

Confidence-Calibrated Small-Large Language Model Collaboration for Cost-Efficient Reasoning

Chuang Zhang¹, Zizhen Zhu², Yihao Wei¹, Bing Tian¹,
Junyi Liu¹, Henan Wang¹, Xavier Wang¹, Yaxiao Liu¹

¹Amazon Web Services ²Tsinghua University

{zhangchg, weiyihao, tiabingn, liujunyi, henanwan, wangxy, liuyaxia}@amazon.com
zzz23@mails.tsinghua.edu.cn

Abstract

Large language models (LLMs) demonstrate superior reasoning capabilities compared to small language models (SLMs), but incur substantially higher costs. We propose **Collaborative REAsoner (COREA)**, a system that cascades an SLM with an LLM to achieve a balance between accuracy and cost in complex reasoning tasks. COREA first attempts to answer questions using the SLM, which outputs both an answer and a verbalized confidence score. Questions with confidence below a pre-defined threshold are deferred to the LLM for more accurate resolution. We introduce a reinforcement learning-based training algorithm that aligns the SLM’s confidence through an additional confidence calibration reward. Extensive experiments demonstrate that our method jointly improves the SLM’s reasoning ability and confidence calibration across diverse datasets and model backbones. Compared to using the LLM alone, COREA reduces cost by 21.5% and 16.8% on out-of-domain math and non-math datasets, respectively, with only an absolute pass@1 drop within 2%.

1 Introduction

Reasoning ability has emerged as a central performance metric for large language models (LLMs), enabling them to solve complex problems in mathematics, science, and coding, etc. A key strategy for unlocking this capability is the generation of explicit chain-of-thought (CoT) reasoning steps (Wei et al., 2023), often combined with reinforcement learning (RL) or supervised fine-tuning (SFT). While highly effective, this approach introduces significantly higher inference cost due to the verbose reasoning traces, posing substantial barriers to real-world integration at scale (Wang et al., 2025).

To address the high inference costs, recent research has explored several approaches. One direction involves training small language models

(SLMs) that are more cost-efficient through knowledge distillation (Yu et al., 2024; Chenglin et al., 2024). However, SLMs typically exhibit weaker reasoning capabilities than LLMs (Li et al., 2025b). Another line of work focuses on optimizing reasoning efficiency within a single model through techniques such as adaptive computation (Snell et al., 2024), latent reasoning compression (Hao et al., 2024), or dynamic reasoning paradigms (Sun et al., 2024). While these methods can reduce computational overhead, they require modifying the model architecture or training procedure, and their effectiveness is still limited by the underlying model’s capacity. Routing-based approaches have also been proposed to direct queries to appropriately-sized models (Ong et al., 2024; Aytes et al., 2025), yet they typically rely on external classifiers or heuristics that may not accurately capture the model’s actual reasoning confidence. These limitations reveal a persistent trade-off: using SLMs alone sacrifices accuracy and robustness on challenging problems, while relying exclusively on LLMs incurs prohibitively high costs.

This contrast motivates a practical question: **can we combine the cost-efficiency of SLMs with the accuracy and robustness of LLMs?** Ideally, an SLM should take the first pass—solving problems it can handle confidently—and defer harder problems to a more powerful LLM. This collaborative approach promises to deliver high overall accuracy while keeping operational costs significantly lower than exclusive LLM usage. However, implementing such a collaborative system is non-trivial. The key technical challenge lies in enabling the SLM to recognize its own limitations and make informed decisions about whether to answer or defer. Specifically, SLMs currently *do not know what they know, nor do they know what they don’t know* (Yin et al., 2023; Prato et al., 2024; Kapoor et al., 2024).

In this work, we propose **COREA (Collaborative REAsoner)**, which empow-

ers a self-aware SLM to dynamically coordinate with an LLM during inference. Self-awareness is achieved by calibrating the SLM’s verbalized confidence using RL training with a customized reward function. In this system, when the SLM is confident and capable of providing a correct answer, it directly responds to the query; when uncertain, i.e., its self-assessed confidence falls below a threshold, it hands the task over to the LLM for further processing.

To summarize, our main contributions are:

1. We propose COREA, an SLM-LLM collaborative system that balances accuracy and cost by enabling the SLM to defer challenging problems to the LLM when confidence is low.
2. We develop an RL training method that combines a verifiable reward with a confidence calibration reward, and demonstrate empirically that it simultaneously improves reasoning ability and confidence calibration.
3. We conduct extensive experiments validating both the effectiveness of our RL training method and the performance of the collaborative system.

2 Related Work

2.1 Efficient Reasoning

LLMs such as GPT-5-thinking (OpenAI, 2025), Claude 4 (Anthropic, 2025), and DeepSeek-R1 (DeepSeek-AI et al., 2025) achieve strong reasoning performance by producing explicit reasoning traces (Li et al., 2025c), but verbose outputs incur high token costs and latency (Wang et al., 2025). Research on efficient reasoning follows three main strategies. **Single-model optimization** reduces overhead through adaptive computation (Snell et al., 2024), latent reasoning compression (Hao et al., 2024), or dynamic reasoning (Sun et al., 2024), but requires architectural modifications and remains limited by the model’s capacity. **Enhancing SLMs** via distillation (Yu et al., 2024; Chenglin et al., 2024; DeepSeek-AI et al., 2025), SFT (Muennighoff et al., 2025), or RL (Li et al., 2025a) enables cost-efficient deployment, yet distilled SLMs often struggle on complex problems (Li et al., 2025b) and exhibit overconfidence. **SLM-LLM collaboration** routes queries between models: query-level routing (Ong et al., 2024; Aytes et al., 2025) uses external classifiers that may misestimate difficulty, while fine-grained methods (Akhauri et al., 2025; Kim et al., 2025; Lee et al., 2024, 2025) and advanced cas-

ades (Gupta et al., 2024; Yue et al., 2023) defer challenging steps or employ uncertainty-based voting. However, these collaborative approaches typically require additional sampling overhead or external modules, in contrast, we directly calibrate the SLM’s internal confidence for reliable routing decisions.

2.2 Uncertainty and Confidence Calibration

Accurate self-assessment is critical for SLM-LLM collaboration, yet SLMs often lack reliable confidence calibration. Early studies analyzed prompting strategies and benchmarks for expressing doubt (Yin et al., 2023; Prato et al., 2024; Kapoor et al., 2024), revealing that RLHF often produces overconfident models (Leng et al., 2025). Research on improving calibration follows three main approaches. **Eliciting verbalized confidence** through prompt engineering (Yang et al., 2025; Lin et al., 2022) enables models to express uncertainty explicitly, however, calibration quality varies significantly across tasks and templates. **Introspection-based methods** leverage hidden states or auxiliary classifiers to detect when the model may be incorrect (Azaria and Mitchell, 2023), still, the performance is generally hard to extend to out-of-distribution tasks. **Training for calibration** incorporates confidence calibration into the learning objective: like including a Brier score (Damani et al., 2025; Xu et al., 2024), Logarithmic score (Stangel et al., 2025) in RL training. We follow the similar training approach, however, we propose several different calibration rewards and make an extensive comparison on these reward designs.

3 Methodology

In this section, we present the design of our COREA framework. We begin by describing the overall system architecture and then detail RL with confidence calibration (RLCC).

3.1 Overall Architecture: The SLM-LLM Collaborative Framework

As illustrated in Figure 1a, COREA employs a cascaded SLM-LLM architecture where queries are first processed by the SLM. The SLM is prompted to generate reasoning steps, an answer, and a verbalized confidence score indicating the likelihood that its answer is correct. If the confidence exceeds a predefined threshold T , the SLM’s answer becomes the final output. Otherwise, the original

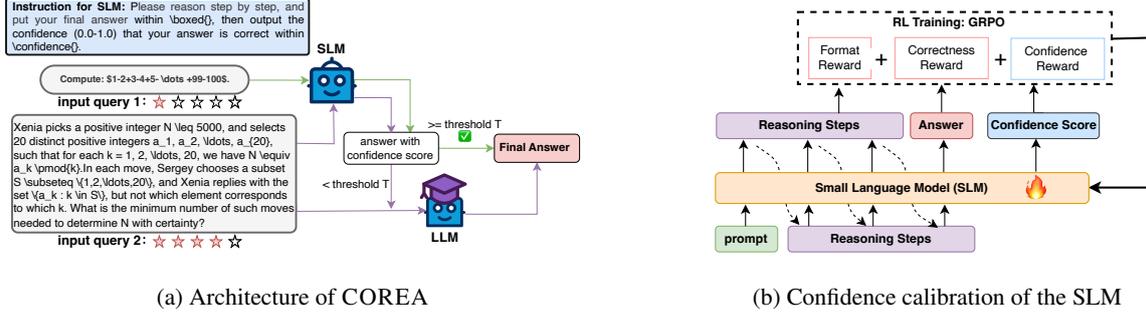


Figure 1: (a) **Architecture of COREA**: COREA cascades an SLM with an LLM. For each query, the SLM is prompted to generate reasoning steps, an answer, and a confidence score. LLM is invoked only when the SLM’s confidence is less than the predefined threshold. (b) **Confidence calibration of the SLM**: The SLM is trained using GRPO with a multi-objective reward function to improve its reasoning ability and calibrate its confidence.

query is passed to the LLM, whose response serves as the final answer.

To elicit the desired response format, we append the instruction after each question: *"Please reason step by step, and put your final answer within `\boxed{\}`, then output the confidence (0.0-1.0) that your answer is correct within `\confidence{\}`."*

3.2 Calibrating Confidence through RL

RL with verifiable rewards (RLVR) focuses only on improving a model’s reasoning ability. RLCC further aligns the model’s confidence by adding additional confidence calibration rewards (abbreviated as confidence reward) on top of the verifiable rewards. Moreover, we incorporate a format reward to ensure the model produces well-structured outputs containing reasoning steps, a final answer, and a confidence score.

Formally, given an input question x , the SLM is prompted to generate reasoning steps y_r , an answer y_a , and a confidence score $y_c \in [0, 1]$ using the instruction in Section 3.1. Let y_a^* denote the ground truth answer and p represent the probability that the SLM can answer the question correctly. The composite reward function is:

$$R = R_{\text{correct}} + R_{\text{format}} + R_{\text{confidence}}, \quad (1)$$

where $R_{\text{correct}} = \mathbb{I}(y_a = y_a^*)$ is the binary correctness reward. R_{format} is a simple format reward which enforces consistent language usage, output of answer and confidence score as required in the prompt, and is normalized to $[0, 1]$.

The confidence reward $R_{\text{confidence}}$ should encourage the model to generate confidence scores that aligns with its correctness probability. It is defined as the negative distance between the confidence

score y_c and the correctness probability p . Several distance metrics can be employed, including L1 distance, L2 distance, and KL distance. Correspondingly, the confidence rewards are

$$R_{L1}(p, y_c) = -|p - y_c| \quad (2)$$

$$R_{L2}(p, y_c) = -(p - y_c)^2 \quad (3)$$

$$R_{KL}(p, y_c) = -p \log \frac{p}{y_c} - (1 - p) \log \frac{1 - p}{1 - y_c} \quad (4)$$

R_{KL} is unbounded, and can lead to instability in training, so we make some clipping, normalization, and remove the constant term to obtain:

$$R_{KL}(p, y_c) = p \frac{\log(\max(y_c, \epsilon))}{\log \epsilon} + (1 - p) \frac{\log(\max(1 - y_c, \epsilon))}{\log \epsilon}, \quad (5)$$

where $\epsilon \in (0, 1)$ is a clipping factor.

Since the correctness probability p is unavailable, we estimate it using group accuracy during GRPO rollouts. Specifically, for a question-answer pair $(x, y_a^*) \sim \mathcal{D}$, N responses $(y_r^0, y_a^0, y_c^0), \dots, (y_r^{N-1}, y_a^{N-1}, y_c^{N-1})$ are sampled from the SLM, then p is estimated as:

$$\hat{p} = \frac{1}{N} \sum_{i=0}^{N-1} \mathbb{I}(y_a^i = y_a^*). \quad (6)$$

Our confidence reward definition is on group level, i.e., each sampled response (y_r^i, y_a^i, y_c^i) in the same group would use the same \hat{p} regardless of its correctness. This is different from the sample level definitions in (Damani et al., 2025; Xu et al., 2024; Stangel et al., 2025). In their definitions, p is estimated as $\hat{p} = \mathbb{I}(y_a^i = y_a^*)$, which depends on the sampled answer’s correctness. We

denote the sample-level confidence rewards as `sample_L1`, `sample_L2` (equivalent to the Brier score in (Damani et al., 2025)), `sample_KL`. We focus on L1 reward and will compare the performance of different rewards in the ablation study.

4 Experimental Setup

4.1 Models and Dataset

SLM: Our main base SLM is Qwen2.5-7B-Instruct (Qwen et al., 2025) (Qwen2.5-7BI). In ablation studies on model backbones, we additionally evaluate Qwen2.5-1.5B-Instruct (Qwen et al., 2025) (Qwen2.5-1.5BI) and Llama3.1-8B-Instruct (Grattafiori et al., 2024) (Llama3.1-8BI). We use instruction-tuned versions for their superior instruction-following capabilities and faster convergence during training.

LLM: We utilize Qwen2.5-32B-Instruct (Qwen et al., 2025) (Qwen2.5-32BI) as the base LLM.

Training and In-Domain Evaluation Datasets: We construct our training and in-domain evaluation sets by randomly sampling from the DeepMath-103K (He et al., 2025) dataset, a meticulously curated mathematical dataset containing 103K question-answer pairs of varying difficulty levels, which also excludes common math benchmark datasets. Specifically, we sample 16K problems for training (referred to as DeepMath16K) and an additional 500 problems as in-domain evaluation dataset (referred to as DeepMath500).

Out-of-Domain (OOD) Evaluation Datasets: We select 5 OOD datasets across mathematical and non-mathematical reasoning tasks. For mathematical reasoning, we utilize Math500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and OlympiadBench (He et al., 2024). For broader reasoning evaluation, we include: 1) GPQA (Rein et al., 2024), a challenging graduate-level science dataset spanning biology, physics, and chemistry with 448 multiple-choice questions; 2) CommonsenseQA (Talmor et al., 2019), a commonsense reasoning dataset containing 1,221 multiple-choice questions. We evaluate correctness using `math_verify` (Hugging Face, 2024) for mathematical datasets and exact match for others.

4.2 Baselines

We consider the following baselines:

Standalone Model:

- **SLM:** Original SLM.
- **RLVR-SLM:** SLM trained using RLVR.

- **Brier-SLM:** SLM trained using RLCC with the Brier score.

- **L1-SLM:** SLM trained using RLCC with the L1 confidence reward.

- **Baseline LLM:** To make a collaborative system useful, the Baseline LLM should have better reasoning ability than the RL-trained SLM, otherwise, there is no need to defer to the LLM. To meet the requirement, we trained the LLM using RLCC with the L1 confidence reward.

Collaborative System: For all systems, we employ the Baseline LLM as the capable LLM. Then the systems differ only in the SLM and the method to generate confidence scores. For simplicity, we use these two factors to denote a system.

- **SLM-Verb:** The original SLM with verbalized confidence.

- **RLVR-SLM-Verb:** The RLVR-SLM with verbalized confidence.

- **RLVR-SLM-AvgProb:** The RLVR-SLM with confidence score obtained by averaging the token probability of the response.

- **RLVR-SLM-Probe:** The RLVR-SLM with a probe to generate confidence score. The probe is a multi-layer perceptron (MLP) which takes as input the average-pooled hidden states from RLVR-SLM’s final layer, and outputs the correctness prediction of RLVR-SLM’s answer. It is trained using the same DeepMath16K, but with answers sampled from RLVR-SLM. Probe model and training details are provided in Appendix A.1.

- **Router + RLVR-SLM:** We employ a router to decide the use of the SLM or the LLM. We assess the RLVR-SLM’s accuracy on DeepMath16k by sampling 8 responses per question, then utilize the accuracy to construct a training dataset for the router with the prompt: "*Question: {Question} Instruction: Estimate and output the probability (0.0-1.0) that a small language model ($\leq 10B$) can answer the question correctly.*" During evaluation, the router generates a probability score for each question. Questions are routed to the RLVR-SLM when the probability exceeds a given threshold, and to the LLM otherwise. The Router utilizes the same base model as RLVR-SLM. Training details are provided in Appendix A.2.

- **Brier-SLM-Verb:** The Brier-SLM with verbalized confidence.

4.3 Evaluation Protocol

The following metrics are considered to evaluate the different systems.

System	DeepMath500			OOD Math			OOD Non-Math		
	Pass@1% ↑	Avg Cost ↓	LLM%	Pass@1% ↑	Avg Cost ↓	LLM%	Pass@1% ↑	Avg Cost ↓	LLM%
Standalone									
SLM	42.7	4423	0.0	60.4	2622	0.0	45.2	2004	0.0
RLVR-SLM	57.6	2511	0.0	71.8	1928	0.0	62.4	1438	0.0
Brier-SLM	55.7	3490	0.0	72.7	2322	0.0	63.5	1708	0.0
L1-SLM*	56.6	3766	0.0	72.1	2589	0.0	64.7	1902	0.0
Baseline LLM	69.0	14882	100.0	79.6	10164	100.0	71.4	7428	100.0
Collaborative									
SLM-Verb	60.3 ^(-8.6)	11594 ^(-22.1%)	43.6	75.7 ^(-3.9)	7300 ^(-28.2%)	34.9	48.8 ^(-22.6)	2609 ^(-64.9%)	9.4
RLVR-SLM-Verb	57.6 ^(-11.4)	2672 ^(-82.0%)	1.0	72.6 ^(-7.0)	2927 ^(-71.2%)	14.0	62.7 ^(-8.7)	1510 ^(-79.7%)	1.1
RLVR-SLM-AvgProb	57.6 ^(-11.4)	2511 ^(-83.1%)	0.0	72.4 ^(-7.1)	2206 ^(-78.3%)	4.5	62.5 ^(-8.9)	1488 ^(-80.0%)	0.9
RLVR-SLM-Probe	63.3 ^(-5.7)	9562 ^(-35.7%)	41.7	75.9 ^(-3.7)	5874 ^(-42.2%)	25.2	67.3 ^(-4.1)	4724 ^(-36.4%)	32.7
Router+RLVR-SLM	61.4 ^(-7.5)	7659 ^(-48.5%)	35.6	74.4 ^(-5.2)	4974 ^(-51.1%)	20.6	65.4 ^(-6.0)	3781 ^(-49.1%)	34.0
Brier-SLM-Verb	66.3 ^(-2.7)	15254 ^(+2.5%)	72.5	78.8 ^(-0.8)	9264 ^(-8.8%)	57.4	70.3 ^(-1.2)	7188 ^(-3.2%)	78.9
COREA *	67.5 ^(-1.5)	13882 ^(-6.7%)	59.9	77.9 ^(-1.7)	7978 ^(-21.5%)	35.5	69.7 ^(-1.7)	6178 ^(-16.8%)	59.0

Table 1: Performance comparison across different systems and datasets. Pass@1, average cost per question (Avg Cost), and LLM usage percentage (LLM%) are reported. OOD Math includes Math500, GSM8K, and OlympiadBench. OOD Non-Math includes GPQA and CommonsenseQA. In all collaborative systems, the Baseline LLM’s Pass@1 serves as the confidence threshold. Values in parentheses indicate the difference from the Baseline LLM. COREA (L1-SLM-Verb) utilizes L1-SLM and verbalized confidence. * indicates our method.

For the SLM component:

Pass@1: A common measure of accuracy.

Expected Calibration Error (ECE): Measures the alignment between predicted confidence and actual accuracy through binning:

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)|, \quad (7)$$

where B_m represents the m -th confidence bin, $\text{acc}(B_m)$ denotes the empirical accuracy of samples in B_m , and $\text{conf}(B_m)$ corresponds to the mean confidence score of predictions in B_m .

AUROC: Computed as the area under the ROC curve generated by varying the confidence threshold. For binary classification of answer correctness:

$$\text{AUROC} = \int_0^1 \text{TPR}(t) \cdot \text{FPR}'(t) dt, \quad (8)$$

where TPR (True Positive Rate) and FPR (False Positive Rate) are functions of threshold t .

Confidence Output Ratio: Measures the ratio of responses with a verbalized confidence score.

For the collaborative system:

Pass@1: Accuracy of the final answer.

System Cost: The system cost depends on the number of input and output tokens. We assume that 1) the cost of output tokens is four times that of input tokens, which aligns with prevailing API pricing models (e.g., GPT 4:1, while Claude 5:1). 2) pricing scales proportionally with model size. For a question, the SLM/Router cost is defined as:

$$\text{SLM/Router Cost} = n_{\text{prompt}} + 4 \cdot n_{\text{SLM}}, \quad (9)$$

where n_{prompt} and n_{SLM} represent the numbers of prompt and output tokens, respectively. Each SLM input token is normalized to a base cost of 1.

The LLM cost is defined as:

$$\text{LLM Cost} = (n_{\text{prompt}} + 4 \cdot n_{\text{LLM}}) \cdot \frac{|\theta_{\text{LLM}}|}{|\theta_{\text{SLM}}|}, \quad (10)$$

where n_{LLM} represents the number of output tokens of the LLM, and the term $\frac{|\theta_{\text{LLM}}|}{|\theta_{\text{SLM}}|}$ adjusts for the relative cost scaling based on parameter counts.

The cost of a system without a router is:

$$\text{System Cost} = \begin{cases} \text{SLM Cost}, & y_c \geq p_T \\ \text{SLM Cost} + \text{LLM Cost}, & y_c < p_T \end{cases}$$

where y_c and p_T are the confidence score of the SLM and the confidence threshold, respectively.

The system cost of Router + RLVR-SLM is:

$$\text{System Cost} = \begin{cases} \text{Router Cost} + \text{SLM Cost}, & y_c \geq p_T \\ \text{Router Cost} + \text{LLM Cost}, & y_c < p_T \end{cases}$$

where y_c is the correctness probability of the router.

4.4 Implementation Details

We conduct training on an AWS p5en.48xlarge instance equipped with 8xH200 GPUs using the SimpleRL-reason framework (Zeng et al., 2025). For GRPO training, we use the framework’s default hyperparameters with the following key settings: sampling temperature of 1.0, maximum response length of 4,096 tokens, and a learning rate of 5×10^{-7} with AdamW optimizer. Each training episode consists of two phases: (1) a rollout

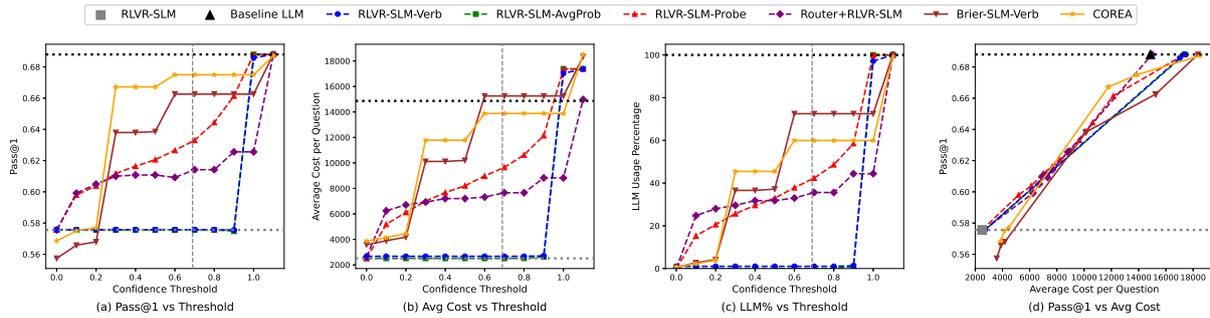


Figure 2: Results on DeepMath-500: (a) Pass@1, (b) average cost, and (c) LLM usage percentage versus confidence threshold; (d) Pass@1 versus average cost for different systems. The vertical dashed lines in (a)-(c) mark the Baseline LLM’s Pass@1 (0.69).

phase generating 8 responses per question with a batch size of 1,024, and (2) a training phase using a batch size of 256. The loss function includes KL divergence regularization (coefficient: 1×10^{-4}) to prevent policy drift and entropy regularization (coefficient: 1×10^{-3}) to encourage exploration. We trained Qwen2.5-1.5BI, Qwen2.5-7BI, Llama3.1-8BI, and Qwen2.5-32BI for 150, 80, 120, and 50 steps, respectively, with training duration determined by total reward convergence. During inference, we set the sampling temperature to 0.6 to reduce output variability and improve consistency. All evaluation results are averaged over 10 runs unless otherwise stated.

5 Results

5.1 Overall Performance

The main results are provided in Table 1. Detailed per-dataset results and experiments with an alternative SLM+LLM pairing are provided in Appendix C and Appendix D, respectively. From the table, we observe that:

Confidence reward does not hurt correctness: This is demonstrated by comparing the standalone models. Compared to RLVR-SLM, both Brier-SLM and L1-SLM achieve comparable Pass@1 performance despite incorporating an additional confidence reward. Furthermore, although all RL trainings are conducted on DeepMath16K, performance improvements extend to the OOD Math and OOD Non-Math datasets as well, demonstrating the good generalizability of the proposed method.

COREA shows superior performance-cost trade-off: In Table 1, we set the Baseline LLM’s Pass@1 as the confidence threshold for all collaborative systems, where the SLM defers to the LLM when its confidence falls below this threshold.

With well-calibrated confidence, the system should achieve substantial cost reductions while maintaining performance, as confident questions are answered by the SLM alone. Our COREA achieves this goal, reducing costs by 6.7%, 21.5%, and 16.8% on DeepMath500, OOD Math, and OOD Non-Math respectively, while limiting Pass@1 degradation to within 2 percentage points.

While some systems achieve either higher Pass@1 or lower costs in certain cases—for instance, Brier-SLM-Verb attains higher Pass@1 on OOD datasets—this comes at the expense of reduced savings. Conversely, systems like RLVR-SLM-Verb, RLVR-SLM-AvgProb, RLVR-SLM-Probe, and Router-RLVR-SLM achieve greater cost reductions but suffer significant Pass@1 drops due to poorly calibrated confidence scores. These results demonstrate that our method successfully maintains near-optimal performance while achieving substantial cost savings through intelligent SLM-LLM collaboration.

5.2 Confidence Threshold Selection

In Table 1, the confidence threshold is fixed at the Baseline LLM’s Pass@1. However, the collaborative system offers flexibility to trade off performance and cost by adjusting this threshold. As shown in Figure 2, when the threshold increases from 0 to 1.1, system Pass@1 increases alongside average cost due to higher LLM usage across all methods. If cost is prioritized, we can sacrifice accuracy for savings. For instance, decreasing the confidence threshold from 0.69 to 0.4 for L1-SLM-Verb yields an additional 14% relative cost reduction with only 0.6 percentage point Pass@1 loss. While other methods achieve greater cost reductions, they incur more severe accuracy drops. We set 1.1 as the maximum threshold since the SLM’s

Method	Pass@1% \uparrow	ECE \downarrow	AUROC \uparrow
SLM-Verb	52.8 \pm 0.9	0.45 \pm 0.01	0.50 \pm 0.01
RLVR-SLM-Verb	66.9 \pm 0.7	0.30 \pm 0.01	0.50 \pm 0.01
RLVR-SLM-AvgProb	66.9 \pm 0.7	0.28 \pm 0.01	0.64 \pm 0.01
RLVR-SLM-Probe	66.9 \pm 0.7	0.17 \pm 0.01	0.72 \pm 0.01
Router+RLVR-SLM	66.9 \pm 0.7	0.20 \pm 0.01	0.56 \pm 0.01
Brier-SLM-Verb	67.5 \pm 0.7	0.17 \pm 0.01	0.72 \pm 0.01
L1-SLM-Verb*	67.8 \pm 0.7	0.12 \pm 0.01	0.72 \pm 0.01

Table 2: Comparison of different methods on accuracy and calibration metrics, averaged across all datasets. Here, the metrics are only evaluated on the SLM.

verbalized confidence caps at 1.0, thus only values > 1.0 ensure all questions defer to the LLM. Figure 2d draws Pass@1 versus average cost curves of different methods, although COREA is not Pareto optimal across all regions, it achieves the best accuracy-cost trade-off in high Pass@1 regions ([0.62, 0.67]).

5.3 Evolution of SLM Self-Awareness

Training Dynamics and Convergence: Figure 3a demonstrates that trainings with and without confidence rewards (either Brier Score or L1 confidence reward) achieve comparable correctness and format rewards, indicating that the SLM can simultaneously improve its reasoning ability and align its confidence estimates. The evolution of key metrics in Figure 3b (Pass@1, ECE, AUROC, and Confidence Output Ratio) further supports this finding. Figure 3c provides a clearer view of the confidence-accuracy alignment process: initially (step 0), the SLM exhibits severe overconfidence with nearly all confidence scores approaching 1.0. Through training, the model progressively diversifies its confidence estimates and aligns them more closely with actual accuracies.

Quantitative Results: Table 2 shows the averaged accuracy and calibration results of different methods on all datasets (detailed results on each dataset are presented in Appendix E). It is noted that RL training improves the SLM’s reasoning performance significantly. Among all methods, the proposed L1-SLM-Verb method achieves the highest Pass@1 and lowest ECE, showing the superiority of the proposed method.

5.4 Ablation Study

5.4.1 Generalizability: Robustness Across SLM Sizes and Architectures

We further conducted RLVR training and RLCC training with L1 confidence reward (RLCC-L1) on Qwen2.5-1.5B, Llama3.1-8B, and Qwen2.5-32B.

Conf. Reward	Pass@1% \uparrow	ECE \downarrow	AUROC \uparrow
L1	67.8 \pm 0.7	0.12 \pm 0.01	0.72 \pm 0.01
Sample_L1	68.3 \pm 0.7	0.16 \pm 0.01	0.68 \pm 0.01
L2	67.5 \pm 0.7	0.31 \pm 0.01	0.62 \pm 0.01
Brier	67.5 \pm 0.7	0.17 \pm 0.01	0.72 \pm 0.01
KL	66.5 \pm 0.7	0.15 \pm 0.01	0.69 \pm 0.01
Sample_KL	67.1 \pm 0.7	0.11 \pm 0.01	0.74 \pm 0.01

Table 3: Comparison of different confidence reward functions on accuracy and calibration metrics, averaged across all datasets.

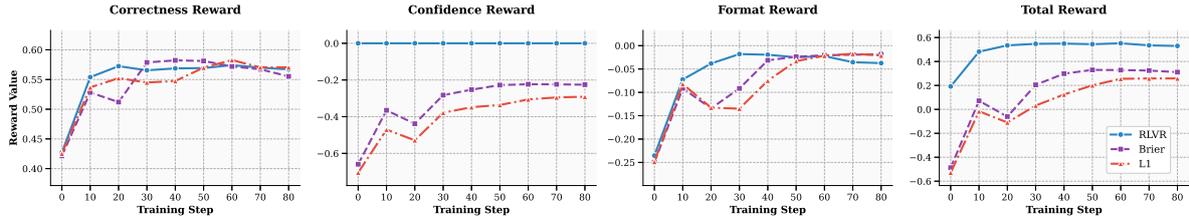
Figure 4 demonstrates that the proposed RLCC method is effective across all tested model sizes and architectures. In terms of reasoning performance, both RLVR and RLCC-L1 substantially improve capabilities for smaller models, with the exception of Qwen2.5-32B, which already exhibits strong baseline performance. For calibration, RLCC-L1 consistently reduces ECE by a large margin across all models, while RLVR yields only slight ECE reductions or even degradations.

5.4.2 Different Reward Functions

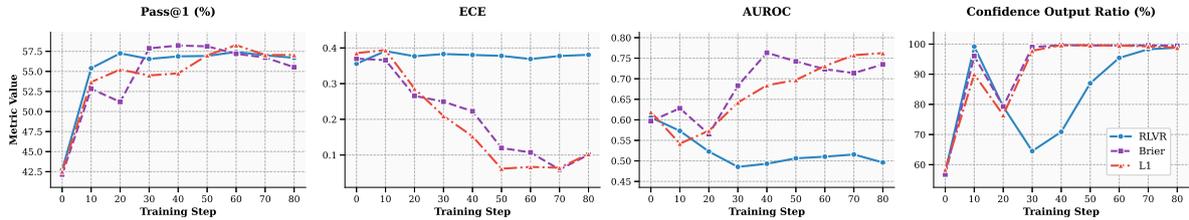
We compare the performance of RLCC with different confidence rewards in Table 3. It can be concluded that RLCC training with these confidence rewards achieve comparable performances in terms of accuracy and confidence calibration, except for the L2 reward, which has a much higher ECE than the others. We hypothesize that this phenomenon stems from instabilities during training. Specifically, although the overall reward plateaus, the model continues to adjust among different rewards. As we adopt a fixed stopping step across all configurations, the selected checkpoint may not represent the optimal point for the L2 reward. Interestingly, even Sample_L1 is not a proper function (Damani et al., 2025) (which means that theoretically the highest reward is not attained at $y_c = p$ in Equation (2)), training with it can still achieve similar performance as other proper functions. Among all, we recommend L1 reward as it achieves a better balance between Pass@1 and ECE.

5.4.3 Impact of Rollout Size

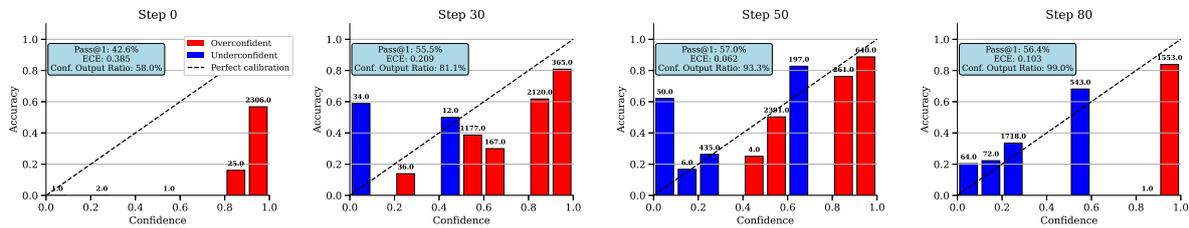
Table 4 presents the performance of different rollout sizes in GRPO training with L1 confidence reward. Contrary to the expectation that larger rollout sizes would monotonically improve both accuracy and calibration through increased exploration diversity and better probability estimation, the results reveal a more nuanced relationship. Rollout size 32



(a) Reward evolution.



(b) Pass@1, ECE, AUROC, Confidence Output Ratio evolution.



(c) Confidence alignment evolution of L1, the number on each bar indicates how many answers are in the confidence bin.

Figure 3: Training process of Qwen2.5-7BI with different reward configurations: RLVR, Brier (RLCC with Brier score), and L1 (RLCC with L1 confidence reward) on DeepMath500, with 8 sampled answers per question.

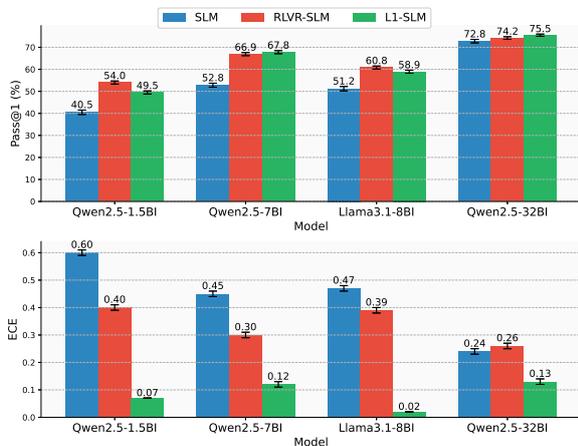


Figure 4: Comparison of Pass@1 and ECE across several models trained with different reward configurations, averaged across all datasets. Detailed results on each dataset are provided in Appendix E.

achieves the highest Pass@1 (68.3%) but exhibits degraded calibration metrics (ECE: 0.17, AUROC: 0.65). Conversely, rollout size 16 attains the best calibration metrics (ECE: 0.10, AUROC: 0.69), yet suffers a slight drop in Pass@1 to 66.7%. Rollout size 8 demonstrates a favorable balance, achieving competitive Pass@1 while maintaining strong cal-

Rollout Size	Pass@1% \uparrow	ECE \downarrow	AUROC \uparrow
4	67.2 \pm 0.8	0.10 \pm 0.01	0.68 \pm 0.01
8	67.8 \pm 0.7	0.12 \pm 0.01	0.72 \pm 0.01
16	66.7 \pm 0.7	0.10 \pm 0.01	0.69 \pm 0.01
32	68.3 \pm 0.5	0.17 \pm 0.01	0.65 \pm 0.01

Table 4: Comparison of different rollout sizes with L1 confidence reward on accuracy and calibration metrics, averaged across all datasets.

ibration. This indicates that the proposed method is not sensitive to rollout size, even with a small rollout size like 4 could perform well.

6 Conclusion

We introduced **COREA**, a collaborative reasoning framework in which an SLM is trained with RL to calibrate its confidence estimates and strategically defer challenging questions to an LLM when uncertain. Through extensive experiments, we demonstrated that the proposed RLCC training method simultaneously enhances the SLM’s reasoning capabilities and aligns its confidence scores across multiple SLM backbones and model scales. These improvements generalize effectively to both mathematical and non-mathematical rea-

soning tasks. Furthermore, COREA demonstrated superior performance-cost tradeoffs compared to strong baselines. By intelligently deferring questions to the LLM based on calibrated confidence scores, our approach achieved substantial cost reductions of 7% ~ 22% while maintaining accuracy within 2 percentage points of the baseline LLM. These results highlight that well-calibrated confidence serves as a powerful mechanism for enabling practical and efficient SLM-LLM collaboration in real-world deployment scenarios.

Limitations

While the proposed framework shows promising results, several limitations remain. First, the verbalized confidence score after training still tend to be discrete instead of continuous although the prompt instructs the model to output a continuous value in $[0, 1]$. This leads to a non-smooth transition of accuracy and cost when we move the confidence threshold. Second, in rare cases, the RL training process can still encounter instabilities, and the model can still balance among different reward components even when the composite reward converges. Hence, how to ensure all components simultaneously converge is worth further study. Finally, we did not extensively discuss different SLM and LLM combinations although we provided a different SLM-LLM pairing in the appendix, however, the parameter count disparity between the SLM and LLM has a significant impact on the cost gain. This also hints that the cost reduction estimation in this paper can be conservative, if LLM is significantly larger than the SLM, the possible gain could be much larger. These factors constrain the current scope of the study and point to directions for further investigation.

References

- Yash Akhauri, Anthony Fei, Chi-Chih Chang, Ahmed F. AbouElhamayed, Yueying Li, and Mohamed S. Abdelfattah. 2025. *Splitreason: Learning to offload reasoning*. *Preprint*, arXiv:2504.16379.
- Anthropic. 2025. Introducing claude 4: Claude opus 4 and claude sonnet 4. <https://www.anthropic.com/news/claude-4>. Accessed: 2025-07-23.
- Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. *arXiv preprint arXiv:2503.05179*.
- Amos Azaria and Tom M. Mitchell. 2023. The internal state of an LLM knows when it’s lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 967–976. Association for Computational Linguistics.
- Li Chenglin, Qianglong Chen, Liangyue Li, Caiyu Wang, Feng Tao, Yicheng Li, Zulong Chen, and Yin Zhang. 2024. Mixed distillation helps smaller language models reason better. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1673–1690.
- 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 Shencfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. 2025. Beyond binary rewards: Training lms to reason about their uncertainty. *CoRR*, abs/2507.16806.
- DeepSeek-AI, Daya Guo, Dejian Yang, and et al. 2025. *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*. *Preprint*, arXiv:2501.12948.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al. 2024. *The llama 3 herd of models*. *Preprint*, arXiv:2407.21783.
- Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkritum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language model cascades: Token-level uncertainty and beyond. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. *Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems*. *Preprint*, arXiv:2402.14008.
- Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. *Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning*. *Preprint*, arXiv:2504.11456.

- 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*. *Preprint*, arXiv:2103.03874.
- Hugging Face. 2024. *Math-verify*. <https://github.com/huggingface/Math-Verify>.
- Sanyam Kapoor, Nate Gruver, Manley Roberts, Katie Collins, Arka Pal, 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 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Yujin Kim, Euiin Yi, Minu Kim, Se-Young Yun, and Taehyeon Kim. 2025. *Guiding reasoning in small language models with llm assistance*. *Preprint*, arXiv:2504.09923.
- Byeongchan Lee, Jonghoon Lee, Dongyoung Kim, Jaehyung Kim, and Jinwoo Shin. 2025. *Collaborative llm inference via planning for efficient reasoning*. *Preprint*, arXiv:2506.11578.
- Jooyoung Lee, Fan Yang, Thanh Tran, Qian Hu, Emre Barut, and Kai-Wei Chang. 2024. Can small language models help large language models reason better?: Lm-guided chain-of-thought. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 2835–2843. ELRA and ICCL.
- Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. 2025. Taming overconfidence in llms: Reward calibration in RLHF. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025a. *Limr: Less is more for rl scaling*. *Preprint*, arXiv:2502.11886.
- Yuetai Li, Xiang Xue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. 2025b. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*.
- Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Xiao Liang, Zhi-jiang Guo, and 2 others. 2025c. *From system 1 to system 2: A survey of reasoning large language models*. *Preprint*, arXiv:2502.17419.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. *s1: Simple test-time scaling*. *Preprint*, arXiv:2501.19393.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. *Routellm: Learning to route llms with preference data*. *arXiv preprint arXiv:2406.18665*.
- OpenAI. 2025. *Introducing gpt-5*. <https://openai.com/index/introducing-gpt-5/>. Accessed: 2025-08-07.
- Gabriele Prato, Jerry Huang, Prasanna Parthasarathi, Shagun Sodhani, and Sarath Chandar. 2024. Do large language models know how much they know? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 6054–6070. Association for Computational Linguistics.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, and 25 others. 2025. *Qwen2.5 technical report*. *Preprint*, arXiv:2412.15115.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. *GPQA: A graduate-level google-proof q&a benchmark*. In *First Conference on Language Modeling*.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.
- Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. 2025. *Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models*. *Preprint*, arXiv:2503.02623.
- Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. *Fast best-of-n decoding via speculative rejection*. *arXiv preprint arXiv:2410.20290*.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. *CommonsenseQA: A question answering challenge targeting commonsense knowledge*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

- Rui Wang, Hongru Wang, Boyang Xue, Jianhui Pang, Shudong Liu, Yi Chen, Jiahao Qiu, Derek Fai Wong, Heng Ji, and Kam-Fai Wong. 2025. Harnessing the reasoning economy: A survey of efficient reasoning for large language models. *Preprint*, arXiv:2503.24377.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.
- Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. 2024. Sayself: Teaching llms to express confidence with self-reflective rationales. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 5985–5998. Association for Computational Linguistics.
- Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. 2025. On verbalized confidence scores for LLMs. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*.
- Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8653–8665. Association for Computational Linguistics.
- Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*.
- Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. 2023. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. *CoRR*, abs/2310.03094.
- Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *Preprint*, arXiv:2503.18892.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Appendix

A Training Details

A.1 Training Details of Probe

Model Architecture. The Probe takes the average-pooled hidden states from the last layer of the SLM as input and outputs the probability that the SLM’s answer is correct. We use a 2-layer MLP with dimensions (2048, 1024) for the two layers. The final head is a sigmoid function.

Training Dataset. The Probe is trained to predict the RLVR-SLM’s correctness. We first use RLVR-SLM to generate a response for each question in DeepMath16K, then evaluate the correctness of each response as the ground truth. Specifically, each item in the training dataset is formatted as ((question, answer), correctness). We use a 9:1 train-eval split.

Training Details. We conduct training on an AWS p5en.48xlarge instance equipped with 8×H200 GPUs. We use a per-device batch size of 32 and a learning rate of 5×10^{-5} with the AdamW optimizer. We train for 20 epochs until the loss plateaus on the evaluation dataset.

A.2 Training Details of Router

Model Architecture. The Router is used to decide whether to allocate a question to the SLM or the LLM. It evaluates the probability that the SLM can answer the question correctly. To achieve this, we use the same base model as the SLM to serve as the Router. We find that directly prompting the SLM to make the prediction is highly inaccurate, so we use supervised fine-tuning to align the Router’s predictions.

Training Dataset. The Router is trained to estimate the RLVR-SLM’s accuracy on each question. We first use RLVR-SLM to generate 8 responses for each question in DeepMath16K, then compute the accuracy as the ground truth for the question. The prompt is: *"Question: {Question} Instruction: Estimate and output the probability (0.0-1.0) that a small language model ($\leq 10B$) can answer the question correctly."* We then convert the dataset into the ShareGPT format.

Training Details. We conduct training on an AWS p5en.48xlarge instance. We use LLaMA-Factory (Zheng et al., 2024) as the training framework. The per-device batch size is 8, gradient accumulation steps is 4, and the learning rate is 5×10^{-5} with the AdamW optimizer. We train for 2 epochs

until the loss plateaus on the evaluation dataset.

B Sample Output

We use the following model abbreviations throughout this paper: The "I" suffix indicates instruction-tuned model variants (e.g., 7BI = 7B-Instruct).

- **Qwen2.5-7BI:** Original Qwen2.5-7B-Instruct model without additional training.
- **RLVR-Qwen2.5-7BI:** Qwen2.5-7B-Instruct trained with reinforcement learning with verifiable reward (RLVR).
- **L1-Qwen2.5-7BI:** Qwen2.5-7B-Instruct trained with RL with an additional L1 confidence reward.

Figures 5–8 show the responses of Qwen2.5-7BI, RLVR-Qwen2.5-7BI, and L1-Qwen2.5-7BI on a DeepMath500 problem, respectively. As demonstrated, Qwen2.5-7BI produces an incorrect answer, while both RLVR-Qwen2.5-7BI and L1-Qwen2.5-7BI solve the problem correctly, demonstrating the effectiveness of RL training in improving the model’s reasoning ability.

Additionally, Qwen2.5-7BI fails to follow the instruction to output the confidence score in the required format. In contrast, both RLVR-Qwen2.5-7BI and L1-Qwen2.5-7BI successfully output scores in the correct format. However, without confidence calibration, RLVR-Qwen2.5-7BI produces an overconfident score of 1.0, similar to Qwen2.5-7BI. Conversely, L1-Qwen2.5-7BI’s confidence score of 0.8 is better aligned with the model’s true accuracy of 0.75, demonstrating the effectiveness of the confidence reward term in improving confidence calibration.

C Main Results for Each Dataset

Tables 5–10 compare the performance of different systems on DeepMath500, Math500, GSM8K, OlympiadBench, GPQA, and CommonsenseQA, respectively. The confidence threshold is set to match the Baseline LLM’s pass@1 for each dataset. The results align with our main paper findings: except for GPQA, COREA achieves significant cost reductions with only marginal accuracy drops compared to the baseline LLM. Compared to other collaborative systems, COREA achieves a superior accuracy-cost balance. On GPQA, the pass@1 drop is slightly larger (-5.1%) due to less well-calibrated confidence scores.

Question

Given that $a > b > c > d$, find the maximum integer value of n such that

$$\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \geq \frac{n}{a-d}.$$

Please reason step by step, and put your final answer within `\boxed{\}`, then output the confidence (0.0 ~ 1.0) that your answer is correct within `\confidence{\}`.

(Ground Truth Answer: 9)

Qwen2.5-7BI

To find the maximum integer value of n such that the inequality

$$\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \geq \frac{n}{a-d}$$

holds for all $a > b > c > d$, we start by considering the properties of the terms involved. Let's denote $x = a - d$, $y = a - c$, $z = a - b$, and $w = b - c$. Since $a > b > c > d$, we have $x > y > z > w > 0$. The inequality then becomes

$$\frac{1}{z} + \frac{1}{w} + \frac{1}{z-w} \geq \frac{n}{x} = \frac{n}{y+z}.$$

To find the maximum possible value of n , we need to analyze the expression on the left side. Let's consider a specific case where $a - d = 4$, $a - c = 3$, $a - b = 2$, and $b - c = 1$. In this case, $x = 4$, $y = 3$, $z = 2$, and $w = 1$, so the inequality is

$$\frac{1}{2} + \frac{1}{1} + \frac{1}{2-1} \geq \frac{n}{3+2} = \frac{n}{5}.$$

Simplifying the left side, we get

$$\frac{1}{2} + 1 + 1 = \frac{1}{2} + 2 = \frac{5}{2}.$$

Therefore, the inequality becomes

$$\frac{5}{2} \geq \frac{n}{5},$$

which simplifies to

$$n \leq \frac{25}{2} = 12.5.$$

Since n must be an integer, the maximum possible value of n is 12.

Figure 5: Question and Response of Qwen2.5-7BI.

D A Different SLM+LLM Pairing

In this section, we evaluate a 1.5B + 32B model pairing. Specifically, we utilize Qwen2.5-1.5B-Instruct as the base SLM and Qwen2.5-32B-Instruct as the base LLM, training both with the RLCC-L1 algorithm. Results are presented in Table 11.

We select the checkpoint at step 30 rather than step 150, as we observed that the SLM gradually becomes underconfident during training, eventually returning 0 as the confidence score in most cases. At step 30, the model achieves better confidence calibration while maintaining comparable reasoning ability.

As shown, COREA remains effective for OOD

Qwen2.5-7BI continued

To confirm that $n = 12$ is indeed the maximum value, we need to check if the inequality holds for $n = 12$ in general. We rewrite the inequality as

$$\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \geq \frac{12}{a-d}.$$

Using the Cauchy-Schwarz inequality in the form

$$\left(\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \right) ((a-b) + (b-c) + (c-d)) \geq (1+1+1)^2 = 9,$$

we get

$$\left(\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \right) (a-d) \geq 9.$$

This simplifies to

$$\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \geq \frac{9}{a-d}.$$

Since $\frac{9}{a-d} < \frac{12}{a-d}$, the inequality $\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \geq \frac{12}{a-d}$ is indeed satisfied for $n = 12$.

To check if $n = 13$ works, we consider the case where $a = 4$, $b = 3$, $c = 2$, and $d = 1$. Then the inequality becomes

$$\frac{1}{1} + \frac{1}{1} + \frac{1}{1} \geq \frac{13}{3},$$

which simplifies to

$$3 \geq \frac{13}{3},$$

or

$$9 \geq 13,$$

which is false. Therefore, $n = 13$ does not work.

Thus, the maximum integer value of n is 12. The confidence in this answer is 1.0.

Figure 6: Response of Qwen2.5-7BI continued.

Non-Math, achieving substantial cost reduction (−42.4%) with only a modest accuracy drop (−3.8%). However, compared to the 7B+32B results (Table 1), the 1.5B+32B system suffers more substantial performance degradation on DeepMath500 and OOD Math despite achieving greater cost reductions. Analysis reveals that the 1.5B SLM’s confidence is more poorly calibrated than that of the 7B model—specifically, it tends to exhibit overconfidence on problems beyond its capability. This suggests that beyond weaker reasoning ability, smaller models may also demonstrate diminished self-awareness. We plan to investigate the scaling laws of model self-awareness in future work.

E Detailed Confidence Calibration Results

Figure 9 shows the Pass@1 accuracy and Expected Calibration Error (ECE) of different base models on each evaluation dataset. These results indicate that RL training effectively improves the model’s reasoning capability across all in-domain and out-of-domain evaluation datasets, with the exception of the Qwen2.5-32BI model, whose reasoning ability is already sufficiently strong that further improvement is limited.

However, RL training without confidence reward fails to calibrate the model’s confidence predictions with its true accuracy performance, and in some cases even degrades calibration, as observed in the Math500, OlympiadBench, and CommonsenseQA with Qwen2.5-32BI. In contrast, incorporating a

To find the maximum integer value of n such that the inequality

$$\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \geq \frac{n}{a-d}$$

holds for any $a > b > c > d$, we need to analyze the expression and find a suitable value for n . We will use the method of substitution and simplification to derive the value of n .

First, let's consider the case where a, b, c , and d are equally spaced. This will help us to find a potential value for n . If we assume $a = d + 3x$, $b = d + 2x$, and $c = d + x$, then the differences are $a - b = x$, $b - c = x$, and $c - d = x$. Substituting these into the inequality, we get:

$$\frac{1}{x} + \frac{1}{x} + \frac{1}{x} \geq \frac{n}{3x}.$$

This simplifies to:

$$\frac{3}{x} \geq \frac{n}{3x}.$$

By canceling x from both sides (assuming $x > 0$), we get:

$$3 \geq \frac{n}{3} \implies n \leq 9.$$

This tells us that the maximum possible value for n is 9. Now, we need to verify that $n = 9$ is indeed the maximum value that satisfies the inequality for all $a > b > c > d$. To do this, we can use the method of inequalities and specifically the Cauchy-Schwarz inequality in the form of the Titu's lemma (also known as the Engel's form of the Cauchy-Schwarz inequality):

$$\frac{1^2}{a-b} + \frac{1^2}{b-c} + \frac{1^2}{c-d} \geq \frac{(1+1+1)^2}{(a-b) + (b-c) + (c-d)} = \frac{9}{a-d}.$$

This inequality shows that the left side is always at least $\frac{9}{a-d}$, which confirms that the maximum value of n is indeed 9.

Therefore, the maximum integer value of n is 9. The confidence in this answer is very high, as it is derived from a general inequality and a specific case analysis.

The final answer is:

9

The confidence in this answer is:

\confidence{1.0}

Figure 7: Response of RLVR-Qwen2.5-7BI.

confidence reward into RL training enables the model to effectively align its confidence with true accuracy, as evidenced by significant ECE reductions across all base models and datasets.

F Results of Different Confidence Reward Functions on Each Dataset

Tables 12 and 13 present detailed accuracy and calibration results for each dataset. As observed,

except for L2, all other confidence reward functions achieve comparable pass@1, ECE, and AU-ROC across the six datasets. This indicates that L1, Sample_L1, Brier, KL, and Sample_KL are all effective in the RL training algorithm. Among these functions, we recommend L1 as it demonstrates the strongest balance between accuracy and calibration.

L1-Qwen2.5-7BI

To solve the inequality

$$\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \geq \frac{n}{a-d},$$

we need to find the maximum integer value of n such that the inequality holds for all $a > b > c > d$.

Step 1: Simplify the Problem First, let's consider the case when the terms in the inequality are equal. This will help us find an upper bound for n . If we assume $a - b = b - c = c - d$, then we can simplify the inequality.

Step 2: Assume Equal Differences Let $a - b = b - c = c - d = x$. Then, we have:

$$a = d + 3x.$$

Step 3: Substitute and Simplify Substitute these values into the inequality:

$$\frac{1}{x} + \frac{1}{x} + \frac{1}{x} \geq \frac{n}{3x}.$$

This simplifies to:

$$\frac{3}{x} \geq \frac{n}{3x}.$$

Step 4: Solve for n We can cancel x from both sides of the inequality (since $x > 0$):

$$3 \geq \frac{n}{3}.$$

Multiplying both sides by 3 gives:

$$9 \geq n.$$

Step 5: Verify the Maximum Value We need to verify that $n = 9$ is indeed the maximum value that satisfies the inequality for all $a > b > c > d$. To do this, we use the Cauchy-Schwarz inequality in the form of the Titu's lemma (also known as the Engel's form of the Cauchy-Schwarz inequality).

Step 6: Apply Titu's Lemma Titu's lemma states that for any real numbers x_i and positive real numbers y_i :

$$\frac{x_1^2}{y_1} + \frac{x_2^2}{y_2} + \dots + \frac{x_k^2}{y_k} \geq \frac{(x_1 + x_2 + \dots + x_k)^2}{y_1 + y_2 + \dots + y_k}.$$

Applying Titu's lemma to our inequality, we get:

$$\frac{1}{a-b} + \frac{1}{b-c} + \frac{1}{c-d} \geq \frac{(1+1+1)^2}{(a-b) + (b-c) + (c-d)} = \frac{9}{a-d}.$$

Step 7: Conclusion From the application of Titu's lemma, we see that the inequality holds with $n = 9$. Therefore, the maximum integer value of n is indeed 9.

9

\confidence{0.8}

Figure 8: Response of L1-Qwen2.5-7BI (our approach).

G Rollout Size Results on Each Dataset

Tables 14 and 15 present detailed results across different rollout sizes for each dataset. The metric

differences are marginal when increasing rollout size from 4 to 32, indicating that our RL training method is robust to rollout size selection. From

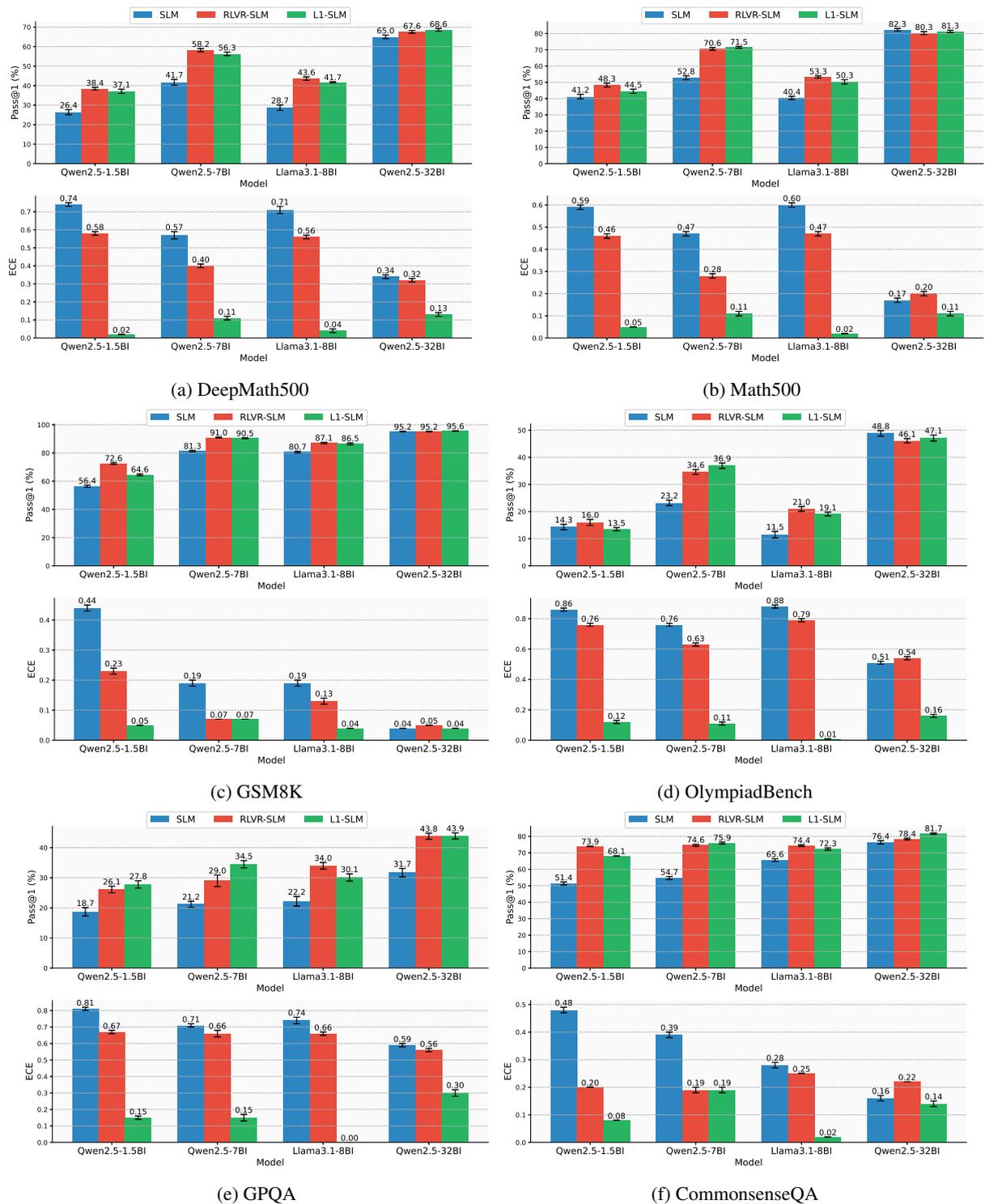


Figure 9: Comparison of Pass@1 and ECE across several models trained with different reward configurations on DeepMath500, Math500, GSM8K, OlympiadBench, GPQA, OlympiadBench, respectively.

System	Pass@1 %	SLM Cost	LLM Cost	Total Cost	LLM%
<i>Standalone</i>					
SLM	42.7	4423	0	4423	0
RLVR-SLM	57.6	2511	0	2511	0
Brier-SLM	55.7	3490	0	3490	0
L1-SLM*	56.6	3766	0	3766	0
Baseline LLM	69.0	0	14882	14882	100
<i>Collaborative</i>					
SLM-Verb	60.3 (-8.6)	4423	7171	11594 (-22.1%)	43.6
RLVR-SLM-Verb	57.6 (-11.4)	2511	160	2672 (-82.0%)	1.0
RLVR-SLM-AvgProb	57.6 (-11.4)	2511	0	2511 (-83.1%)	0
RLVR-SLM-Probe	63.3 (-5.7)	2511	7051	9562 (-35.7%)	41.7
Router+RLVR-SLM	61.4 (-7.5)	1503	6040	7659 (-48.5%)	35.6
Brier-SLM-Verb	66.3 (-2.7)	3490	11765	15254 (+2.5%)	72.5
COREA *	67.5 (-1.5)	3766	10116	13882 (-6.7%)	59.9

Table 5: Performance comparison across different systems on DeepMath500. Pass@1, SLM Cost per question, LLM Cost per question, Total Cost per question, and LLM usage percentage (LLM%) are reported. In all collaborative systems, the Baseline LLM’s Pass@1 serves as the confidence threshold. Values in parentheses indicate the difference from the Baseline LLM. COREA (L1-SLM-Verb) utilizes L1-SLM and verbalized confidence. * indicates our method.

a training efficiency perspective, we recommend using a rollout size of 4 or 8.

System	Pass@1%	SLM Cost	LLM Cost	Total Cost	LLM%
<i>Standalone</i>					
SLM	54.1	3113	0	3113	0
RLVR-SLM	70.5	2125	0	2125	0
Brier-SLM	74.4	2627	0	2627	0
L1-SLM*	72.4	3020	0	3020	0
Baseline LLM	81.2	0	11499	11499	100
<i>Collaborative</i>					
SLM-Verb	77.2 (-4.0)	3113	5409	8522 (-25.9%)	42.1
RLVR-SLM-Verb	70.8 (-10.4)	2125	212	2337 (-79.7%)	1.5
RLVR-SLM-AvgProb	70.5 (-10.7)	2125	0	2125 (-81.5%)	0
RLVR-SLM-Probe	77.5 (-3.7)	2125	5126	7252 (-36.9%)	32.4
Router+RLVR-SLM	74.4 (-6.8)	1459	4031	5603 (-51.3%)	25.8
Brier-SLM-Verb	81.0 (-0.2)	2627	8705	11332 (-1.5%)	63.6
COREA *	80.0 (-1.2)	3020	6591	9610 (-16.4%)	43.1

Table 6: Performance comparison across different systems on Math500. Pass@1, SLM Cost per question, LLM Cost per question, Total Cost per question, and LLM usage percentage (LLM%) are reported. In all collaborative systems, the Baseline LLM’s Pass@1 serves as the confidence threshold. Values in parentheses indicate the difference from the Baseline LLM. COREA (L1-SLM-Verb) utilizes L1-SLM and verbalized confidence. * indicates our method.

System	Pass@1%	SLM Cost	LLM Cost	Total Cost	LLM%
<i>Standalone</i>					
SLM	81.7	1388	0	1388	0
RLVR-SLM	90.9	1348	0	1348	0
Brier-SLM	90.9	1369	0	1369	0
L1-SLM*	90.3	1487	0	1487	0
Baseline LLM	95.6	0	6073	6073	100
<i>Collaborative</i>					
SLM-Verb	92.1 (-3.6)	1388	1280	2667 (-56.1%)	19.4
RLVR-SLM-Verb	92.1 (-3.5)	1348	1623	2972 (-51.1%)	25.1
RLVR-SLM-AvgProb	92.2 (-3.4)	1348	525	1874 (-69.1%)	8.4
RLVR-SLM-Probe	92.8 (-2.9)	1348	646	1994 (-67.2%)	9.0
Router+RLVR-SLM	91.2 (-4.4)	1324	104	1532 (-74.8%)	1.5
Brier-SLM-Verb	95.4 (-0.3)	1369	3392	4761 (-21.6%)	51.1
COREA *	94.2 (-1.4)	1487	1204	2691 (-55.7%)	17.1

Table 7: Performance comparison across different systems on GSM8K. Pass@1, SLM Cost per question, LLM Cost per question, Total Cost per question, and LLM usage percentage (LLM%) are reported. In all collaborative systems, the Baseline LLM’s Pass@1 serves as the confidence threshold. Values in parentheses indicate the difference from the Baseline LLM. COREA (L1-SLM-Verb) utilizes L1-SLM and verbalized confidence. * indicates our method.

System	Pass@1%	SLM Cost	LLM Cost	Total Cost	LLM%
<i>Standalone</i>					
SLM	23.4	4671	0	4671	0
RLVR-SLM	35.2	2916	0	2916	0
Brier-SLM	36.0	3960	0	3960	0
L1-SLM*	36.5	4424	0	4424	0
Baseline LLM	47.0	0	17167	17167	100
<i>Collaborative</i>					
SLM-Verb	42.5 (-4.5)	4671	10778	15449 (-10.0%)	59.8
RLVR-SLM-Verb	35.7 (-11.3)	2916	361	3277 (-80.9%)	1.4
RLVR-SLM-AvgProb	35.2 (-11.8)	2916	0	2916 (-83.0%)	0
RLVR-SLM-Probe	41.6 (-5.5)	2916	9519	12434 (-27.6%)	51.4
Router+RLVR-SLM	41.6 (-5.4)	1293	9800	11236 (-34.6%)	54.2
Brier-SLM-Verb	44.9 (-2.1)	3960	12572	16532 (-3.7%)	65.2
COREA *	44.3 (-2.7)	4424	12677	17101 (-0.4%)	65.6

Table 8: Performance comparison across different systems on OlympiadBench. Pass@1, SLM Cost per question, LLM Cost per question, Total Cost per question, and LLM usage percentage (LLM%) are reported. In all collaborative systems, the Baseline LLM’s Pass@1 serves as the confidence threshold. Values in parentheses indicate the difference from the Baseline LLM. COREA (L1-SLM-Verb) utilizes L1-SLM and verbalized confidence. * indicates our method.

System	Pass@1%	SLM Cost	LLM Cost	Total Cost	LLM%
<i>Standalone</i>					
SLM	21.6	3927	0	3927	0
RLVR-SLM	29.5	2451	0	2451	0
Brier-SLM	32.8	3282	0	3282	0
L1-SLM*	34.2	3775	0	3775	0
Baseline LLM	43.4	0	13420	13420	100
<i>Collaborative</i>					
SLM-Verb	22.9 (-20.4)	3927	630	4557 (-66.0%)	4.1
RLVR-SLM-Verb	29.7 (-13.7)	2451	108	2559 (-80.9%)	0.6
RLVR-SLM-AvgProb	29.5 (-13.9)	2451	0	2451 (-81.7%)	0
RLVR-SLM-Probe	39.7 (-3.7)	2451	9302	11753 (-12.4%)	65.5
Router+RLVR-SLM	35.1 (-8.3)	1407	5810	7463 (-44.4%)	42.6
Brier-SLM-Verb	39.0 (-4.3)	3282	7808	11090 (-17.4%)	54.0
COREA *	38.2 (-5.1)	3775	6734	10509 (-21.7%)	45.8

Table 9: Performance comparison across different systems on GPQA. Pass@1, SLM Cost per question, LLM Cost per question, Total Cost per question, and LLM usage percentage (LLM%) are reported. In all collaborative systems, the Baseline LLM’s Pass@1 serves as the confidence threshold. Values in parentheses indicate the difference from the Baseline LLM. COREA (L1-SLM-Verb) utilizes L1-SLM and verbalized confidence. * indicates our method.

System	Pass@1%	SLM Cost	LLM Cost	Total Cost	LLM%
<i>Standalone</i>					
SLM	53.9	1298	0	1298	0
RLVR-SLM	74.5	1067	0	1067	0
Brier-SLM	74.8	1130	0	1130	0
L1-SLM*	75.9	1215	0	1215	0
Baseline LLM	81.7	0	5229	5229	100
<i>Collaborative</i>					
SLM-Verb	58.3 (-23.4)	1298	597	1895 (-63.8%)	11.4
RLVR-SLM-Verb	74.8 (-6.9)	1067	58	1125 (-78.5%)	1.3
RLVR-SLM-AvgProb	74.6 (-7.1)	1067	68	1134 (-78.3%)	1.2
RLVR-SLM-Probe	77.4 (-4.3)	1067	1079	2145 (-59.0%)	20.6
Router+RLVR-SLM	76.5 (-5.2)	737	1614	2431 (-53.5%)	30.9
Brier-SLM-Verb	81.7 (+0.0)	1130	4626	5756 (+10.1%)	88.1
COREA *	81.3 (-0.4)	1215	3373	4589 (-12.3%)	63.8

Table 10: Performance comparison across different systems on CommonsenseQA. Pass@1, SLM Cost per question, LLM Cost per question, Total Cost per question, and LLM usage percentage (LLM%) are reported. In all collaborative systems, the Baseline LLM’s Pass@1 serves as the confidence threshold. Values in parentheses indicate the difference from the Baseline LLM. COREA (L1-SLM-Verb) utilizes L1-SLM and verbalized confidence. * indicates our method.

System	DeepMath500			OOD Math			OOD Non-Math		
	Pass@1% ↑	Avg Cost ↓	LLM%	Pass@1% ↑	Avg Cost ↓	LLM%	Pass@1% ↑	Avg Cost ↓	LLM%
<i>Standalone</i>									
SLM	25.8	3422	0.0	41.5	2435	0.0	42.8	1558	0.0
L1-SLM	26.5	1311	0.0	35.3	1273	0.0	35.7	449	0.0
Baseline LLM	69.0	69450	100.0	79.6	47430	100.0	71.4	34663	100.0
<i>Collaborative</i>									
SLM-Verb	68.7 (-0.3)	72495 (+4.4%)	99.3	79.1 (-0.5)	49692 (+4.8%)	99.0	71.3 (-0.1)	35492 (+2.4%)	98.4
COREA *	55.7 (-13.2)	46490 (-33.1%)	63.1	70.7 (-8.9)	28996 (-38.9%)	75.2	67.6 (-3.8)	19977 (-42.4%)	76.2

Table 11: Performance comparison of 1.5B SLM + 32B LLM pairing across different systems and datasets. Pass@1, average cost per question (Avg Cost), and LLM usage percentage (LLM%) are reported. OOD Math includes Math500, GSM8K, and OlympiadBench. OOD Non-Math includes GPQA and CommonsenseQA. In all collaborative systems, the Baseline LLM’s Pass@1 serves as the confidence threshold. Values in parentheses indicate the difference from the Baseline LLM. * indicates our method.

Method	DeepMath500			Math500			GSM8K		
	Pass@1%	ECE	AUROC	Pass@1%	ECE	AUROC	Pass@1%	ECE	AUROC
SLM	41.7±1.5	0.57±0.02	0.47±0.02	52.8±1.2	0.47±0.01	0.48±0.02	81.3±0.5	0.19±0.01	0.52±0.00
RLVR	58.2±0.8	0.40±0.01	0.49±0.01	70.6±0.8	0.28±0.01	0.50±0.01	91.0±0.3	0.07±0.00	0.49±0.01
L1	56.3±0.9	0.11±0.01	0.77±0.01	71.5±0.6	0.11±0.01	0.81±0.01	90.5±0.3	0.07±0.00	0.78±0.01
Sample_L1	58.4±0.8	0.17±0.01	0.75±0.01	73.6±0.9	0.07±0.01	0.81±0.01	91.1±0.3	0.05±0.00	0.70±0.01
L2	54.8±1.1	0.22±0.02	0.63±0.01	72.6±0.9	0.39±0.01	0.67±0.01	89.7±0.2	0.53±0.00	0.67±0.01
Brier	55.9±1.1	0.12±0.01	0.73±0.01	73.8±1.1	0.15±0.02	0.81±0.01	90.8±0.2	0.20±0.00	0.80±0.01
KL	56.2±0.8	0.12±0.02	0.73±0.02	71.5±1.2	0.16±0.01	0.76±0.02	90.3±0.4	0.23±0.00	0.77±0.01
Sample_KL	54.9±1.2	0.13±0.01	0.75±0.01	74.0±0.7	0.11±0.01	0.86±0.01	90.3±0.5	0.08±0.00	0.81±0.01

Table 12: Comparison of different confidence reward functions on accuracy and calibration metrics for DeepMath500, Math500, and GSM8K. Results are reported as mean ± standard deviation.

Method	OlympiadBench			GPQA			CommonsenseQA		
	Pass@1%	ECE	AUROC	Pass@1%	ECE	AUROC	Pass@1%	ECE	AUROC
SLM	23.2±1.0	0.76±0.01	0.48±0.01	21.2±1.0	0.71±0.01	0.49±0.03	54.7±0.9	0.39±0.01	0.52±0.01
RLVR	34.6±0.9	0.63±0.01	0.49±0.01	29.0±1.9	0.66±0.02	0.50±0.03	74.6±0.6	0.19±0.01	0.52±0.01
L1	36.9±1.0	0.11±0.01	0.77±0.01	34.5±1.2	0.15±0.02	0.54±0.02	75.9±0.6	0.19±0.01	0.63±0.01
Sample_L1	37.5±1.4	0.22±0.01	0.74±0.01	32.3±0.8	0.40±0.01	0.56±0.02	75.8±0.6	0.18±0.01	0.60±0.00
L2	36.5±0.5	0.10±0.00	0.60±0.01	32.2±1.6	0.04±0.01	0.54±0.02	76.9±0.9	0.28±0.01	0.60±0.01
Brier	35.6±0.8	0.10±0.01	0.77±0.01	32.9±1.2	0.11±0.01	0.54±0.02	74.9±0.6	0.22±0.01	0.62±0.01
KL	35.4±0.7	0.09±0.01	0.68±0.01	30.9±1.2	0.14±0.01	0.55±0.01	73.3±0.7	0.12±0.01	0.61±0.01
Sample_KL	36.3±1.2	0.09±0.01	0.82±0.01	32.8±0.9	0.18±0.01	0.55±0.02	73.8±0.5	0.12±0.01	0.63±0.01

Table 13: Comparison of different confidence reward functions on accuracy and calibration metrics for OlympiadBench, GPQA, and CommonsenseQA. Results are reported as mean ± standard deviation.

Rollout Size	DeepMath500			Math500			GSM8K		
	Pass@1%	ECE	AUROC	Pass@1%	ECE	AUROC	Pass@1%	ECE	AUROC
4	55.8±1.0	0.12±0.01	0.67±0.01	71.2±1.0	0.13±0.01	0.74±0.02	91.7±0.4	0.07±0.01	0.73±0.02
8	56.3±0.9	0.11±0.01	0.77±0.01	71.5±0.6	0.11±0.01	0.81±0.01	90.5±0.3	0.07±0.00	0.78±0.01
16	54.7±1.1	0.09±0.01	0.69±0.01	68.9±1.3	0.08±0.01	0.74±0.01	91.3±0.3	0.07±0.00	0.74±0.00
32	57.2±0.9	0.19±0.01	0.73±0.01	72.1±0.5	0.11±0.01	0.78±0.02	91.7±0.2	0.06±0.00	0.62±0.01

Table 14: Comparison of different rollout sizes with L1 confidence reward on accuracy and calibration metrics for DeepMath500, Math500, and GSM8K. Results are reported as mean ± standard deviation.

Rollout Size	OlympiadBench			GPQA			CommonsenseQA		
	Pass@1%	ECE	AUROC	Pass@1%	ECE	AUROC	Pass@1%	ECE	AUROC
4	35.2±0.9	0.07±0.01	0.63±0.02	29.0±1.1	0.20±0.01	0.56±0.02	75.4±0.9	0.11±0.01	0.69±0.01
8	36.9±1.0	0.11±0.01	0.77±0.01	34.5±1.2	0.15±0.02	0.54±0.02	75.9±0.6	0.19±0.01	0.63±0.01
16	33.3±1.0	0.08±0.01	0.69±0.01	31.7±1.2	0.24±0.01	0.55±0.02	75.6±0.5	0.09±0.01	0.66±0.01
32	34.2±0.7	0.20±0.01	0.80±0.01	34.0±1.4	0.42±0.02	0.53±0.02	77.5±0.3	0.21±0.00	0.55±0.01

Table 15: Comparison of different rollout sizes with L1 confidence reward on accuracy and calibration metrics for OlympiadBench, GPQA, and CommonsenseQA. Results are reported as mean ± standard deviation.