) by sweeping $\lambda \in \{0.0, 0.2, \ldots, 1.0\}$ to navigate the alignment-calibration frontier. Detailed algorithmic backgrounds and implementa-

[1] https://huggingface.co/datasets/EleutherAI/hendrycks_math
[2] https://huggingface.co/datasets/openai/gsm8k
[3] https://huggingface.co/datasets/google-research-datasets/mbpp
[4] https://huggingface.co/datasets/HuggingFaceH4/MATH-500

Model	Method	Perf.	Reliability & Calibration			
		Acc (↑)	AUROC (↑)	ECE (↓)	ECE (+TS) (↓)	Brier (↓)
Qwen-3-1.7B	Base Model	41.44%	0.408	0.101	0.074	0.255
	+ Slow Thinking (ICL)	41.34%	0.601	0.091	0.101	0.245
	STaR	38.16%	0.430	0.124	0.102	0.257
	+ Model Merging	42.02%	0.413	0.111	0.072	0.257
	+ Slow Thinking (ICL)	42.98%	0.617	**0.042**	0.049	**0.239**
	Ours (EPICAR)	42.34%	**0.637**	0.297	0.079	0.323
	+ Model Merging	**43.08%**	0.543	0.161	**0.018**	0.270
	+ Slow Thinking (ICL)	32.06%	0.631	0.204	0.054	0.253
Qwen-3-4B	Base Model	40.66%	0.676	0.093	0.093	0.232
	+ Slow Thinking (ICL)	27.98%	0.755	0.275	0.269	0.240
	STaR	43.20%	0.765	0.273	0.154	0.283
	+ Model Merging	42.58%	0.784	0.240	0.168	0.263
	+ Slow Thinking (ICL)	**55.78%**	0.796	0.150	0.131	0.232
	Ours (EPICAR)	43.50%	**0.835**	0.137	0.139	0.193
	+ Model Merging	43.54%	0.826	0.176	0.164	0.211
	+ Slow Thinking (ICL)	41.06%	0.820	**0.126**	**0.080**	**0.186**
Qwen-3-8B	Base Model	45.86%	0.727	0.196	0.121	0.259
	+ Slow Thinking (ICL)	55.16%	0.673	0.134	0.065	0.251
	STaR	49.52%	0.710	0.179	0.117	0.258
	+ Model Merging	48.08%	0.712	0.190	0.122	0.261
	+ Slow Thinking (ICL)	54.52%	0.659	0.190	**0.032**	0.271
	Ours (EPICAR)	49.76%	**0.797**	0.131	0.088	**0.206**
	+ Model Merging	49.72%	0.769	0.123	0.103	0.217
	+ Slow Thinking (ICL)	**55.56%**	0.780	**0.107**	0.148	0.220

Table 2: **Qwen-3 Family Results.** EPICAR demonstrates superior discriminative reliability (AUROC) across all scales and outperforms STaR (Zelikman et al., 2022), especially at 3B and 8B scale. Smaller models exhibit mixed results in calibration error and scaling efficiency compared to the baseline. We compare our results with weight-space interventions (Hu et al., 2025) and inference-time scaling (Yoon et al., 2025).

tion for these baselines are provided in Appendix E.

Inference-time Scaling Protocol To evaluate how internalized calibration scales with inference-time compute, we sample $K \in \{1, 10, 30\}$ reasoning paths. We compare two aggregation strategies: (1) **Self-Consistency (SC)** (Wang et al., 2023), which selects the plurality answer via simple majority voting, and (2) **Confidence-Informed Self-Consistency (CISC)** (Taubenfeld et al., 2025). Unlike SC, CISC performs a weighted majority vote by utilizing the model's self-assessed confidence $\mathcal{V}(r_i)$ as a fidelity signal. This allows the model to surpass the accuracy of high-sample-count SC with significantly fewer samples (e.g., $K = 10$ matching $K = 30$), thereby reducing inference compute. The formal definition of the CISC aggregation and normalization process is provided in Appendix F.

Implementation Performance is measured via Accuracy, AUROC, ECE, and Brier Score. To assess the model's intrinsic uncertainty beyond logit-level shifts, we also report ECE after applying *Temperature Scaling (TS)* (Guo et al., 2017). Training uses LoRA ($r = 16, \alpha = 32$) on 4x NVIDIA H100 GPUs. Following Kapoor et al. (2024), we update hidden features to support accurate verbalized confidence. Detailed hyperparameters, prompting templates, and the optimization procedure for TS are provided in Appendix B, and Appendix G.

6 Results and Analysis

In this section, we evaluate our method's performance across different model families and sizes. We report Accuracy for reasoning capability, and AUROC, ECE, and Brier Score for calibration quality. To ensure a rigorous comparison, we contrast our EPICAR against the standard STaR baseline, as well as state-of-the-art inference-time scaling (*Slow Thinking* (Yoon et al., 2025)) and weight-space intervention (*Model Merging* (Hu et al., 2025)) strategies. For *Model Merging*, we report the results obtained using the optimal λ that yields the best result with respect to accuracy and calibration, following Hu et al. (2025). See Appendix K for full results for all λ.

6.1 Main Findings: Accuracy and Reliability Synergy

Our results across both model families (Tables 1 and 2) demonstrate that internalizing the self-evaluation objective allows LLMs to navigate the reasoning-reliability frontier more effectively, particularly as model scale increases.

Mitigating Calibration Cost of Iterative SFT Standard iterative SFT (STaR) consistently incurs a calibration cost, improving accuracy while often degrading discriminative reliability. For instance, in Llama-3-1B, while STaR marginally boosts accuracy, it drops AUROC to 0.491. In contrast, our approach recovers this discriminative power (AUROC 0.573), albeit with a slight trade-off in reasoning accuracy at this scale. However, for larger models like Llama-3-3B and 8B, our method achieves strict Pareto-dominance, effectively preventing the overconfident model collapse typical of positive-only feedback loops (e.g., reducing ECE from 0.376 to 0.108 in Llama-3-3B). Notably, EPICAR achieves the lowest Brier Score in most cases (e.g., 0.097 for Llama-3-3B), indicating superior overall predictive quality.

Foundation for Inference-time Scaling A pivotal finding is the synergy between our internalized calibration and *Slow Thinking* (Yoon et al., 2025) in capable models. While Slow Thinking generally boosts performance, it can be unstable on uncalibrated models. Our calibrated framework stabilizes this behavior in the 8B variants, achieving a performance of 55.56% on Qwen-3-8B. We note, however, that this synergy is scale-dependent; in intermediate sizes (e.g., Qwen-3-4B), the added inference complexity does not yield performance gains over STaR, suggesting that a critical mass of reasoning capability is required to effectively leverage the self-evaluation signal.

Weight-space Interpolation & Merging We observe significant synergy with *Model Merging* (Hu et al., 2025), indicating that our reliability signals are complementary to weight-space interventions. While STaR yields marginal gains from merging, applying it to our checkpoints unlocks substantial headroom: the *Ours + Merging* variant achieves the highest Llama-3-8B accuracy of 15.02%. Precision calibration is also enhanced, with Qwen-3-1.7B reaching a near-perfect ECE (+TS) of 0.018, demonstrating the flexibility of our method for post-hoc optimization.

Synergy with Post-hoc Calibration EPICAR also serves as a robust foundational representation for post-hoc calibration. We benchmarked our approach against Temperature Scaling and Thermometer (Shen et al., 2024). As shown in Table 3, EPICAR natively achieves strong intrinsic discrimination without external MLPs (e.g., outperforming STaR+Thermometer's AUROC natively on Qwen-3-8B). Furthermore, applying Thermometer *on top of* an EPICAR-trained model yields the absolute best overall calibration quality across metrics.

Model	Method	Acc (%)	AUROC ↑	ECE ↓	Brier ↓
Qwen-3-8B	STaR	50.00	0.760	0.203	0.261
	EPICAR (Ours)	48.80	0.799	0.146	0.213
	STaR + TS	50.00	0.758	0.153	0.243
	EPICAR + TS	49.80	**0.824**	0.135	**0.196**
	STaR + Thermo.	50.00	0.770	0.062	0.197
	EPICAR + Thermo.	49.80	0.768	**0.058**	0.197
Llama-3-8B	STaR	16.40	0.553	0.459	0.351
	EPICAR (Ours)	16.20	0.594	0.389	0.288
	STaR + TS	16.40	0.553	0.349	0.258
	EPICAR + TS	17.40	0.582	0.332	0.253
	STaR + Thermo.	16.40	0.696	**0.046**	0.125
	EPICAR + Thermo.	17.40	**0.736**	0.060	**0.124**

Table 3: **Post-hoc Calibration Comparison (MATH-500).** Applying post-hoc methods *on top of* EPICAR yields the best overall performance, demonstrating it provides a superior foundational representation. (TS: Temperature Scaling, Thermo.: Thermometer)

6.2 Out-of-Distribution Generalization

To evaluate the generalization performance of our internalized calibration objective, we conduct zero-shot evaluations on GSM8K (Cobbe et al., 2021). This serves as a critical out-of-distribution (OOD) benchmark, as the models were trained exclusively on the MATH dataset.

Reliability of Verbalized Confidence Rankings As shown in Table 4, our method consistently enhances the model's discriminative reliability across both families. A key finding is that while standard iterative self-training (STaR) often leads to a degradation in AUROC (e.g., Llama-3-1B falling to 0.478), our approach significantly improves it,

Family	Method	Acc. (↑)	AUROC (↑)	ECE (↓)	Brier (↓)
Llama-3-1B	Base Model	3.49%	0.545	0.841	0.741
	STaR	3.49%	0.478	**0.840**	**0.740**
	Ours (EPICAR)	**3.64%**	**0.645**	0.922	0.887
Llama-3-3B	Base Model	14.94%	0.527	0.303	0.223
	STaR	17.13%	0.497	0.284	0.228
	Ours (EPICAR)	**21.46%**	**0.606**	**0.020**	**0.166**
Llama-3-8B	Base Model	31.39%	**0.565**	0.329	0.324
	STaR	32.15%	0.561	0.338	0.332
	Ours (EPICAR)	**37.45%**	**0.565**	**0.223**	**0.284**
Qwen-3-1.7B	Base Model	77.33%	0.490	**0.445**	**0.376**
	STaR	78.54%	0.493	0.541	0.463
	Ours (EPICAR)	**79.30%**	**0.660**	0.671	0.610
Qwen-3-4B	Base Model	85.06%	0.611	0.364	0.255
	STaR	86.05%	0.693	**0.099**	0.122
	Ours (EPICAR)	**88.25%**	**0.756**	0.132	**0.106**
Qwen-3-8B	Base Model	85.90%	0.654	**0.215**	0.162
	STaR	87.95%	0.678	**0.215**	**0.147**
	Ours (EPICAR)	**89.46%**	**0.722**	0.364	0.216

Table 4: **Zero-shot GSM8K Results.** EPICAR consistently improves reasoning accuracy while maintaining or enhancing AUROC.

reaching 0.645 and 0.606 for Llama-3-1B and 3B, respectively. This demonstrates that the verbalized confidence scores produced by our framework are better separated, assigning substantially higher probabilities of correctness to true reasoning paths than to erroneous ones, even in unseen domains.

Discriminative Power vs. Absolute Calibration Despite high AUROC, some variants exhibit elevated Raw ECE. We hypothesize that while our dual-objective training effectively teaches the model to *distinguish* between correct and incorrect paths, it does not explicitly supervise the absolute alignment of verbalized probability logits with empirical accuracy. Consequently, the model maintains a strong sense of *relative* certainty, but the absolute values remain somewhat detached from the true probability space. We identify the explicit optimization of these verbalized probability logits as a crucial direction for future work to bridge the gap between discriminative power and absolute calibration.

6.3 Generalization to Other Tasks

Finally, we examine whether our approach generalizes beyond mathematics to the domain of code generation. We apply our framework to the MBPP benchmark (Austin et al., 2021), focusing on the 8B model variants.

Table 5 summarizes the performance across three stages: Base Model, STaR, and Ours. While our method does not always achieve the absolute highest AUROC, it consistently outperforms the STaR baseline in discriminative reliability. Specifically, in Llama-3-8B, STaR incurs a significant calibration cost, dropping AUROC from 0.551 to 0.523, whereas our method mitigates this degradation, recovering it to 0.538.

Model	Method	Acc. (↑)	AUROC (↑)	ECE (+TS) (↓)	Brier (↓)
Llama-3-8B	Base Model	37.35%	**0.551**	0.398	0.391
	STaR	37.74%	0.523	0.390	0.387
	Ours (EPICAR)	**39.30%**	0.538	**0.113**	**0.246**
Qwen-3-8B	Base Model	45.14%	0.622	0.066	**0.286**
	STaR	45.14%	0.577	0.066	0.285
	Ours (EPICAR)	**45.91%**	**0.628**	**0.059**	0.285

Table 5: **Results on MBPP coding task.** Our method (EPICAR) consistently improves Pass Rate (Accuracy) and demonstrates competitive calibration performance across different model families.

Crucially, EPICAR achieves a significantly lower Brier Score (0.246 vs. 0.387 in Llama-3-8B) and the lowest ECE (+TS) across both families, reaching 0.113 for Llama-3-8B and 0.059 for Qwen-3-8B. This improvement in a proper scoring rule (Brier Score) confirms that internalizing self-evaluation signals effectively prevents the overconfident model collapse typical of positive-only self-training, even in cross-domain reasoning tasks like programming.

To rigorously verify that the learned epistemic signal is a generalized meta-cognitive capability rather than an artifact of model scale, we expanded our zero-shot cross-domain evaluation on MBPP across all six model families (1B to 8B). Crucially, no MBPP data was seen during training. As shown in Table 6, EPICAR robustly transfers to this unseen coding domain, achieving superior AUROC, ECE, and Brier scores in most cases compared to STaR. This confirms that internalized calibration generalizes effectively across scales and tasks.

6.4 Inference-time Scaling Performance

We further investigate how internalized calibration impacts the inference-time scaling laws of reasoning models. By ensembling K sampled reasoning paths, we analyze the synergy between internalized confidence and test-time compute.

Figure 2 illustrates the results on the MATH-500 benchmark. Our framework demonstrates superior scaling efficiency compared to the STaR baseline. Notably, when paired with CISC (Taubenfeld et al., 2025), our 8B model achieves a $3\times$ reduction in inference compute—matching or exceeding the $K = 30$ performance of STaR with only $K = 10$ samples. This suggests that EPICAR provides a more robust foundation for compute scaling,

Model	Method	Pass (%)	AUROC ↑	ECE ↓	Brier ↓
Llama-3-1B	STaR	**22.57**	**0.537**	0.774	0.774
	EPICAR	21.79	0.496	**0.727**	**0.699**
Llama-3-3B	STaR	**26.07**	0.513	0.674	0.670
	EPICAR	25.68	**0.590**	**0.253**	**0.254**
Llama-3-8B	STaR	**38.13**	0.453	0.604	0.600
	EPICAR	36.96	**0.472**	**0.523**	**0.522**
Qwen-3-1.7B	STaR	**40.86**	0.541	**0.236**	**0.296**
	EPICAR	38.91	**0.654**	0.288	0.315
Qwen-3-4B	STaR	34.63	0.464	0.262	0.304
	EPICAR	**35.02**	**0.470**	**0.118**	**0.238**
Qwen-3-8B	STaR	**43.19**	0.572	0.336	0.354
	EPICAR	40.47	**0.589**	**0.100**	**0.244**

Table 6: **Zero-shot Cross-Domain Transfer (MATH → MBPP).** All models were trained exclusively on MATH. EPICAR robustly transfers the epistemic calibration signal to an unseen domain across all model scales.

Method	K	Ext. Verifier	Acc (%)	AUROC ↑	ECE ↓	Brier ↓
STaR (Verbal)	1	✗	50.00	0.760	0.203	0.261
EPICAR	1	✗	48.80	0.799	0.146	0.213
V-STaR (Best-of-K)	10	✓	51.20	**0.893**	0.148	0.174
EPICAR (CISC)	10	✗	**58.00**	0.808	**0.120**	**0.157**

Table 7: **Comparison with External Verifier (Qwen-3-8B on MATH-500).** Under the same sampling budget ($K = 10$), EPICAR structurally eliminates the multi-model overhead while outperforming V-STaR.

as internalized reliability signals effectively suppress erroneous consensus paths that often cause performance saturation in uncalibrated models. A comprehensive analysis of all scales and reliability trajectories is provided in Appendix A.

We additionally benchmark our internalized confidence against V-STaR (Hosseini et al., 2024), which relies on a separately trained external verifier. As shown in Table 7, under the same sampling budget of $K = 10$ candidate generations, our CISC approach surpasses V-STaR in accuracy, ECE, and Brier score. Crucially, because V-STaR must score each candidate using a separate external model, it incurs approximately 21× the inference compute of a single EPICAR forward pass (see Appendix M.3 for the detailed compute breakdown). This highlights the immense practical efficiency of internalizing self-evaluation directly into the generator.

7 Conclusion

In this work, we addressed the *calibration cost* inherent in iterative self-training, where models gain reasoning accuracy at the expense of reliability and uncertainty representation. We introduced EPICAR, a framework that reframes open-ended reasoning training as an epistemic learning problem by internalizing a dual-objective of accuracy reinforcement and explicit meta-cognitive self-evaluation. Our experiments across Llama-3 and Qwen-3 families demonstrate that EPICAR consistently achieves *Pareto-superiority*, simultaneously enhancing both performance and calibration while generalizing robustly across model scales to unseen domains like code generation (MBPP). Furthermore, we demonstrated that EPICAR serves as a superior foundational representation for post-hoc calibration techniques. Crucially, our single-model approach structurally eliminates the massive multi-model overhead inherent to external verifiers. By leveraging self-assessed confidence through weighted ensembling (CISC), our models match high-sample-count performance with significantly fewer reasoning paths, yielding an effective 3× reduction in the overall inference compute budget while avoiding the ∼21× multi-model overhead of external verifiers like V-STaR. Ultimately, this work advocates for a paradigm shift in model alignment, where uncertainty calibration is treated as an integral training objective for trustworthy and compute-efficient reasoning agents.

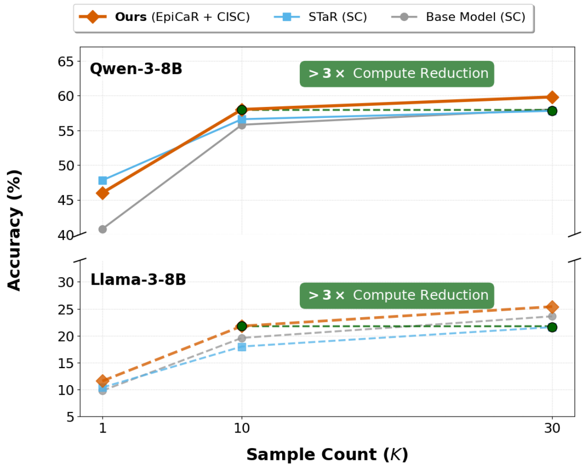

Figure 2: **Inference-time Scaling on MATH-500.** Visual representation of ensemble accuracy across sample sizes K. Our framework paired with CISC achieves superior scaling efficiency, outperforming STaR and establishing a new frontier for compute-optimal reasoning.

Limitations

Despite the performance gains and improved calibration, our work has several limitations.

Domain Scope and Verification. While we demonstrated that EPICAR generalizes across mathematics and code generation, our evaluation remains centered on domains with objective, automated ground-truth verification. It remains to be seen whether EPICAR can be effectively applied to more subjective or open-ended reasoning domains, such as open-ended QA or creative writing, where defining a clear signal for knowing what it does not know is inherently more difficult.

Model Scale and Capacity Constraints. Our experiments were conducted on models up to 8B parameters. As identified in Appendix A, our method is sensitive to the baseline reasoning capacity. In extremely low-accuracy regimes, such as Llama-3-1B, the scarcity of successful reasoning paths during the iterative loop can prevent the model from learning the nuanced discriminative features necessary for accurate self-evaluation. This suggests a critical mass threshold for effective internalized calibration.

Generalization Gap in Absolute Calibration. We observe a notable discrepancy between discriminative power and absolute numerical calibration in out-of-distribution (OOD) scenarios. While EPICAR consistently maintains superior AUROC (ranking) across benchmarks like GSM8K, the absolute probabilities it verbalizes (measured by Raw ECE) exhibit higher volatility compared to in-distribution results on MATH. This highlights a fundamental gap between learning to rank outputs and achieving absolute logit alignment in unseen domains.

Comparison with RL and Verifier-based Methods. While recent RL-based approaches like RLCR (Damani et al., 2025) achieve calibration through specialized reward structures, they often suffer from training instability and high hyperparameter sensitivity. In contrast, EPICAR operates within a stable iterative SFT framework. Furthermore, unlike V-STaR (Hosseini et al., 2024), which requires a separate verifier model, our single-model approach avoids approximately $21\times$ the inference compute overhead of V-STaR at $K = 10$ (Table 7).

Potential for Reinforcement Learning Integration. A promising direction for future research is the integration of our internalized calibration objective into reinforcement learning (RL) frameworks. The self-evaluation signals generated in our dual-loop framework could serve as an intrinsic reward mechanism or a dense feedback signal for RL algorithms such as PPO (Schulman et al., 2017), DPO (Rafailov et al., 2023), or GRPO (DeepSeek-AI et al., 2025).

Sensitivity of the Verbalization Paradigm. EPICAR adopts the verbalized confidence paradigm, which remains somewhat susceptible to prompt sensitivity (Xia et al., 2025). However, our extended prompt analysis (Appendix M.1) demonstrates that explicit training via EPICAR significantly internalizes the self-evaluation signal, reducing cross-template variance by up to 44% compared to zero-shot baselines like STaR. Nevertheless, further research is required to ensure absolute robustness across all linguistic contexts.

Ethical Considerations

Our research aims to improve LLM reliability, a critical step toward safe AI deployment. However, we acknowledge several ethical implications.

Misuse and Over-reliance: By enhancing a model's ability to express confidence, there is a risk that users may over-rely on outputs when the model reports high certainty. While EPICAR significantly reduces overconfidence, no technique is infallible, and "confident hallucinations" may still occur, potentially leading to errors in high-stakes decision-making.

Bias Amplification: As our framework utilizes iterative self-training on model-generated data, there is a potential to amplify biases present in the pretrained state or the training distribution. We recommend applying our method alongside rigorous bias-detection and mitigation protocols.

Environmental Impact and Cost Trade-offs: Our framework incurs higher training compute than single-pass SFT due to its iterative nature and dual-task data processing. However, this one-time training cost unlocks a $3\times$ reduction in inference-time compute during deployment and bypasses the massive multi-model overhead of external verifiers, making it a highly favorable trade-off for real-world applications where inference efficiency dominates.

Use of AI Assistants. In accordance with the ACL Policy on AI Assistance, we acknowledge the use of Gemini 3 Pro[5] to assist with code debugging and writing polishing. All experimental designs, data analyses, and scientific claims presented in this work were verified by the authors.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00345809, Research on AI Robustness Against Distribution Shift in Real-World Scenarios; and No. RS-2023-00222663, Center for Optimizing Hyperscale AI Models and Platforms), and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2026-25507282, Development of AI learning technologies based on brain cognitive science and biological learning principles).

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

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

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

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

Tingchen Fu, Deng Cai, Lemao Liu, Shuming Shi, and Rui Yan. 2024. Disperse-then-merge: Pushing the limits of instruction tuning via alignment tax reduction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2967–2985. Association for Computational Linguistics.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, and 1 others. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

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.

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. *NeurIPS*.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-STaR: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling (COLM)*.

Tiancheng Hu, Benjamin Minixhofer, and Nigel Collier. 2025. Navigating the alignment-calibration trade-off: A pareto-superior frontier via model merging. *arXiv preprint arXiv:2510.17426*.

Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506.

Hyunbin Jin, Je Won Yeom, Seunghyun Bae, and Taesup Kim. 2025. "well, keep thinking": Enhancing llm reasoning with adaptive injection decoding. *arXiv preprint arXiv:2503.10167*.

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

Sanyam Kapoor, Manley Roberts, Katherine Collins, Arka Pal, Nate Gruver, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. 2024. Large language models must be taught to know what they don't know. In *Advances in Neural Information Processing Systems*, volume 36.

Yoonjeon Kim, Doohyuk Jang, and Eunho Yang. 2025. Meta-awareness enhances reasoning models: Self-alignment reinforcement learning. *arXiv preprint arXiv:2510.03259*.

[5]https://deepmind.google/technologies/gemini/

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pages 611–626. ACM.

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

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the alignment tax of rlhf. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 580–606. Association for Computational Linguistics.

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

Keming Lu, Bowen Yu, Fei Huang, Yang Fan, Runji Lin, and Chang Zhou. 2024. Online merging optimizers for boosting rewards and mitigating tax in alignment. *arXiv preprint arXiv:2405.17931*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Maohao Shen, Subhro Das, Kristjan Greenewald, Prasanna Sattigeri, Gregory Wornell, and Soumya Ghosh. 2024. Thermometer: Towards universal calibration for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 44607–44631. PMLR.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20090–20111, Vienna, Austria. Association for Computational Linguistics.

Ante Wang, Linfeng Song, Ye Tian, Baolin Peng, Lifeng Jin, Haitao Mi, Jinsong Su, and Dong Yu. 2024. Self-consistency boosts calibration for math reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6023–6029, Miami, Florida, USA. Association for Computational Linguistics.

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

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

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. Association for Computational Linguistics.

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

Ruihan Yang, Caiqi Zhang, Zhisong Zhang, Xinting Huang, Sen Yang, Nigel Collier, Dong Yu, and Deqing Yang. 2025b. LoGU: Long-form generation with uncertainty expressions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18947–18968. Association for Computational Linguistics.

Dongkeun Yoon, Seungone Kim, Sohee Yang, Sunkyoung Kim, Soyeon Kim, Yongil Kim, Eunbi Choi, Yireun Kim, and Minjoon Seo. 2025. Reasoning models better express their confidence. In *The 39th Conference on Neural Information Processing Systems (NeurIPS)*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488.

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. In *The 2nd Conference on Language Modeling (COLM)*.

A Full Results and Analysis of Inference-time Ensembling

A.1 Comprehensive Ensemble Accuracy of All Scales

We perform an extensive evaluation of the scaling properties of *EpiCaR* across all model variants using the **MATH-500** test set. By varying the number of sampled reasoning paths $K \in \{1, 10, 30\}$, we map the compute-performance frontier as summarized in Table 8.

Internalized Calibration as a Scaling Foundation. Our experimental results consistently demonstrate that our method (**Ours**) establishes a superior and more stable baseline for inference-time scaling compared to both the pre-trained Base Model and the STaR baseline. At the 8B parameter scale, the synergy between internalized calibration and weighted aggregation becomes most apparent. Pairing our model with **CISC** (Taubenfeld et al., 2025) achieves state-of-the-art results: 25.40% for Llama-3 and 59.80% for Qwen-3 at $K = 30$. This trajectory suggests that when a model is trained to generate its own reliability signals, the performance gains from additional inference compute do not just accumulate—they accelerate, as the model's self-evaluation serves as a high-fidelity filter for the sampling process.

Synergy, Error Suppression, and Saturation. The synergy between EpiCaR and the CISC aggregation protocol is a pivotal finding. While standard Self-Consistency (SC) (Wang et al., 2023) relies on a frequentist plurality vote, it is vulnerable to "common errors" where incorrect reasoning paths form a deceptive majority. As observed in the Qwen-3 8B results, the STaR baseline exhibits significant performance saturation, reaching a plateau at 57.80% for $K = 30$. In contrast, our calibrated model continues to scale effectively, effectively breaking the saturation ceiling. This confirms that a model that "knows what it knows" can utilize verbalized confidence to suppress high-frequency erroneous paths. By assigning low weights to incorrect but common reasoning traces, EpiCaR ensures that the ensemble consensus is driven by logical fidelity rather than mere repetition.

The Critical Mass Threshold in Reasoning capacity. A nuanced and critical finding is that the advantages of internalized calibration are not uniform across all model scales, identifying what we term a *"Critical Mass Threshold."* In the extremely low-accuracy regime, such as with **Llama-3 1B** (baseline accuracy $\sim$1.80%), the gap between our method and the STaR baseline diminishes. We hypothesize that when the model's initial reasoning capability is too sparse, the training process lacks a sufficient density of correct reasoning paths to learn the subtle discriminative features required for accurate self-evaluation. This suggests that while EpiCaR is a powerful scaling tool, its benefits are most salient once a model reaches a fundamental level of reasoning competence. Future work should investigate whether specialized data augmentation can lower this threshold for smaller model architectures.

A.2 Full Reliability Analysis of Scaling: SC vs. CISC

To assess the qualitative aspects of uncertainty estimation within an ensemble, we evaluate reliability metrics across $K \in \{5, 10, 30\}$. For the **CISC** variants, we derive a scalar ensemble confidence score $\mathcal{C}_{CISC}$ for each unique answer a:

$$\mathcal{C}_{CISC}(a) = \frac{\sum_{i:\text{ans}(r_i)=a} \mathcal{V}(r_i)}{\sum_{j=1}^{K} \mathcal{V}(r_j)} \tag{4}$$

where $\mathcal{V}(r_i) \in [0, 1]$ represents the model's internalized verbalized confidence for path r_i. The comprehensive results for AUROC and ECE are detailed in Table 9.

Reliability Collapse in STaR Baseline. A striking phenomenon observed in our scaling analysis is the **Reliability Collapse** of the STaR baseline as K increases. For instance, in Llama-3 8B, STaR's AUROC degrades significantly from 0.7895 ($K = 5$) to 0.7387 ($K = 30$). This degradation suggests a fundamental flaw in positive-only iterative training: by exclusively reinforcing successful paths, the model homogenizes its error patterns, leading to incorrect reasoning paths that form a deceptively high-confidence consensus. As more samples are added, these "confident hallucinations" dominate the ensemble, undermining its discriminative power. In sharp contrast, our model (**Ours**) sustains or even improves its AUROC as compute scales. By internalizing logical failure modes during training, EpiCaR maintains a clear margin between correct and incorrect reasoning paths, which is essential for the trustworthy deployment of LLMs in high-stakes reasoning tasks.

Metric Divergence: The Interplay of ECE and AUROC. We observe a characteristic divergence in reliability trends as K scales: while ECE generally improves, AUROC tends to decay. We hypothesize that this is driven by two counteracting statistical forces:

- **Statistical Smoothing:** ECE benefits from larger sample sizes as the averaged ensemble confidence naturally converges toward the true population accuracy, reducing absolute calibration error.
- **Consensus-Driven Noise:** AUROC suffers as the probability of incorrect paths forming a high-confidence plurality increases with K, narrowing the discriminative gap between true and false positives.

Our framework effectively mitigates the decay rate of AUROC compared to all baselines. By ensuring that the model's confidence is tied to the *intrinsic logic* of the path rather than the *extrinsic frequency* of the answer, EpiCaR achieves a more robust reliability-compute trade-off.

Family	Size	Method	Ensemble	$K=1$	$K=10$	$K=30$
Llama-3	1B	Base Model	SC CISC	1.80%	2.00% 2.40%	3.80% **4.40%**
		STaR	SC CISC	**3.20%**	**3.20%** 3.00%	**4.40%** 3.80%
		Ours	SC CISC	1.60%	2.80% **3.20%**	3.40% 3.40%
	3B	Base Model	SC CISC	5.40%	8.20% 8.40%	10.60% 10.20%
		STaR	SC CISC	4.80%	8.40% 9.40%	**15.00%** 14.60%
		Ours	SC CISC	**7.40%**	**13.60%** **13.60%**	13.40% 13.20%
	8B	Base Model	SC CISC	9.80%	19.60% 20.20%	23.60% 23.40%
		STaR	SC CISC	10.40%	18.00% 17.80%	21.60% 21.80%
		Ours	SC CISC	**11.60%**	21.00% **21.80%**	24.80% **25.40%**
Qwen-3	1.7B	Base Model	SC CISC	29.60%	33.40% 35.80%	39.60% 39.80%
		STaR	SC CISC	37.20%	46.60% 46.80%	48.80% 47.40%
		Ours	SC CISC	**40.40%**	**51.40%** 51.20%	**53.00%** 52.40%
	4B	Base Model	SC CISC	36.60%	51.40% 51.60%	54.60% 55.40%
		STaR	SC CISC	41.80%	53.00% 53.60%	54.60% 55.00%
		Ours	SC CISC	**44.40%**	54.20% **55.00%**	53.80% **57.20%**
	8B	Base Model	SC CISC	40.80%	55.80% 55.80%	58.00% 58.40%
		STaR	SC CISC	**47.80%**	56.60% 56.80%	57.80% 58.00%
		Ours	SC CISC	46.00%	57.40% **58.00%**	59.20% **59.80%**

Table 8: **Comprehensive Ensemble Accuracy on MATH-500 (%).** Comparison of unweighted Self-Consistency (SC (Wang et al., 2023)) and Confidence-Informed Self-Consistency (CISC (Taubenfeld et al., 2025)) across multiple model families and sizes. $K = 1$ values are merged as the methods are identical without multiple samples. Bolding indicates the highest accuracy within each size group for a specific K.

Family	Size	Method	Ensemble	$K = 5$ AUROC ↑	$K = 5$ ECE ↓	$K = 10$ AUROC ↑	$K = 10$ ECE ↓	$K = 30$ AUROC ↑	$K = 30$ ECE ↓
Llama-3	1B	Base	SC	0.4784	0.2380	0.4760	0.1644	0.6490	0.0995
			CISC	0.4824	0.2416	0.5442	0.1633	0.6213	0.0941
		STaR	SC	**0.6870**	**0.2172**	0.7390	**0.1490**	0.6643	**0.0915**
			CISC	0.6848	0.2247	**0.7540**	0.1539	**0.7308**	0.0981
		Ours	SC	0.5636	0.2516	0.6398	0.1742	0.6478	0.1273
			CISC	0.5564	0.2551	0.7078	0.1731	0.6588	0.1284
	3B	Base	SC	0.7002	0.2252	0.7225	0.1450	0.6760	0.0843
			CISC	0.7758	0.2488	0.7390	0.1544	0.6843	0.0875
		STaR	SC	0.7413	0.2356	0.6972	0.1470	0.6169	**0.0460**
			CISC	0.6967	0.2540	0.6942	0.1458	0.6343	0.0558
		Ours	SC	**0.8170**	**0.2080**	**0.7556**	**0.1212**	0.7561	0.0951
			CISC	0.7475	0.2618	0.7303	0.1468	**0.7972**	0.0913
	8B	Base	SC	0.7470	0.1952	0.7283	0.0978	0.7184	0.0472
			CISC	0.7499	0.2108	0.7034	0.1083	0.7059	0.0613
		STaR	SC	0.7895	0.1812	0.7741	0.1058	0.7387	0.0529
			CISC	0.7892	0.2057	0.7810	0.1164	0.7300	0.0452
		Ours	SC	0.7824	**0.1628**	**0.7988**	**0.0684**	**0.7722**	0.0397
			CISC	**0.8511**	0.2175	0.7795	0.0782	0.7636	**0.0383**
Qwen-3	1.7B	Base	SC	0.7315	0.2156	0.7182	0.1508	0.7065	0.1181
			CISC	0.7414	0.2303	0.7208	0.1624	0.6967	0.1057
		STaR	SC	0.7638	0.1960	0.7657	0.1344	0.7521	0.1174
			CISC	0.7771	0.2177	0.7572	0.1307	0.7688	0.1157
		Ours	SC	0.7853	**0.1672**	0.7756	**0.0986**	0.7750	**0.0900**
			CISC	**0.7950**	0.1860	**0.7774**	0.1186	**0.7812**	0.0930
	4B	Base	SC	0.7869	**0.1304**	0.7830	**0.0968**	0.7667	**0.0573**
			CISC	0.7834	0.1360	**0.7954**	0.1015	0.7721	0.0478
		STaR	SC	0.8170	0.2028	0.7864	0.1390	0.7713	0.1229
			CISC	**0.8266**	0.2063	0.7916	0.1431	0.7762	0.1295
		Ours	SC	0.7737	0.1776	0.7544	0.1242	**0.7859**	0.1108
			CISC	0.7764	0.1915	0.7776	0.1503	0.7781	0.1139
	8B	Base	SC	0.7626	**0.0884**	0.7213	**0.0786**	0.7457	**0.0767**
			CISC	0.7791	0.1205	0.7359	0.0945	0.7516	0.0707
		STaR	SC	0.7972	0.1632	0.7735	0.1260	0.7854	0.1009
			CISC	**0.8128**	0.1598	0.7919	0.1296	0.7909	0.1018
		Ours	SC	0.7767	0.1640	0.7926	0.1098	0.7914	0.0959
			CISC	0.7932	0.1630	**0.8077**	0.1201	**0.8010**	0.0998

Table 9: **Comprehensive Reliability Scaling across All Models (AUROC and ECE).** Comparison of frequency-based (SC) and weighted (CISC) confidence estimates across training stages. Bold entries indicate the best performance (highest AUROC or lowest ECE) within each size group for a given K.

B Full Experimental Configurations

B.1 Iterative Training Loop Details

For both the STaR baseline and EpiCaR, we conduct three complete iterations ($T = 3$) of the self-improvement loop. In each iteration, the model generates K candidate reasoning paths for each problem in the MATH training set. For STaR, only paths reaching the ground-truth answer are retained for fine-tuning. For our method, we perform the mixing strategy (Section 4.1) using both correct and incorrect attempts to refine the self-evaluation objective. We find that performance and calibration metrics typically stabilize after the third iteration, after which further training often leads to marginal returns.

B.2 Data Processing and Validation Protocol

For the iterative SFT loop, we utilize the training split of the **MATH** dataset (12,500 problems). To evaluate the robustness of internalized calibration in out-of-distribution (OOD) scenarios, we test on **GSM8K** without further fine-tuning. A critical component of our evaluation is the split-validation protocol. Calibration metrics can be overly optimistic if the scaling parameters are optimized on the same set used for reporting. Thus, we randomly sample $N = 500$ instances from the test set as a "calibration-validation" set. We find the optimal temperature T that minimizes ECE on this set and apply it to the remaining test instances ($N_{test} - 500$) for final reporting.

B.3 Data Licensing and Usage

We utilize standard public benchmarks consistent with their intended research purposes. The **MATH** (Hendrycks et al., 2021) and **GSM8K** (Cobbe et al., 2021) datasets are released under the MIT License, while the **MBPP** (Austin et al., 2021) dataset is distributed under the CC-BY 4.0 License.

B.4 Baseline Implementation Details

- **STaR:** We implement the standard self-taught reasoner by filtering self-generated reasoning paths that reach the ground-truth answer. The model is fine-tuned for 3 iterations, with the checkpoint from T_i serving as the generator for T_{i+1}.
- **Slow Thinking (ICL):** We adopt the few-shot prompts from Yoon et al. (2025), which encourage models to "think slow" by verbalizing internal verification steps. This baseline represents the upper bound of inference-time calibration without training.
- **Model Merging:** We use weight-space interpolation between the base model (θ_{base}) and SFT checkpoints (θ_{SFT}). The coefficient λ represents the weight of the SFT model.

B.5 Training and Hyperparameters

All models were fine-tuned using the Hugging Face TRL library. We employ Parameter-Efficient Fine-Tuning (PEFT) via LoRA to ensure scalability.

Hyperparameter	Value
Optimizer	AdamW
Learning Rate	1×10^{-5}
LR Scheduler	Cosine
Batch Size	32
Max Sequence Length	2048
LoRA Rank (r)	16
LoRA Alpha (α)	32
LoRA Target Modules	All linear layers
Precision	bfloat16

Table 10: Detailed Hyperparameters for Dual-Objective Iterative SFT.

B.6 Inference Implementation

To evaluate the effectiveness of our proposed method in terms of both reasoning accuracy and confidence calibration, we implemented a high-throughput inference pipeline using the `vLLM` library (Kwon et al., 2023). All models were loaded in `bfloat16` precision with eager execution mode enabled to ensure stability.

B.7 Hyperparameters and Decoding Strategies

We utilized distinct decoding strategies tailored to the nature of each benchmark:

- **Greedy Decoding for Standard Benchmarks (MATH, GSM8K):** For the primary evaluation of reasoning accuracy and calibration, we employed greedy decoding (`temperature=0.0`). The generation max length was set to 1024 tokens to allow sufficient reasoning steps (CoT). To accurately extract answers and measure verbalized confidence, we utilized a prefix injection technique (e.g., "*So, the answer is* \boxed{") with a forced stop token at the closing brace. This injection step acts as a constraint that improves the format compliance and accuracy of the final answer.

- **Sampling for Robustness (MATH-500):** To assess the model's performance under self-consistency (Wang et al., 2023), we employed sampling with `temperature=0.7` and generated $N = 30$ candidate paths per problem. The `repetition_penalty` was dynamically adjusted to maintain diversity without degradation.

- **Code Generation (MBPP):** For the code generation tasks, we adopted a few-shot prompting strategy consistent with prior work (Austin et al., 2021). The inference was performed with `temperature=0.0` for pass@1 evaluation, employing specific stop sequences (e.g., "Problem:", "Tests:") to prevent hallucination beyond the solution function.

C Implementation Details of Adaptive Injection Decoding (AID)

Our *Adaptive Injection Processor* is implemented as a stateful `LogitsProcessor` that monitors the decoding process at each step. Unlike static constraints, it dynamically transitions through states based on the model's output. The operational logic is categorized into three primary mechanisms:

1. Triggering and Injection State The processor maintains an `is_injecting` flag and an `injection_step` counter. The injection is triggered under two conditions: (1) when the model attempts to emit a termination token (EOS) before a `\boxed{}` tag is generated, or (2) when the sequence reaches a soft length limit (e.g., $L_{max} - 150$ tokens). During the injection state, the model's logits are overridden to force the generation of the transition phrase: *"\nSo, the answer is \boxed{"*.

2. Stateful Format Enforcement To ensure the injected format is correctly finalized, the processor tracks the sequence for the existence of the boxed token and its corresponding closing brace (`}`).

- **Premature EOS Prevention:** If the model attempts to terminate while a box is open (`has_injected` is True but no closing brace is found), the processor overrides the EOS token with a forced closing brace (`}`).

- **Content Length Control:** To prevent infinite or malformed generation within the answer box, the processor enforces a maximum content length ($C_{max} = 40$). If exceeded, it mandates a closing brace and subsequent termination.

3. Zombie and Sampling Defense The processor utilizes a `finished_mask` to prevent "zombie" generations where a model might continue after a logical EOS. Additionally, in cases where no box is detected, the processor suppresses all stop tokens until the injection logic is triggered, ensuring that every sample produced for the training pipeline is parsable and evaluatable.

AID as a De-noising Filter. Our ablation study (Section J) shows a performance collapse without Adaptive Injection Decoding (AID). We clarify that AID does not introduce reasoning bias; rather, it acts as a sanitization process to remove label noise. Without AID, a mathematically sound derivation might be labeled as "no" due to minor formatting errors (e.g., missing a closing `\boxed{}`). Such mislabeling provides a catastrophic signal, teaching the model that "valid logic is incorrect." AID ensures that the "no" label strictly reflects logical failures, thereby protecting the integrity of the self-evaluation supervision.

D Details of MBPP Evaluation and Execution

For the code generation task, we implement a robust evaluation pipeline to ensure that the model's reasoning capabilities are not underestimated due to minor formatting inconsistencies.

1. Robust Code Extraction Language models often produce conversational fillers or incomplete code snippets. As shown in our implementation, we employ a multi-stage extraction heuristic:

- **Block Extraction:** We first attempt to parse the output for Markdown-style Python blocks (` ```python ... ``` `).

- **Keyword-based Recovery:** If no block is found, the processor searches for functional keywords such as `import`, `def`, or `return` to identify the start of the logic.

- **Signature Injection:** A common failure mode in LLMs is omitting the function header. To mitigate this, our pipeline checks if the canonical function signature (e.g., `def remove_Occ(...)`) is present; if missing, it automatically prepends the signature to the model's output to ensure the script is functionally complete and ready for execution.

2. Sandboxed Execution and Verification Correctness is verified by executing the extracted code against a set of hidden unit tests provided by the MBPP dataset.

- **Environment:** Execution is performed in a separate subprocess using Python's `multiprocessing` to prevent side effects and handle potential infinite loops via a 3-second timeout.

- **Dependency Injection:** To support diverse coding problems, the environment is preloaded with essential libraries, including `math`, `heapq`, `collections`, and `itertools`.

- **Strict Pass Criterion:** A task is labeled as *Correct* ($y = 1$) if and only if the code executes without errors and passes every `assert` statement in the test list.

3. Prompting Strategy We use a 3-shot prompt consisting of diverse Python tasks (e.g., sum of squares, average of cubes) to steer the model toward a consistent "Reasoning-then-Solution" format. This mirrors the iterative self-training objective used in our math reasoning tasks, allowing for a direct comparison of calibration performance across different domains.

E Technical Overview of Baselines and Related Concepts

In this section, we provide detailed technical backgrounds for the primary baselines and concepts utilized in our study: STaR, Slow Thinking (ICL), and Model Merging.

E.1 Self-Taught Reasoner (STaR)

STaR (Zelikman et al., 2022) is an iterative bootstrapping framework designed to improve a model's reasoning capabilities using only a large dataset of questions and answers without intermediate rationales. Its core loop consists of:

1. **Rationale Generation:** The model is prompted (via few-shot) to generate a Chain-of-Thought (CoT) and a final answer.

2. **Filtering:** Only rationales that lead to the correct ground-truth answer are retained.

3. **Rationalization:** For questions where the model fails, it is provided with the correct answer as a "hint" to generate a backward-reasoned rationale.

4. **Fine-tuning:** The model is fine-tuned on the combined set of successful and rationalized paths.

In our experiments, STaR serves as the primary iterative baseline, representing the "accuracy-first" reinforcement approach that often incurs a high calibration cost.

E.2 Slow Thinking via In-Context Learning (ICL)

Based on the findings of Yoon et al. (2025), **Slow Thinking** refers to the cognitive behaviors exhibited by advanced reasoning models (e.g., DeepSeek-R1, o1), such as exploring alternative approaches, self-verification, and backtracking within an extended CoT.

- **Mechanism:** Unlike standard CoT, slow thinking is non-linear. The model dynamically adjusts its internal confidence as it rejects erroneous paths or confirms logical steps.

- **ICL Implementation:** For non-reasoning models (baselines), we elicit these behaviors via In-Context Learning (ICL) by providing few-shot exemplars that demonstrate these "slow" cognitive patterns. This serves as a strong baseline for reliability without parameter updates.

Prompt Templates. For the *Slow Thinking (ICL)* baseline, we adopt the prompting strategy from Yoon et al. (2025)[6]. This template encourages the model to engage in meta-cognitive reasoning within <think> tags before providing a final answer. The full template used for our experiments is provided in Listing 1.

```
First, generate your thought process
    aloud in first-person in between the
     <think> tags. Then, outside of the
    <think> tag, generate your final
    answer. You must stop thinking
    outside of the <think> tag, and
    commit to providing an answer in the
     format of **Answer**: $ANSWER. Here
     are some examples.

[Example 1]
Question: Who was the next British Prime
     Minister after Arthur Balfour?
Output:
<think>
```

[6] https://github.com/MattYoon/reasoning-models-confidence

```
Okay, so I need to figure out who was
    the next British Prime Minister
    after Arthur Balfour... [Detailed
    reasoning process] ...So, the
    sequence is Balfour, then Campbell-
    Bannerman, then Asquith.
</think>
**Answer**: Henry Campbell-Bannerman
[Example 1 End]

[Example 2]
Question: Men Against the Sea and
    Pitcairn's Island were two sequels
    to what famous novel?
Output:
<think>
... [Detailed reasoning process] ...So
    the famous novel they're sequels to
    is "Mutiny on the Bounty."
</think>
**Answer**: Mutiny on the Bounty.
[Example 2 End]

[Example 3]
Question: River Phoenix died during the
    making of which movie?
Output:
<think>
... [Detailed reasoning process] ...So,
    to sum up my thoughts: Paramount was
     originally called the "Hartford
    Film Company."
</think>
**Answer**: Hartford Film Company.
[Example 3 End]

This is the end of the examples. The
    following is the question you need
    to answer following the style of the
     examples.
```

Listing 1: Full Few-Shot Prompt Template for Slow Thinking (ICL)

E.3 Alignment-Calibration Trade-off and Model Merging

Hu et al. (2025) formalize the **Alignment Tax** not just as a loss of task performance, but as a **Calibration Cost**—where models become universally overconfident after instruction tuning.

- **Model Merging:** To mitigate this, they propose interpolating the weights of a Pre-Trained (PT) base model (θ_{PT}) and its Instruction-Tuned (IT) counterpart (θ_{IT}):

$$\theta_{merged} = (1 - \lambda)\theta_{PT} + \lambda\theta_{IT} \tag{5}$$

  where λ is the merging coefficient.

- **Pareto-Superior Frontier:** This simple linear interpolation often reveals a "sweet spot" where the merged model achieves higher accuracy than both parents while substantially recovering the calibration lost during alignment. We use this post-hoc method to benchmark the limits of weight-space optimization against our intrinsic training-time approach, *EpiCaR*.

F Formal Definition of Confidence-Informed Self-Consistency (CISC)

To maximize the utility of our model's internalized calibration during inference, we adopt *Confidence-Informed Self-Consistency (CISC)* (Taubenfeld et al., 2025) as our primary scaling strategy. Given a question q and a set of m sampled reasoning paths and answers $\{(r_1, a_1), \ldots, (r_m, a_m)\}$, the CISC aggregation process consists of three stages:

1. Confidence Extraction. For each reasoning path r_i, we extract the verbalized confidence score $c_i = \mathcal{V}(r_i)$. In our framework, these scores are generated via the verbalized confidence described in Section 3.1.

2. Softmax Normalization. To balance the relative importance of answer frequency (majority) and individual path reliability, the raw confidence scores are normalized using a temperature-scaled softmax:

$$\tilde{c}_i = \frac{\exp(c_i/T)}{\sum_{j=1}^{m} \exp(c_j/T)} \tag{6}$$

where T is a tunable hyper-parameter. As $T \to \infty$, CISC collapses to vanilla SC (uniform weights). As $T \to 0$, the mechanism prioritizes the single path with the highest confidence, effectively becoming a "greedy" confidence selection.

3. Weighted Aggregation. The final answer $\hat{a}_{CISC}$ is determined by summing the normalized confidence weights for each unique answer candidate:

$$\hat{a}_{CISC} = \arg\max_{a} \sum_{i=1}^{m} \mathbb{1}[a_i = a] \cdot \tilde{c}_i \tag{7}$$

G Calibration and Temperature Scaling Details

In addition to standard calibration metrics, we employ *Temperature Scaling (TS)* (Guo et al., 2017) to evaluate the quality of the model's confidence scores independently of its overconfidence bias.

Temperature Scaling Procedure. We optimize a single scalar parameter $T > 0$ for each model family and dataset. The temperature T is determined by minimizing the Negative Log-Likelihood (NLL) on a held-out validation set ($N_{val} = 500$):

$$\min_{T} -\sum_{i=1}^{N_{val}} \log\left(\sigma(\mathbf{z}_i/T)\right) \quad (8)$$

where $\mathbf{z}_i$ represents the model's output logits for the correct answer. We denote the ECE calculated using these calibrated probabilities as **ECE-TS**. This allows us to distinguish between models that are inherently uncalibrated and those that simply require a linear adjustment to their confidence variance.

H Extended Related Work

Comparison with RL and Verifier-based Methods. While recent RL-based approaches like RLCR (Damani et al., 2025) achieve calibration, they often suffer from training instability and high hyperparameter sensitivity. In contrast, EPICAR operates within a stable iterative SFT framework. Furthermore, unlike V-STaR (Hosseini et al., 2024), which requires a separate verifier model, our approach unifies generation and verification into a single model. This integration is the key driver behind our reported 3× reduction in inference compute, as it eliminates the overhead of managing multiple model forward passes.

Prompting vs. Fine-tuning for Calibration. Recent studies have debated whether LLMs can express uncertainty through zero-shot prompting alone. While some suggest that scale improves self-evaluation (Kadavath et al., 2022), Kapoor et al. (2024) argue that prompting on its own is insufficient for reliable calibration due to high sensitivity to linguistic variances. They demonstrate that fine-tuning on even a small graded dataset significantly outperforms black-box prompting by leveraging the model's internal features. EPICAR builds on this insight by integrating this calibration-tuning into an iterative reasoning loop, ensuring that the model's confidence is grounded in its evolving reasoning capacity rather than static prompt templates.

I Detailed Training Objective

We prioritize a minimalist design to ensure robustness. Unlike multi-task frameworks that require tuned auxiliary coefficients, EPICAR utilizes the standard Causal Language Modeling (CLM) loss:

$$\mathcal{L}_{\text{EpiCaR}} = \mathcal{L}_{CLM} = -\sum_{t} \log P_\theta(w_t \mid w_{<t}) \quad (9)$$

The model treats reasoning tokens and the self-evaluation token (e.g., `yes/no`) as a single continuous sequence.

Intrinsic Curriculum via Natural Mixing. Rather than using a fixed mixing weight, we adopt a natural mixing strategy. By pooling all reasoning and self-evaluation samples as described in Algorithm 1, the training distribution dynamically evolves. In early iterations where model accuracy is low, the dataset is naturally dominated by negative self-evaluation samples (“`no`”). This forces the model to prioritize self-objectivity and uncertainty representation (epistemic grounding). As reasoning capability improves, the proportion of positive reasoning samples naturally increases. This serves as an intrinsic curriculum that stabilizes the learning trajectory without manual intervention.

J Ablation Study: Impact of Adaptive Injection Decoding

To validate the structural importance of our proposed *Adaptive Injection Decoding (AID)*, we perform an ablation analysis using Llama-3-8B and Qwen-3-8B on the MATH dataset. We compare our full method (**Ours**) against a variant trained without injection decoding (**w/o AID**), where formatting errors are treated as standard incorrect samples.

Model	Variant	Acc. (↑)	AUROC (↑)	ECE (↓)
Llama-3-8B	**Ours (Full)**	**14.47%**	**0.594**	**0.415**
	w/o AID	2.56%	0.507	0.580
Qwen-3-8B	**Ours (Full)**	**49.62%**	**0.800**	**0.133**
	w/o AID	47.83%	0.784	0.152

Table 11: **Ablation on Adaptive Injection Decoding (AID).** Removing AID leads to significant performance degradation, highlighting its role in preventing formatting noise from corrupting the evaluation signal.

As shown in Table 11, the removal of AID results in a significant degradation in both reasoning accuracy and calibration metrics. For Llama-3-8B, the accuracy drops catastrophically to 2.56%. This collapse occurs because, without AID, valid reasoning paths that suffer from minor formatting issues (e.g., missing box delimiters) are mislabeled as "incorrect" (“`no`”). This introduces severe label noise, confusing the model's self-evaluation mechanism and destabilizing the reinforcement loop. AID effectively filters this noise, ensuring that the negative signal purely reflects *logical* failures rather than syntactic ones.

K Full Results for Model Merging

In this section, we provide the comprehensive results of our model merging experiments across the Llama-3 and Qwen-3 model families. Table 13 and Table 14 details the performance and reliability metrics for varying interpolation coefficients λ, ranging from the pure base model ($\lambda = 0.0$) to the fully fine-tuned models ($\lambda = 1.0$).

Effect of Interpolation Coefficient (λ) The results demonstrate that model merging—interpolating weights between the base model and the fine-tuned model—serves as an effective regularizer. For most model scales, we observe that an intermediate value of λ (e.g., $0.6 \leq \lambda \leq 0.8$) often provides a superior balance between task-specific accuracy and probabilistic calibration. Interestingly, in larger models such as Qwen-3-8B and Llama-3.1-8B, our method (EPICAR) maintains high discriminative performance (AUROC) even at $\lambda = 1.0$, whereas STaR often exhibits signs of overconfidence or degraded calibration as the fine-tuning progresses.

Comparison: STaR vs. EPICAR Across all tested architectures, EPICAR consistently outperforms the STaR baseline in both discriminative reliability (AUROC) and calibration error (ECE and Brier Score). Notably:

- **Discriminative Power:** EPICAR achieves significantly higher AUROC scores, particularly at the 4B and 8B scales. This indicates that the epistemic uncertainty captured during our training process allows the model to better distinguish between correct and incorrect reasoning paths.
- **Calibration Stability:** While STaR often shows fluctuating ECE values as λ increases, EPICAR tends to show a more stable or improving trend in Brier scores. This suggests that the calibration objective in EPICAR successfully aligns the model's confidence with its actual predictive correctness.

Scaling Trends We observe a clear scaling trend where larger models (8B) benefit more from the weight-space intervention. While smaller models like Llama-3.2-1B show mixed results in calibration error, they still exhibit improved AUROC when merged with our epistemically-calibrated weights. This confirms that EPICAR provides a more robust initialization for model merging compared to standard self-training objectives, effectively preserving the base model's general knowledge while injecting calibrated reasoning capabilities.

L Extended Reliability Analysis

In this section, we provide the full set of reliability diagrams across all evaluated benchmarks: MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and MBPP (Austin et al., 2021). These diagrams visualize the relationship between the model's verbalized confidence and its empirical accuracy, providing qualitative evidence for the calibration improvements discussed in the main text.

We partition the model's predictions into $M = 10$ equally-spaced bins based on verbalized confidence. A perfectly calibrated model would align with the dashed diagonal line ($y = x$).

As shown in Figures 3 through 6, our method (EPICAR) consistently demonstrates better alignment and higher discriminative power (AUROC) compared to the STaR baseline, which frequently exhibits overconfident clusters in the lower accuracy bins—a clear symptom of the *calibration cost* of standard iterative SFT.

M Extended Ablation Studies and Sensitivity Analysis

To further validate the robustness and structural choices of EPICAR, we provide the following extended analyses.

M.1 Prompt Sensitivity Analysis

To evaluate the robustness of verbalized confidence against prompt variations, we tested four diverse self-evaluation templates (Original, Formal, Casual, Detailed) on MATH-500. As shown in Table 16, EPICAR not only improves the mean AUROC but also consistently reduces the cross-template variance (standard deviation). This indicates that explicit training genuinely internalizes self-evaluation, making it significantly more robust to phrasing variations than zero-shot metacognitive baselines.

M.2 Loss Weighting Ablation

To validate the structural choice of using standard unweighted CLM loss, we ablated the explicit loss weighting between the reasoning (α) and calibration ($1 - \alpha$) tasks on Qwen-3-8B. As shown in Table 17, reasoning accuracy remained stable, while ECE achieved a broad, near-optimal plateau around $\alpha = 0.5$. This confirms that the standard unified SFT objective implicitly balances the tasks well without the need for manual hyperparameter tuning.

M.3 Inference Compute Breakdown of External Verifiers

In Section 6.4, we note that V-STaR incurs approximately $21\times$ the inference compute of a single EPICAR forward pass. This empirical multiplier is directly derived from the structural overhead of utilizing a separate external verifier model.

As detailed in Table 12, evaluating a single problem with EPICAR ($K = 1$) requires exactly 1 generation pass, as the model outputs its verbalized confidence concurrently with the reasoning solution. In contrast, evaluating V-STaR with $K = 10$ candidate solutions requires 10 full autoregressive generation passes from the generator model, followed by 10 independent evaluation passes from the external verifier model to score each generated candidate. Assuming the generator and verifier models share the same architecture scale, this necessitates at least 20 full model forward passes. Factoring in the computational overhead of processing the generated CoT context twice (once during generation, once during verification) and multi-model orchestration, the total inference compute scales to $\sim 21\times$ that of a single EPICAR pass. Even when EPICAR is scaled to $K = 10$ using CISC, it only requires 10 passes, exactly halving the compute of V-STaR while matching or exceeding its calibration performance.

Method	K	Gen. Passes	Verifier Passes	Est. Compute
EPICAR (Single)	1	1	0	$1\times$
STaR (Majority Vote)	30	30	0	$30\times$
EPICAR (CISC)	10	10	0	$10\times$
V-STaR (Best-of-K)	10	10	10	$\approx 21\times$

Table 12: **Inference Compute Breakdown.** V-STaR requires passing generated candidates through a separate verifier model, effectively doubling the forward passes per candidate and incurring significant multi-model overhead compared to EPICAR's internalized single-pass evaluation.

Model	Method	λ	Acc. (↑)	AUROC (↑)	ECE (↓)	Brier (↓)
Llama-3-1B	Base Model	0.0	3.30%	0.525	0.841	0.740
	STaR + Merging	0.2	3.34%	0.530	0.841	0.740
		0.4	3.18%	0.529	0.842	0.741
		0.6	3.54%	0.518	**0.838**	**0.737**
		0.8	3.50%	0.506	**0.838**	**0.737**
		1.0	3.46%	0.491	**0.838**	**0.737**
	Ours (EPICAR) + Merging	0.2	3.20%	0.531	0.845	0.745
		0.4	3.53%	0.555	0.848	0.754
		0.6	**3.62%**	0.551	0.862	0.779
		0.8	3.44%	**0.595**	0.880	0.809
		1.0	3.30%	0.573	0.871	0.800
Llama-3-3B	Base Model	0.0	7.56%	0.555	0.376	0.216
	STaR + Merging	0.2	7.60%	0.562	0.377	0.217
		0.4	7.46%	0.564	0.378	0.217
		0.6	7.44%	0.562	0.378	0.217
		0.8	7.38%	0.542	0.380	0.219
		1.0	7.38%	0.562	0.382	0.219
	Ours (EPICAR) + Merging	0.2	7.70%	0.568	0.353	0.200
		0.4	7.82%	0.559	0.301	0.168
		0.6	7.80%	0.552	0.237	0.134
		0.8	7.86%	**0.593**	0.167	0.106
		1.0	**8.58%**	0.568	**0.108**	**0.097**
Llama-3-8B	Base Model	0.0	13.30%	0.544	0.496	0.368
	STaR + Merging	0.2	13.26%	0.531	0.496	0.368
		0.4	13.54%	0.546	0.495	0.367
		0.6	13.74%	0.556	0.494	0.367
		0.8	13.72%	0.555	0.492	0.365
		1.0	13.46%	0.570	0.494	0.365
	Ours (EPICAR) + Merging	0.2	13.90%	0.552	0.486	0.361
		0.4	14.12%	0.553	0.470	0.348
		0.6	**15.02%**	0.571	0.443	0.328
		0.8	14.04%	0.592	0.434	0.312
		1.0	14.42%	**0.595**	**0.415**	**0.298**

Table 13: **Full Results for Llama-3 Family with Model Merging.** Accuracy and calibration metrics across all λ values for Llama-3 (1B, 3B, and 8B).

Model	Method	λ	Acc. (↑)	AUROC (↑)	ECE (↓)	Brier (↓)
Qwen-3-1.7B	Base Model	0.0	41.44%	0.408	0.101	**0.255**
	STaR + Merging	0.2	41.26%	0.407	0.120	0.256
		0.4	42.02%	0.413	0.111	0.257
		0.6	41.64%	0.409	0.121	0.258
		0.8	41.24%	0.421	0.123	0.261
		1.0	38.16%	0.430	0.124	0.257
	Ours (EpiCaR) + Merging	0.2	42.46%	0.438	0.101	0.257
		0.4	42.42%	0.497	**0.099**	0.257
		0.6	**43.08%**	0.543	0.161	0.270
		0.8	42.72%	0.600	0.214	0.284
		1.0	42.34%	**0.637**	0.297	0.323
Qwen-3-4B	Base Model	0.0	40.66%	0.676	**0.093**	0.232
	STaR + Merging	0.2	41.78%	0.697	0.102	0.233
		0.4	41.60%	0.746	0.143	0.235
		0.6	42.32%	0.760	0.191	0.248
		0.8	42.58%	0.784	0.240	0.263
		1.0	43.20%	0.765	0.273	0.283
	Ours (EpiCaR) + Merging	0.2	42.20%	0.704	0.097	0.230
		0.4	42.72%	0.782	0.138	0.220
		0.6	43.40%	0.804	0.170	0.222
		0.8	**43.54%**	0.826	0.176	0.211
		1.0	43.50%	**0.835**	0.137	**0.193**
Qwen-3-8B	Base Model	0.0	45.86%	0.727	0.196	0.259
	STaR + Merging	0.2	45.84%	0.714	0.200	0.264
		0.4	47.14%	0.721	0.193	0.261
		0.6	47.44%	0.727	0.197	0.262
		0.8	48.08%	0.712	0.190	0.261
		1.0	49.52%	0.710	0.179	0.258
	Ours (EpiCaR) + Merging	0.2	46.48%	0.719	0.175	0.254
		0.4	47.10%	0.729	0.145	0.243
		0.6	48.54%	0.746	**0.112**	0.229
		0.8	49.72%	0.769	0.123	0.217
		1.0	**49.76%**	**0.797**	0.131	**0.206**

Table 14: **Full Results for Qwen-3 Family with Model Merging.** Accuracy and calibration metrics across all λ values for Qwen-3 (1.7B, 4B, and 8B).

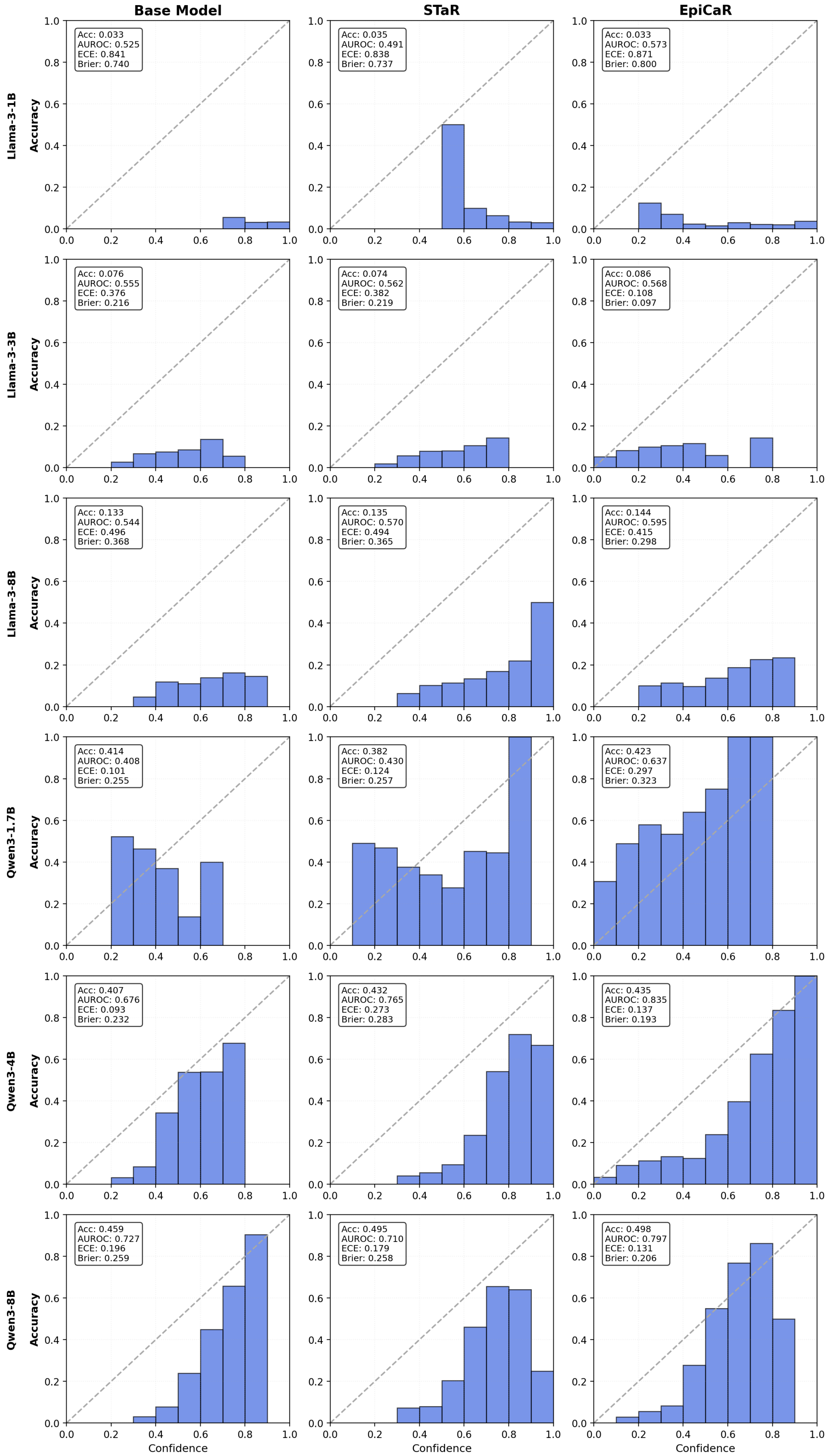

Figure 3: **Reliability Diagram: MATH (Standard).** Comparison of calibration performance between the base model, STaR, and EPICAR on the MATH dataset.

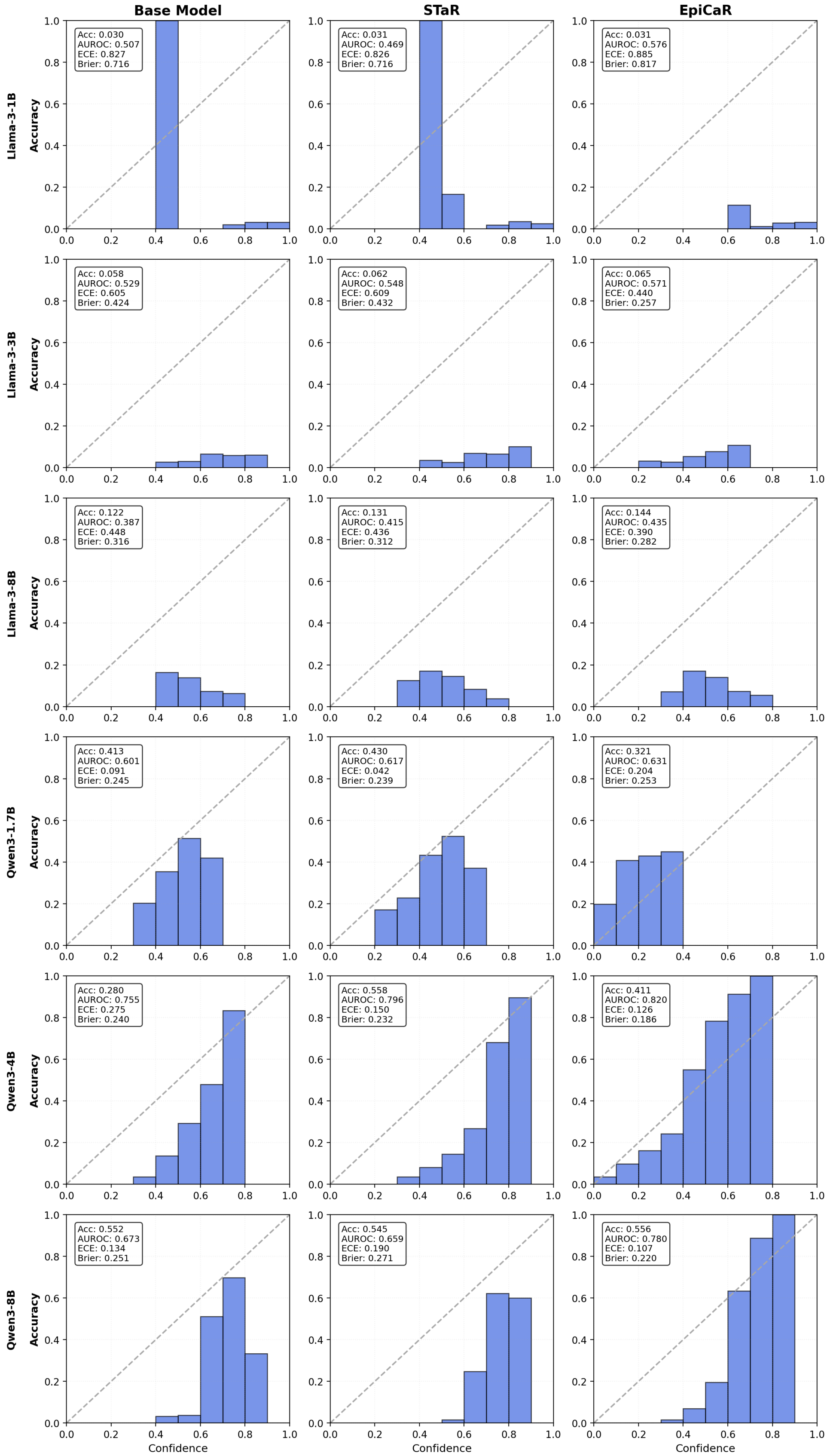

Figure 4: **Reliability Diagram: MATH (Slow Thinking).** Visualizing how internalized calibration interacts with inference-time "slow thinking" behaviors.

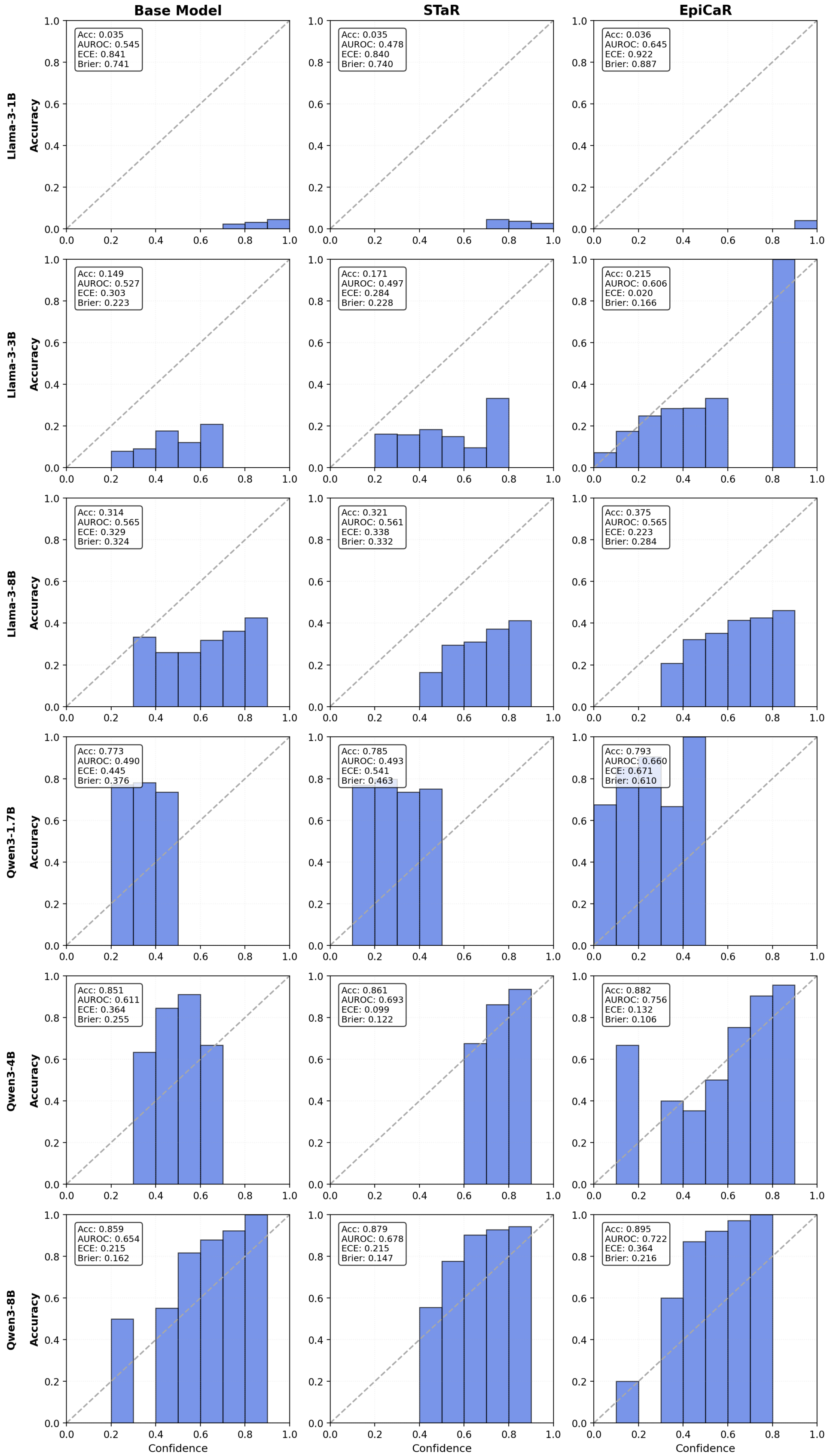

Figure 5: **Reliability Diagram: GSM8K (Zero-Shot).** Evaluation of epistemic uncertainty calibration in an out-of-distribution mathematical reasoning context.

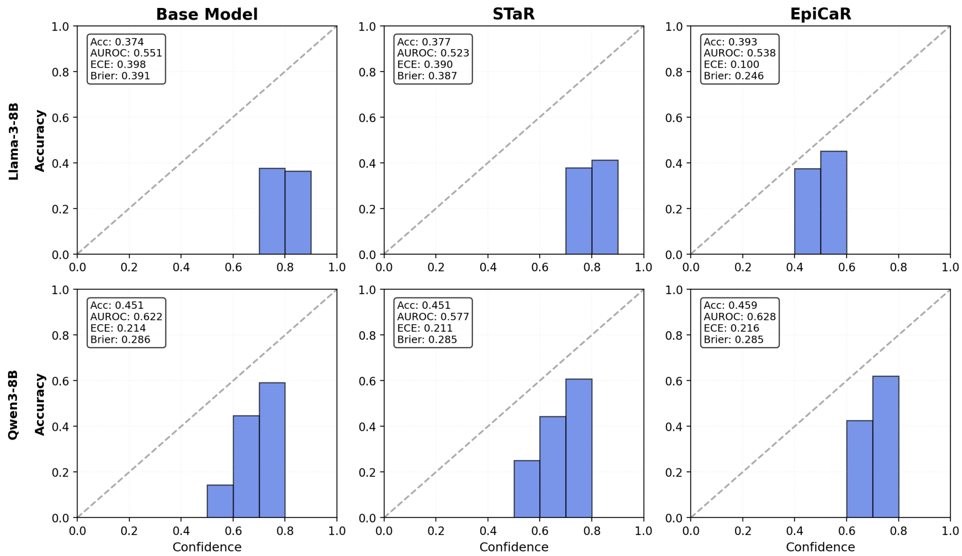

Figure 6: **Reliability Diagram: MBPP (Code Generation).** Cross-domain robustness of EPICAR in programming tasks.

Template	Self-Evaluation Prompt
Original	Is the answer correct? Choose ONLY one letter. A) Yes B) No. Your choice:
Formal	Based on careful mathematical verification, is this solution correct? Choose: A) Yes B) No. Answer:
Casual	Right or wrong? A) Right B) Wrong. Pick one:
Detailed	After checking each reasoning step and the final computation, is the answer correct? A) Yes, it is correct B) No, it is incorrect. Your choice:

Table 15: **Diverse Self-Evaluation Templates.**

Model	Method	AUROC (mean $\pm$ std)	ECE (mean $\pm$ std)
Qwen-3-8B	STaR	0.721 ± 0.045	0.267 ± 0.064
	EPICAR	$\mathbf{0.797 \pm 0.040}$	$\mathbf{0.203 \pm 0.055}$
Llama-3-8B	STaR	0.570 ± 0.057	0.645 ± 0.130
	EPICAR	$\mathbf{0.571 \pm 0.032}$	$\mathbf{0.594} \pm 0.155$

Table 16: **Prompt Sensitivity Analysis.** EPICAR effectively reduces cross-template variance.

Weight (α)	Acc (%)	AUROC $\uparrow$	ECE $\downarrow$
1.0 (Reasoning only)	44.20	0.6853	0.2658
0.9	43.20	0.7007	0.1213
0.7	**45.60**	0.7323	**0.0918**
0.5 (Ours, standard)	43.60	0.7616	0.0920
0.3	44.40	**0.7823**	0.1162

Table 17: **Loss Weighting Ablation.** The unweighted SFT objective ($\alpha = 0.5$) natively achieves near-optimal calibration.