Scaling with Confidence: Calibrating Confidence of LLMs for Adaptive Test Time Scaling

Anonymous ACL submission

Abstract

Training large language models (LLMs) with reinforcement learning (RL) has significantly advanced their performance on reasoning and question-answering tasks. However, prevailing RL reward designs typically prioritize response correctness, neglecting to incentivize models to express their confidence accurately. This leads to a critical problem: performance gains are often accompanied by poor calibration between confidence and accuracy, misleading models to overconfidently hallucinate when uncertain. To address this limitation, we propose **C**orrectness and **C**onfidence **C**alibration **R**einforcement **L**earning (**C3RL**), a novel RL algorithm integrating correctness, calibration and dataset-informed reference accuracy rewards together. Comprehensive evaluation across 8 text and multimodal datasets demonstrates that C3RL enhances calibration without sacrificing accuracy, outperforming the current state-of-the-art method in both performance and calibration metrics. Utilizing the well-calibrated verbalized confidence from C3RL, we further introduce **C**onfidence-based **A**daptive Test Time **S**caling (**CAS**), an adjustable inference-time strategy that allocates computational resources based on response confidence. Experiments show that CAS surpasses majority voting on both in-domain and out-of-domain datasets while reducing the inference budget by up to 12.33 times. We believe the synergy of C3RL and CAS paves the way for deploying more reliable and resource-efficient LLMs. The code, data and models will be released.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable performance in reasoning, question answering, and code generation (Brown et al., 2020; Achiam et al., 2023). However, their impressive capabilities are accompanied by a critical failure mode: when faced with uncertainty, LLMs frequently produce overconfident but incorrect outputs—a phenomenon widely referred to as *hallucination* (Ji et al., 2023). Such behavior is not merely an academic curiosity, but can lead to serious consequences, including misinformation propagation and decision failures in safety-critical domains.

To mitigate these risks, it is desirable for LLMs not only to generate correct outputs but also to reliably express their epistemic uncertainty. This property, known as *confidence calibration*, refers to the ability of a model to produce probability estimates that faithfully reflect the likelihood of correctness (Guo et al., 2017). Well-calibrated confidence serves as a foundation to abstain from uncertain answering, defer to human oversight, or dynamically adjust computational budget when confidence is low.

Despite its importance, confidence calibration for LLMs remains underexplored and faces several limitations. First, they typically reduce calibration to a binary classification task (certain vs. uncertain), which is too coarse to capture the nuanced uncertainty of LLM predictions. Second, they often improve calibration at the expense of accuracy, since models may learn to produce deliberately incorrect low-confidence answers to minimize calibration loss. Moreover, they rarely explore the potential of calibrated confidence for downstream decision-making, such as adaptive inference and resource allocation (Zhao et al., 2024).

Our work addresses these gaps by jointly optimizing correctness and calibration within a reinforcement learning framework and by exploiting calibrated confidence for adaptive test-time scaling. Specifically, we design an approach that explicitly teaches LLMs to signal low confidence when they are uncertain . Moreover, we leverage the resulting calibrated confidence to guide early stopping in test time scaling, adjusting more computation to low-confidence samples. To the best of our knowledge, our method is the first to integrate confidence calibration and test-time scaling

in a unified framework, thereby improving both the reliability and efficiency of inference.

Our contributions are summarized as follows:

Correctness and Confidence Calibration Reinforcement Learning(C3RL): We propose C3RL, a novel RL algorithm that simultaneously improves correctness and calibration, preserving overall performance while reducing overconfidence.

Confidence-based Adaptive Test Time Scaling(CAS): We introduce an early-stopping mechanism for test time scaling that leverages calibrated confidence to dynamically allocate inference budget. This approach dedicates more computation to low-confidence ones, striking an improved efficiency–accuracy balance.

Unified Framework: Together, C3RL and CAS form a coherent pipeline that enables LLMs to express calibrated confidence and to adaptively control inference cost, paving the way for more trustworthy and efficient LLMs.

2 Related Work

Recent studies reveal that, LLMs often exhibit overconfidence (Xiong et al., 2023; Mei et al., 2025) and hallucination (Zhang et al., 2025; Liang et al., 2024), which undermines their reliability. Existing efforts to improve confidence estimation and calibration can be broadly grouped into two major lines: (i) prompt and sampling-based calibration and (ii) training and RL-based calibration. We further discuss related work on adaptive test time scaling.

2.1 Prompt and Sampling based Calibration

Prompt-based approaches guide LLMs to explicitly verbalize their confidence or self-assess their reliability after producing an answer. (Tian et al., 2023; Xiong et al., 2023; Yang et al., 2024a) show that verbalized confidence is often poorly calibrated though calibration improves with model scale and reasoning capability. More recent reasoning-oriented studies (Mei et al., 2025; Yoon et al., 2025) demonstrate that introspective or "slow thinking" chain-of-thought (CoT) prompting can yield better alignment between confidence and correctness. However, these prompting-based techniques depend heavily on prompt wording, limiting robustness and scalability (Zhao et al., 2021).

Another line of research estimates confidence implicitly from response consistency. These sampling-based approaches rely on response diversity or majority voting (Kuhn et al., 2023; Aichberger et al., 2025; Kang et al., 2025), where agreement across multiple generations serves as uncertainty. While these methods correlate well with empirical accuracy, they are computationally expensive.

2.2 Training and RL based Calibration

Training-based methods aim to learn calibrated confidence distributions through supervised or reinforcement objectives. Lin et al. (2022) introduces dataset-level supervision by correlating response types with empirical accuracy, while R-Tuning (Zhang et al., 2023) augments supervised fine-tuning data with binary "I am sure/unsure" labels. Although effective for coarse uncertainty, such designs fail to capture fine-grained, context-dependent confidence.

More principled formulations employ reinforcement learning (RL) to jointly optimize correctness and calibration. *SaySelf* (Xu et al., 2024) and Stangel et al. (2025) train LMs with proper scoring rules (e.g., Brier or log loss) as rewards, directly aligning verbalized confidence with output correctness. However, these objectives often prioritize calibration alone, potentially encouraging low-confidence incorrect answers to minimize loss. RLCR (Damani et al., 2025) incorporates calibrated reward functions to balance accuracy and calibration, though applications remain limited to question-answering or Maths tasks. Instead, our model trained with C3RL on reasoning-intensive tasks maintains accuracy and reaches optimal improvement in calibration.

2.3 Adaptive Test Time Scaling

Adaptive sampling has emerged as a powerful approach to reduce the computational cost of test time scaling. Adaptive-Consistency (Aggarwal et al., 2023) introduces a model-agnostic technique that dynamically adjusts the number of samples per question using lightweight stopping criteria based on the frequency of answers. Li et al. (2023) takes an alternative approach by dividing the sampling process into sequential windows and stopping when all samples within a window are the same. However, these adaptive sampling methods rely on frequency or entropy to approximate confidence, failing to utilize the confidence signal itself directly.

3 Method

In this section, we introduce the construction process of training data (Section 3.1), the reward de-

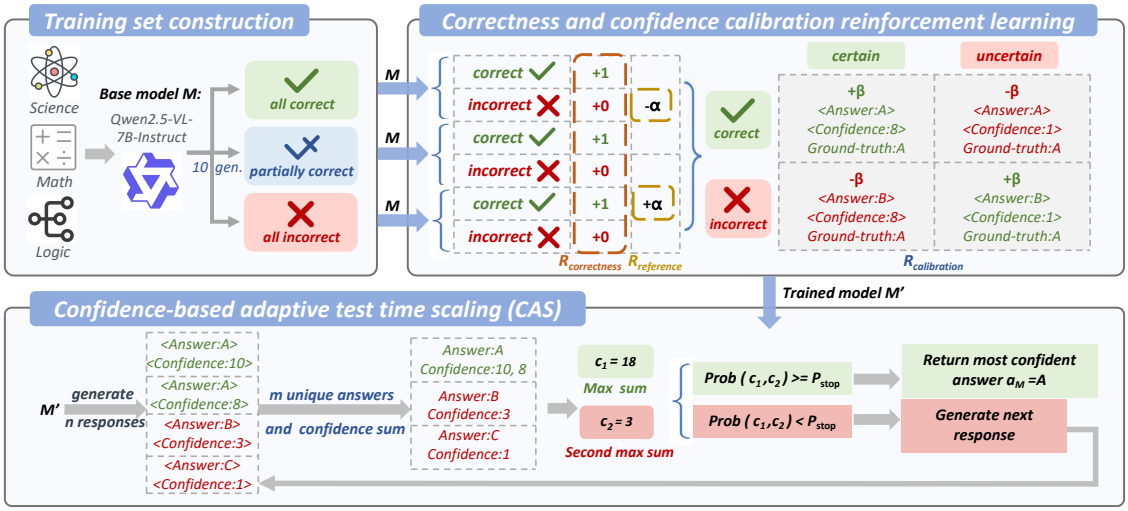


Figure 1: Pipeline for the framework of training set construction, Correctness and Confidence Calibration Reinforcement Learning(C3RL) and Confidence-based Adaptive Test Time Scaling(CAS). The base model first generates 10 responses for each sample and categorizes the samples into "all correct", "partially correct" and "all incorrect". In C3RL, $R_{\text{correctness}}$ encourages correct answers, $R_{\text{reference}}$ rewards/penalizes correctness change based on the assigned categories, and $R_{\text{calibration}}$ incentivizes calibration of confidence. In CAS, the stopping probability (Eq. 1) is calculated from the most confident answer and the second most confident answer. If the most confident answer dominates existing generations (c_1 is much larger than c_2), the inference will stop to save budget.

sign for C3RL (Section 3.2) and the adaptive test time scaling strategy based on calibrated confidence (Section 3.3). Figure 1 shows the pipeline of the whole framework.

3.1 Training Dataset Construction

Training LLMs via Reinforcement Learning requires comprehensive and moderately difficult datasets. We create our high-quality training datasets and assign an accuracy tag to each data.

We first collect datasets in mathematics, science and logic reasoning. The selected datasets are denoted as $\mathcal{D}$, with the following components: (1) **Mathematics and Science:** NuminaMath-TIR (Jia LI and Polu, 2024) (69k numerical-answer math problems targeting computation and reasoning) and WebInstruct-verified (Ma et al., 2025) (4.5k selected numerical science problems with integer answers); (2) **Logical Reasoning:** Logic-NLI (Tian et al., 2021) (16k problems unbinding the target first-order logic reasoning from commonsense inference) and LogiQA (Liu et al., 2020) (8,678 QA problems focusing on deductive reasoning). The detailed filtering process is in A.1.1. $\mathcal{D}$ is randomly split into $\mathcal{D}_{train}$ (106k samples, 99%) and $\mathcal{D}_{test}$ (1,073 samples, 1%).

For each question q in $\mathcal{D}_{train}$, we use the base model (Qwen2.5VL-7B-Instruct (Team, 2025) or Llama-3.2-3B-Instruct (Meta, 2024)) to generate 10 answers with a decoding temperature 0.7. The answer generation prompt is in A.2.1. We calculate the accuracy of this answer set: $Acc_q = \sum_{i=1}^{10} \mathbb{I}(a_i = a^*)/10$, where $\mathbb{I} \in \{0, 1\}$ is an indicator function and a^* is the ground truth. Then we assign a reference accuracy tag T_q from the following three options: "all correct" (Acc_q = 1), "partially correct" (0<Acc_q <1) and "all incorrect" (Acc_q = 0).

To be specific, D_{train} composes of 13.7k "all correct" questions, 68.1k "partially correct" questions, and 24.3k "all incorrect" questions. In the following section, we incorporate this reference accuracy into our reward design.

3.2 C3RL

Previous RL training methods for calibration simply incentivize calibration and pose potential risks that models learn to give incorrect answers deliberately with low confidence. To prevent such a "calibrated but wrong" shortcut, we design a novel Correctness and Confidence Calibration Reinforcement Learning (**C3RL**). For each question $q \in \mathcal{D}_{train}$, we prompt the base model to generate an answer a_q with an associated verbalized confidence $c_{a_q} \in \{1, 2, ..., 10\}$, where $c_{a_q} = 1$ means the model is totally uncertain and $c_{a_q} = 10$ means the model is totally certain. We train the model (same as the base model) utilizing the following

reward:

$$R_{\text{C3RL}}(q, a_q, c_{a_q}, T_q) = R_{\text{correctness}} + R_{\text{calibration}} + R_{\text{reference}}$$

The three parts in the reward are as follows:

Correctness Reward: The traditional RLVR (Lambert et al., 2024) reward $R_{\text{correctness}}(a_q) = \mathbb{I}(a_q = a^*) \in \{0, 1\}$. This part ensures the model will update in the "correct" direction instead of "calibrated but wrong" way.

Calibration Reward: We set a confidence threshold t=5 to classify the answer a_q into "certain" ($c_{a_q} > t$) and "uncertain" ($c_{a_q} \leq t$) categories. We encourage "certain correct" and "uncertain incorrect" answers with $R_{\text{calibration}} = +\beta$ while penalizing "certain incorrect" and "uncertain correct" responses with $R_{\text{calibration}} = -\beta$. The ablation process to determine β is detailed in A.5.

Reference Accuracy Reward: To utilize the prior accuracy information in data construction process, we customize a reward for question q based on reference accuracy tag T_q. When the model answers correctly from formerly "all incorrect" questions, we incentivize it with $R_{\text{reference}} = +\alpha$. On the contrary, if the model hallucinates with incorrect answers from formerly "all correct" questions, we punish it with $R_{\text{reference}} = -\alpha$. This design forces the model to remain correct if it can originally give correct answers and further stops the model from falling into "calibrated but wrong" risks. The ablation process to select α is elaborated in A.5.

3.3 Confidence-based Adaptive Test Time Scaling (CAS)

Prior adaptive self-consistency methods typically rely on frequency or entropy to determine early-stopping criteria. However, such signals are often uncalibrated and fail to accurately reflect the model's true confidence. To address this limitation, we propose a Confidence-based Adaptive test-time Scaling (CAS) strategy, which integrates calibrated verbalized confidence into the stopping criterion. After generating each batch of responses, CAS applies a well-calibrated stopping criterion based on the verbalized confidence of model outputs to decide whether to (1) continue sampling or; (2) halt and return the current highest confidence-weighted answer.

Inspired by the idea that a high-confidence answer which is dominant over all the other responses suggests a reliable stopping criteria, we design CAS as follows:

For a question q to be sampled N times, there have been $n < N$ generated responses $\{r_1, r_2, ..., r_n\}$ with m ($m \leq n$) unique extracted answers $A_q = \{a_1, a_2, ..., a_m\}$, and for each unique answer $a_i \in A_q$, assume it corresponds to j responses $R_{a_i} = \{r_{i_1}, r_{i_2}, ...r_{i_j}\}$. Then we calculate the sum of verbalized confidence for a_i: $c_{a_i} = \frac{1}{10}\sum_{k=1}^{j} \text{conf}(r_{i_k})$, where $\text{conf}(r) \in \{1, 2, ..., 10\}$ extracts the verbalized confidence of response r. The sums of verbalized confidence for all unique answers in A_q form a confidence set $C_q = \{c_{a_1}, c_{a_2}, ..., c_{a_m}\}$, and we can select the maximum and second maximum of C_q: $c_1 = \max(C_q), c_2 = \max(C_q \backslash \{c_1\})$. Intuitively, c_1 corresponds to the "most confident" answer, and c_2 corresponds to the "second most confident" answer. Following the BETA-distribution stopping criteria described in Adaptive-Consistency (Aggarwal et al., 2023), our confidence-based stopping probability can be calculated as:

$$\text{Prob}(c_1, c_2) = \frac{1}{A}\int_0^{0.5} x^{c_2}(1-x)^{c_1}\, dx, \quad (1)$$

where coefficient $A = \int_0^1 x^{c_2}(1-x)^{c_1}\, dx$ normalizes the integral on the interval [0,1] to 1. We set a probability threshold P_{stop} to decide whether to stop sampling. If $\text{Prob}(c_1, c_2) \geq P_{\text{stop}}$ or n reaches the maximum generation time N, the sampling process ends; otherwise, the inference continues with next generation. When c_1 is larger compared to c_2 (which means the most confident answer dominates the generations), the probability to stop is higher.

The detailed algorithm is described in Algorithm 1. We also design a modified equivalent Algorithm 2 to support parallel inference instead of generating responses one by one. The detail of Algorithm 2 is elaborated in A.6.

4 Experiments

The main experiments aim to address the following three research questions: (1) **how C3RL balances accuracy and calibration** (Sec 4.1); (2) **how CAS saves inference budget in test times scaling** (Sec 4.2) and (3) **how C3RL contributes to CAS performance** (Sec 4.3).

Algorithm 1 Confidence-based Adaptive Test Time Scaling (CAS)

Require: Question q, max sampling time N, threshold P_{stop}
Initialize $n \leftarrow 0$, $A_q \leftarrow \emptyset$, $C_q \leftarrow \emptyset$
while $n < N$ **and** stopping condition not met **do**
  Generate 1 additional sample, $n \leftarrow n + 1$
  Update answer set A_q, confidence set C_q
  $c_1, c_2 \leftarrow$ max and second max of C_q
  Compute $\text{Prob} = \frac{1}{A} \int_0^{0.5} x^{c_2}(1-x)^{c_1} dx$
  if $\text{Prob} \geq P_{\text{stop}}$ **then break**
  end if
end while
return a_M (most confident answer)

4.1 Evaluation of C3RL

4.1.1 Benchmarks

The benchmarks include text and multimodal datasets across knowledge-intensive tasks: (1) **Text:** AGIEval (Zhong et al., 2023) and MMLU test set (Hendrycks et al., 2021). (2) **Multimodal:** MMMU (Yue et al., 2024), MathVista (Lu et al., 2024) and LogicVista (Xiao et al., 2024). The details of the benchmarks are in A.1.2.

4.1.2 Evaluation metrics

We use the following metrics to test LLMs' ability to give correct and calibrated responses:

Accuracy (↑): To measure the correctness of LLMs on the evaluation set: $\text{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(a_{q_i})$

AUROC (↑): Area Under ROC Curve measures the ability to distinguish between correct and incorrect answers under different confidence thresholds:

$$\text{AUROC} = \int_1^{10} \text{TPR}(\text{FPR}(t)^{-1})\, d(t), \quad (2)$$

where $t \in [1, 10]$ is confidence threshold, TPR is true positive rate, and FPR is false positive rate.

ECE (↓): We choose Binned Expected Calibration Error to describe the gap between verbalized confidence and accuracy:

$$\text{ECE} = \frac{m}{N} \sum_{i=1}^{\lceil \frac{N}{m} \rceil} |\text{Acc}(B_i) - \text{Conf}(B_i)|, \quad (3)$$

where N is the number of samples, $m = 20$ is the fixed bin size , B_i is the set of data in the i-th bin.

4.1.3 Baselines

We compare the following methods:

Non-RL Methods (1) Base: We directly ask the untrained model to output both answer and associated confidence ; (2) Self Consistency(SC) (Wang et al., 2022) : Each question is sampled 10 times and the final answer is obtained from majority voting; (3) Supervised Finetuning with reference accuracy(SFT+Ref) (Yang et al., 2024b): The model is supervised fine-tuned to calibrate with reference accuracy calculated in 10-time sampling.

RL Methods (1) RLVR: The off-the-shelf method using the correctness reward ; (2) SaySelf (Xu et al., 2024) :The reward simply contains calibration Brier score; (3) RLCR (Damani et al., 2025): Based on SaySelf, RLCR adds the correctness score to ensure the performance, which is current state-of-the-art RL method to tradeoff between performance and calibration; (4) C3RL (Ours) and variants: C3RL integrates the correctness reward, calibration reward and reference accuracy reward together. We exclude the reference accuracy reward (C3RL w/o Ref) to conduct ablation study.

4.1.4 Results: Calibration-Accuracy Trade-off

Table 1 shows the comparison of different methods in both accuracy and calibration metrics.

In-domain Performance As shown in Table 1, all calibration-aware reward methods consistently outperform the basic RLVR approach in both AUROC and ECE on the in-domain test set $\mathcal{D}_{test}$. Notably, Qwen2.5VL-7B-Instruct trained after C3RL achieves highly competitive performance, closely matching RLVR in accuracy while attaining the similar superior ECE (0.076) as the best-calibrated method, SaySelf. This demonstrates C3RL's strong ability to maintain high accuracy alongside excellent calibration, even though it does not explicitly surpass the current state-of-the-art RLCR in performance.

Generalization As shown in Table 1 and Table 2, C3RL exhibits remarkable generalization capabilities on out-of-domain(OOD) datasets. Compared to other top-calibrated RL methods, Qwen2.5VL-7B-Instruct trained after C3RL is superior to SaySelf—the best RL strategy in calibration—in ECE metric and surpasses it greatly in accuracy (7.9% in text sets and 19.2% in multimodal sets). Moreover, C3RL consistently outperforms the current state-of-the-art RLCR across both accuracy and calibration metrics on OOD evaluations, demonstrating an optimal and robust balance between performance and reliability. When generalized to a different family model Llama-3.2-

Text	In-domain			Agieval			MMLU			Avg.		
Method	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓
Qwen2.5VL-7B-Instruct												
Base	45.3%	0.531	0.448	51.0%	0.543	0.375	68.3%	0.587	0.188	54.9%	0.554	0.337
SC	53.9%	0.762	0.088	53.7%	0.711	0.208	69.9%	0.759	0.141	59.2%	0.744	0.146
SFT+Ref	48.0%	0.720	0.098	49.1%	0.753	0.105	62.1%	0.706	0.098	53.1%	0.726	0.100
RLVR	**66.6%**	0.514	0.285	**58.1%**	0.534	0.377	71.1%	0.602	0.239	**65.3%**	0.550	0.300
SaySelf	55.0%	**0.807**	**0.075**	50.8%	**0.745**	**0.147**	64.0%	**0.731**	0.129	56.6%	**0.761**	0.117
RLCR	65.1%	0.751	0.109	57.5%	0.664	0.205	70.7%	0.607	0.175	64.4%	0.674	0.163
C3RL w/o Ref	64.9%	0.700	0.119	**58.1%**	0.682	0.197	71.2%	0.648	0.111	64.7%	0.677	0.142
C3RL (Ours)	64.4%	0.748	0.076	57.8%	0.688	0.154	**71.4%**	0.653	**0.092**	64.5%	0.696	**0.107**
Llama-3.2-3B-Instruct												
Base	37.2%	0.563	0.134	34.8%	0.557	0.423	59.6%	0.586	0.271	43.9%	0.569	0.276
SC	40.8%	0.787	0.091	28.6%	0.644	0.352	59.9%	0.733	0.158	43.1%	0.721	0.200
SFT+Ref	37.6%	0.572	0.168	32.1%	0.594	0.240	52.8%	0.628	0.162	40.8%	0.598	0.190
RLVR	45.9%	0.587	0.463	**37.3%**	0.522	0.468	63.4%	0.537	0.208	48.9%	0.549	0.380
SaySelf	13.1%	0.676	**0.060**	19.4%	0.611	**0.072**	37.4%	**0.715**	0.110	23.3%	0.667	**0.081**
RLCR	43.3%	**0.799**	0.072	34.4%	0.658	0.111	62.3%	0.670	0.153	46.7%	**0.709**	0.112
C3RL w/o Ref	47.3%	0.697	0.136	37.2%	0.652	0.218	63.5%	0.688	**0.087**	49.3%	0.679	0.147
C3RL (Ours)	**49.1%**	0.734	0.142	**37.3%**	**0.660**	0.246	**64.0%**	0.680	0.089	**50.1%**	0.691	0.159

Table 1: Qwen2.5VL-7B-Instruct and Llama-3.2-3B-Instruct results across in-domain test set D_{test} , Agieval, and MMLU benchmarks. Among RL methods, the best values are highlighted in bold, and the suboptimal values are underlined. Our proposed C3RL method is shaded. All the methods except for Self Consistency are tested three times with seed=0,42,2025 separately and the results are averaged. The maximum standard deviation among all results is $\pm 0.9\%$, which is in a reasonable range.

3B-Instruct (Meta, 2024), although achieving better calibration, RLCR sacrifices more performance and can't match RLVR in accuracy, which means there are more deliberately low-confidence incorrect answers. On the contrary, C3RL can still surpass RLVR in performance and decrease calibration error greatly at the same time, proving the stability of C3RL to different model families and sizes.

Ablation Study When the reference accuracy reward is ablated (C3RL w/o Ref), the model remains competitive accuracy but suffers a consistent degradation in calibration. This clearly displays the specific contribution of the reference accuracy reward to enhancing calibration capability.

4.1.5 Analysis: C3RL Advantage

The advantage of C3RL over RLCR in balancing performance and calibration is due to the following two reasons: (1) Reference accuracy information: The ablation study has demonstrated the effectiveness of reference accuracy reward in advancing calibration; (2) Importance of learning confidence threshold: RLCR, driven by the Brier Score (see A.2.3), tends to train LLMs to indicate totally confident (c=10) or totally inconfident (c=1) responses, which degrades to a binary case. On the contrary, C3RL guides the model to learn the confidence threshold t=5 and gives fine-grained confidence to distinguish between certainty and uncertainty. The difference in confidence distribution between C3RL and RLCR is demonstrated in A.4.

4.2 Can CAS Improve Test Time Scaling?

Given the impressive results of self-consistency in Table 1, it is natural to combine verbalized confidence and self-consistency together. Although there have been test time scaling methods leveraging confidence to enhance accuracy (e.g. confidence-weighted majority voting), they fail to apply calibrated confidence to saving inference budget. We conduct experiments to demonstrate the inference efficiency of Confidence-based Adaptive Test Time Scaling (CAS).

4.2.1 Experimental Setup for CAS

To evaluate the performance and inference efficiency of CAS, we employ the following settings:

Benchmarks We select 6 benchmarks of reasonable sizes (less than 2000 samples): (1) **Text**: MMLU validation set (Hendrycks et al., 2021), FOLIO (Han et al., 2022) and GSM8K (Cobbe et al., 2021); (2) **Multimodal** : The same as CAS evaluation multimodal sets. The details of the benchmarks are in A.1.2.

Baselines

(1) **Majority Voting (MV)** (Wang et al., 2022):

Method	MMMU			MathVista			LogicVista			Avg.		
	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓
Qwen2.5VL-7B-Instruct												
Base	0.473	0.571	0.381	0.543	0.623	0.284	0.368	0.541	0.493	0.461	0.578	0.386
SC	0.509	0.621	0.211	0.591	0.818	0.077	0.446	0.678	0.157	0.515	0.706	0.148
SFT+Ref	0.368	0.651	0.095	0.452	0.702	0.091	0.306	0.578	0.092	0.375	0.644	0.093
RLVR	0.520	0.525	0.417	0.614	0.595	0.304	**0.467**	0.522	0.495	0.534	0.547	0.405
SaySelf	0.331	**0.784**	0.160	0.431	**0.822**	0.094	0.261	**0.739**	**0.152**	0.341	**0.782**	0.135
RLCR	0.509	0.651	0.146	0.600	0.690	0.117	0.431	0.608	0.284	0.513	0.650	0.182
C3RL w/o Ref	0.524	0.613	0.173	**0.622**	0.669	0.139	0.462	0.591	0.268	**0.536**	0.624	0.193
C3RL (Ours)	**0.530**	0.655	**0.096**	**0.622**	0.682	**0.074**	0.446	0.596	0.158	0.533	0.644	**0.109**

Table 2: Qwen2.5VL-7B-Instruct results across multimodal reasoning benchmarks MMMU, MathVista and LogicVista. Among RL methods, the best values are highlighted in bold, and the suboptimal values are underlined. Our proposed C3RL method is shaded and achieves superior calibration performance with the lowest ECE scores on MMStar and MathVista. All the methods except for Self Consistency are tested three times with seed=0,42,2025 separately and the results are averaged. The maximum standard deviation among all results is $\pm 1.0\%$, which is in a reasonable range.

Method	In-domain			OOD (Text)			OOD (Multimodal)		
	Avg. N↓	Saved Times↑	Acc↑	Avg. N↓	Saved Times↑	Acc↑	Avg. N↓	Saved Times↑	Acc↑
MV (Wang et al., 2022)	64	×1	67.3%	64	×1	78.1%	64	×1	57.8%
AC (Aggarwal et al., 2023)	13.71	×4.67	67.4%	12.87	×4.97	78.1%	15.24	×4.20	57.9%
CAS (Ours)	**10.91**	**×5.87**	67.5%	**5.19**	**×12.33**	78.2%	**7.13**	**×8.98**	57.9%

Table 3: Comparative results of **MV** (Majority Voting), **AC** (Adaptive Consistency) and **CAS** (Confidence-based Adaptive Test Time Scaling) on In-domain, OOD (text), and OOD (multimodal). By adjusting the threshold P_{stop} of AC and CAS, we compare their needed inference budget Avg. N to match the accuracy of MV($N = 64$).

Each question is generated N times and the most frequent answer is voted. (2) **Adaptive-Consistency (AC)** (Aggarwal et al., 2023): A strategy dynamically adjusts the number of samples based on a stopping criteria to detect dominant answers. The final answer is voted from the truncated samples.

Settings To evaluate all the methods equally, we compare the inference budget needed to achieve the same accuracy as Majority Voting(N=64). It's guaranteed by controlling threhold P_{stop} for AC and CAS separately. We use the Qwen2.5VL-2.5-7B-Instruct model trained after C3RL.

4.2.2 Results

As shown in Table 3, our proposed CAS method demonstrates remarkable efficiency gains while maintaining competitive accuracy across all evaluation settings.

In terms of sampling efficiency, CAS achieves outstanding cost reduction compared to baseline methods. Notably, CAS consistently outperforms the strongest baseline Adaptive-Consistency. The performance gap is particularly striking on OOD (Text) tasks, where CAS uses less than half of the sampling budget of Adaptive-Consistency (5.19 vs. 12.87 samples).

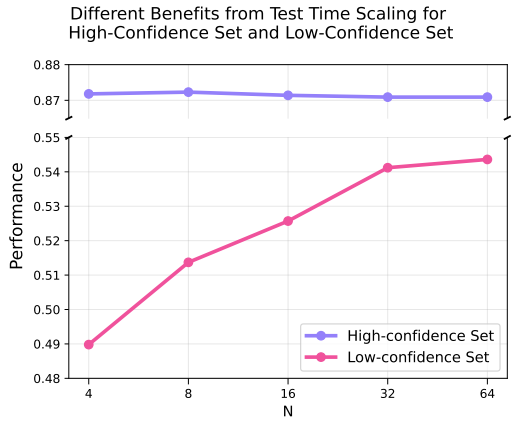


Figure 2: Different benefits from test time scaling for High-confidence set (CAS stopping $n \leq 4$) and Low-confidence set (CAS stopping $n > 4$) on OOD (Text). The 0.55-0.86 range of y-axis is cut for clarity. The High-Confidence set benefits much less from test time scaling than the Low-Confidence set.

These efficiency gains come with no cost of accuracy. This demonstrates that CAS can precisely identify high-quality reasoning paths early in the generation process, eliminating unnecessary computations without sacrificing final performance.

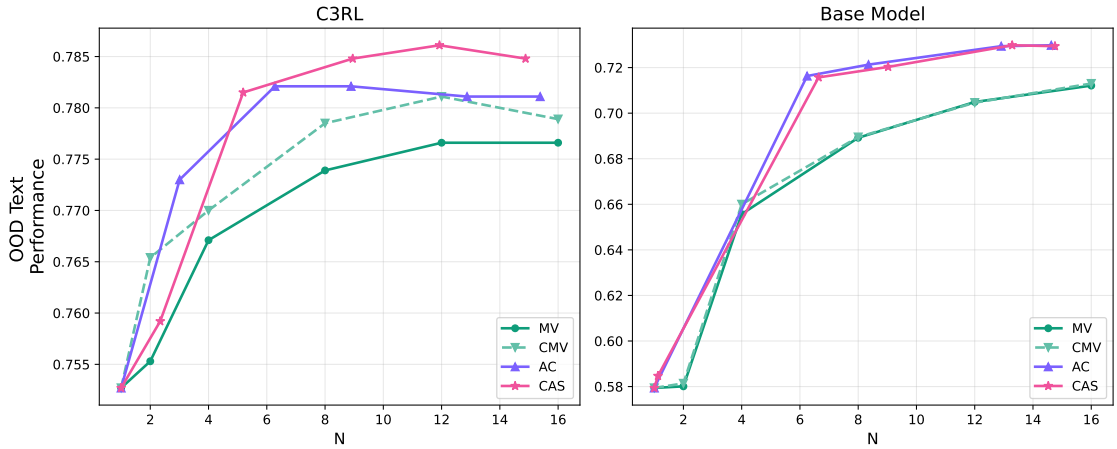


Figure 3: Performance of C3RL model vs. Base model with different test time scaling methods on OOD (Text). **MV** denotes Majority Voting, **CMV** denotes Confidence-weighted Majority Voting, **AC** denotes Adaptive Consistency, **CAS** denotes Confidence-based Adaptive Test Time Scaling. C3RL model shows performance stratification on different methods, while the base model improves little for confidence-based methods.

4.2.3 Analysis: Imbalanced Budget Allocation

CAS adjusts the inference budget from high-confidence sampling traces to low-confidence traces with early-stopping strategy. To verify the rationality of this imbalanced mechanism, we compare the benefits from test time scaling for high-confidence and low-confidence samples.

We use the C3RL model for generation and choose the three text benchmarks in Sec 4.2.1 for evaluation. We divide the samples into two sets: High-confidence Set and Low-confidence Set. For each question, if the CAS sampling time $n \leq 4$ (which means CAS stops early because the final answer reaches high confidence quickly), we categorize this question to High-confidence Set. Otherwise, we assign the sample to Low-confidence Set. We study how these two sets benefit differently from majority voting.

As shown in Figure 2, when test time N increases to 64, the High-confidence Set remains unchanged in performance (<0.2%), while the Low-confidence Set jumps dramatically from 49.0% to 54.5%. The fact that the Low-confidence Set benefits much more from test time scaling than its counterpart rationalizes the CAS strategy of shifting limited computing resource from high-confidence samples to low-confidence samples.

4.3 Can C3RL contribute to CAS?

To explore the influence of calibrated confidence in test time scaling, we compare two models: The base model and the model trained with C3RL. The Confidence-weighted Majority Voting (CMV) method is also evaluated to display the contribution of calibrated confidence.

As shown in Figure 3, C3RL+CAS outperforms all baseline methods when number of samples N is sufficiently large ($\geq$8), demonstrating the scalable effectiveness of our approach. A key observation is the stratification of test-time scaling strategies in the C3RL-trained model. In comparison, the base model benefits little from CAS than AC. This stark contrast underscores that confidence calibration through C3RL is prerequisite for CAS to unlock its full potential. The performance leap from base+CAS to C3RL+CAS demonstrates their synergistic relationship: C3RL provides well-calibrated verbalized confidence, and CAS efficiently leverages it for test-time scaling.

5 Conclusion

This work presents Correctness and Confidence Calibration Reinforcement Learning (C3RL) to support downstream Confidence-based Adaptive test time Scaling (CAS). By integrating correctness, calibration and prior reference information together, C3RL achieves the best trade-off between performance and calibration, providing credible verbalized confidence. Based on well-calibrated confidence, CAS takes inference most efficiently among test time scaling methods by adjusting computing resources to low-confidence samples. Our results highlight a unified pipeline for reliable LLMs and efficient inference under limited resources.

Limitations

There are several limitations for our C3RL-CAS framework. First, the high-quality training data in science domain is not sufficient compared to Maths and logic reasoning domains, which may result in training bias. Second, although C3RL achieves best accuracy-calibration trade-off, its calibration ability is still inferior to SaySelf, which means if we want to enhance the model accuracy, we have to sacrifice calibration. Third, the advantage of CAS over Adaptive-Consistency in small generation cases (N<8) is not considerable, which requires more reasonable stopping criteria and better-calibrated confidence from C3RL.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, and 81 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. *arXiv preprint arXiv:2305.11860*.

Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. 2025. Improving uncertainty estimation through semantically diverse language generation. In *The Thirteenth International Conference on Learning Representations*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, , and Feng Zhao. 2024. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

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.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

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.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, and 7 others. 2022. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Qianxi He, Qingyu Ren, Shanzhe Lei, Xuhong Wang, and Yingchun Wang. 2025. Beyond correctness: Confidence-aware reward modeling for enhancing large language model reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27215–27231, Suzhou, China. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Lewis Tunstall Ben Lipkin Roman Soletskyi Shengyi Costa Huang Kashif Rasul Longhui Yu Albert Jiang Ziju Shen Zihan Qin Bin Dong Li Zhou Yann Fleureau Guillaume Lample Jia LI, Edward Beeching and Stanislas Polu. 2024. Numinamath tir. [https://huggingface.co/AI-MO/NuminaMath-TIR](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).

Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*.

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

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. arXiv:2411.15124.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2023. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *arXiv preprint arXiv:2312.12832*.

Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. 2024. Learning to trust your feelings: Leveraging self-awareness in llms for hallucination mitigation. *arXiv preprint arXiv:2401.15449*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. 2025. General-Reasoner: Advancing llm reasoning across all domains. *arXiv:2505.14652*.

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

Meta. 2024. Llama-3.2-3b-instruct. `https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct`.

Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. 2025. Trust me, i'm wrong: Llms hallucinate with certainty despite knowing the answer. *Preprint*, arXiv:2502.12964.

Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. 2025. Rewarding doubt: A reinforcement learning approach to confidence calibration of large language models. *arXiv e-prints*, pages arXiv–2503.

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.

Qwen Team. 2025. Qwen2.5-vl technical report. *arXiv preprint*, arXiv:2502.13923.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Yuxuan Tong and 1 others. 2024. Hybridflow: Unifying training and inference for large language models via hybrid programming.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint*, arXiv:2203.11171.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. 2024. Logicvista: A multimodal llm logical reasoning benchmark in visual contexts.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

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. *arXiv preprint arXiv:2405.20974*.

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

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2024b. Alignment for honesty. *Preprint*, arXiv:2312.07000.

Xunjian Yin, Baizhou Huang, and Xiaojun Wan. 2023. Alcuna: Large language models meet new knowledge. *Preprint*, arXiv:2310.14820.

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. *arXiv preprint arXiv:2505.14489*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*, 63:67.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Chen Xu, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025. Siren's song in the ai ocean: A survey on hallucination in large language models. pages 1–46.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, Tongshuang Wu, and Jianshu Chen. 2024. Fact-and-reflection (far) improves confidence calibration of large language models. *arXiv preprint arXiv:2402.17124*.

Ziwei Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *Preprint*, arXiv:2304.06364.

A Appendix

A.1 Experimental Data

A.1.1 Training Dataset Construction

We modify the two reasoning datasets Webinstruct-verified (Ma et al., 2025) and Numina-NLI (Jia LI and Polu, 2024) to support regex check of response correctness.

The Webinstruct-verified dataset is modified from the original 231,833 samples. We filter the science data (domains in "physics", "chemistry", "biology", "computer science", "engineering") and select the Integer answer samples. The selected sample number is 9,246.

For Numina-NLI and Webinstruct-verified datasets whose answers are numerical expressions, we check whether the generated samples are "math valid". We exclude the questions with no valid responses among 10 generations. For Logic-NLI (Tian et al., 2021) and LogiQA (Liu et al., 2020) datasets, we follow the selection process specified in C2RM (He et al., 2025).

Finally, $\mathcal{D}$ composes of the following training datasets: Numina-NLI (69,043), Webinstruct-verified (4,488), LogicNLI (16,000) and LogiQA (17,730).

A.1.2 Benchmark Details

We evaluate reasoning and logic capabilities across reasoning, math and logic benchmarks with text and multimodal datasets:

Text: (1) AGIEval (Zhong et al., 2023): 8269 problems from college admission tests, law school admission tests, math competitions, lawyer qualification tests, and national civil service exams. We use the combination of train, validation and test set for evaluation; (2) OlympiadBench (text) (He et al., 2024): Olympiad-level questions including problems from maths and physics competitions. We select the 2,673 text samples for evaluation; (3) MMLU (Hendrycks et al., 2021): A multiple choice benchmark across STEM, the humanities and the social science. The validation set contains 1,531 samples and the test set includes 14,041 samples. For C3RL evaluation, we use valiation set + test set; For CAS evaluation, we only use the validation set; (4) Folio: 223 multiple-choice questions with logical reasoning tasks; (5) GSM8K (Grade School Math 8K) (Cobbe et al., 2021): 8.5K high quality linguistically diverse grade school math word problems, and we only use the 1,319 questions in test set.

Multimodal (1) MathVision (Wang et al., 2024): High-quality maths problems with visual contexts from real math competitions; the test set contains 3,040 samples; (2) MathVista (Lu et al., 2024): 1,000 challenging Maths reasoning questions with visual contexts; (3) LogicVista (Xiao et al., 2024): 448 multimodal questions focusing on logical reasoning tasks (spatial, deductive, inductive, numeric, and mechanical reasoning); (4) MMStar (Chen et al., 2024): 1,500 human-curated challenging samples to test the multimodal abilities in general tasks like fine-grained perception, counting and logic reasoning.

A.2 Training Details

This section complements the details to the training process.

A.2.1 Generation Prompt

Prompt to generate answers

"Answer the question following this exact format: **Strict Requirements**: **Structure**: " "After reasoning, output ONLY the [final answer] within \boxed{{}} format: <think> [your analysis] </think>" "<ANSWER is: \boxed{Your answer}> **No Extra Text**: Do not explain your answer outside<think> tags. Put your answer into <ANSWER is: \boxed{Your answer}>. Based on your answer, please attach a confidence signal ranging from 1-10 to specify whether you are certain about your answer. 1 means you are totally uncertain (strong inconfidence), while 10 means you are totally certain (strong confidence). If you need more information to answer the question, please attach 1. We will compare your answer with the ground truth to check the correctness. If your answer is correct and accompanied by strong confidence, you will be rewarded; if your answer is incorrect but assigned strong confidence, you will be punished. The signal should be in the format of <CONFIDENCE:NUMBER>, where NUMBER ranges from 1 to 10, directly appended to your answer. The last line of your output should be in the format: <ANSWER is: \boxed{Your answer}>|<CONFIDENCE:NUMBER>."

Table 4: Prompt to generate answers and elicit verbalized confidence

In Sec 3.1, we need to generate 10 samples per question. The model answer generation prompt is shown in Table 4. The prompt: (1) restricts the formats of responses so that the answers can be extracted by rule and (2) elicits the verbalized confidence with an elaborated description of certainty and uncertainty.

A.2.2 Training Settings

We set the generation hyperparameters as shown in Table 5. We employ VeRL (Tong et al., 2024) framework and use 8 NVIDIA-A800-SXM4-80GB GPUs to support C3RL training of Qwen2.5-VL-7B-Instruct.

Config	Value
actor-lr	1×10^{-6}
kl_coef	0.001
max_prompt_length	2,048
max_response_length	2,048
train_batch_size	1,024
ppo_mini_batch_size	256
clip_ratio	0.20
sample_temperature	0.7
rollout.n	10
total_training_steps	103

Table 5: Training hyper-parameters during C3RL

A.2.3 Reward Details

For all the RL in the experiments, we first assign a format reward $R_{\text{format}}(a_q)$ to the responses: Following the prompt designed in Table 4, if both answer and confidence can be extracted, we give a format reward $R_{\text{format}}(a_q)$=+0.1; otherwise, the final reward R_{C3RL} will be set to -1 directly.

For C3RL reward, $R_{\text{correctness}}(a_q) = 0.9\mathbb{I}(a_q = a^*)$ so that a correct answer will have $R_{\text{correctness}}(a_q)$ + $R_{\text{format}}(a_q)$ = 1 (equal to RLVR reward).

For SaySelf reward, we use the Brier Score R_{SaySelf}=1-2$(\mathbb{I}(a_q = a^*) - \text{conf}(r)/10)^2$.

For RLCR reward, we increment a correctness reward based on SaySelf reward: $R_{\text{RLCR}} = \mathbb{I}(a_q = a^*) - (\mathbb{I}(a_q = a^*) - \text{conf}(r)/10)^2$.

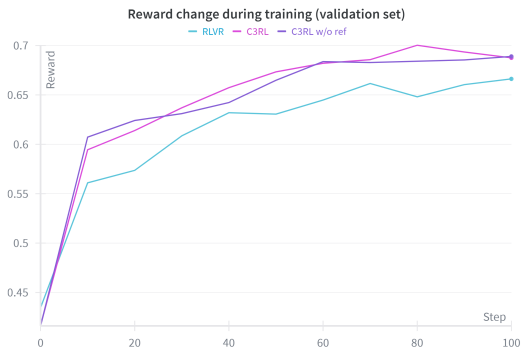
A.2.4 Training Reward Curves

The reward curves and the performance curves are shown in Figure 4. All the methods converge after 100 steps. Figure 4a displays the effectiveness of calibration reward (C3RL w/o Ref curve is higher than RLVR curve) and reference accuracy reward (C3RL curve is higher than C3RL w/o Ref curve). As for performance, 4d shows that all the strategies match at the similar level except for SaySelf, which demonstrates the significance of correctness reward to guarantee accuracy.

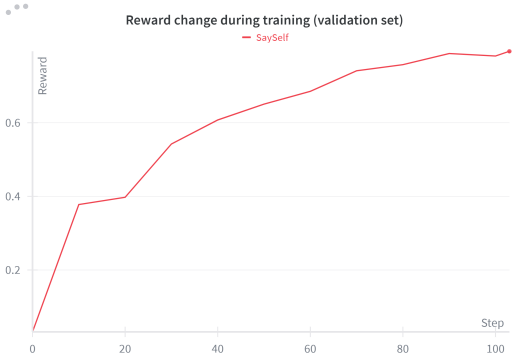
A.3 Evaluation Details

A.3.1 Evaluation Settings

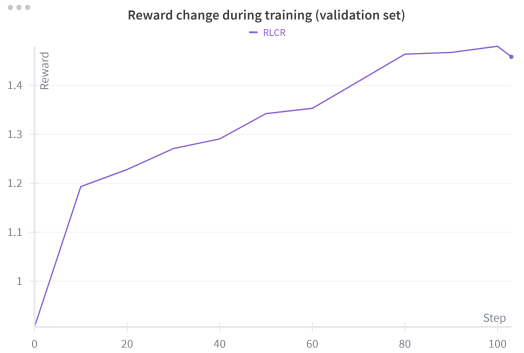
The evaluation hyperparameters keeps exactly the same as the generation hyperparameters in training



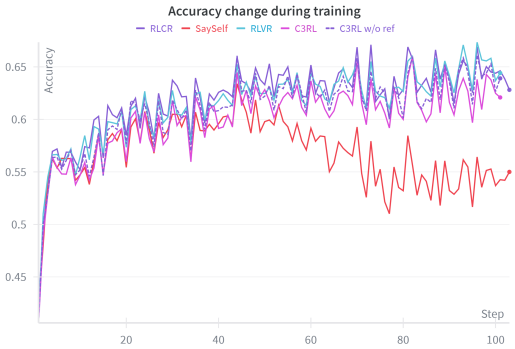
(a) RLVR, C3RL and C3RL w/o ref rewards



(b) SaySelf reward



(c) RLCR reward



(d) Accuracy comparison during training

Figure 4: Training curves

process A.2.2. The evaluation prompts also follow the confidence elicitation prompts in Table 4. We set the decoding temperature 0.7, top_k=-1, top_p=1, and seed=0, 42, 2025 separately for parallel experiments.

A.3.2 Detailed Evaluation Results

The C3RL evaluation of Qwen2.5VL-7B-Instruct (Team, 2025) for all the datasets are elaborated in Table 6 and Table 7. A very interesting phenomenon is that on all the multimodal benchmarks and MMLU, RLVR shows worse calibration metrics than the base model, which echoes the assumption that RLVR can produce overconfident and poorly calibrated models. Moreover, our proposed C3RL algorithm has more outstanding ability on multimodal tasks than on text tasks, which demonstrates that its advantage over other RL methods can be enhanced when generalized to multimodal sets, suggesting the occurrence of general confidence awareness.

A.3.3 Results for Llama-3.2-3B-Instruct

The evaluation of C3RL on Llama-3.2-3B-Instruct (Meta, 2024) for text datasets are detailed in Table 8. Although achieving better calibration, RLCR sacrifices more performance and can't match RLVR in accuracy, which means there are more deliberately low-confidence incorrect answers. On the contrary, C3RL can still surpass RLVR in performance and decrease calibration error greatly at the same time, proving the stability of C3RL to different model families and sizes.

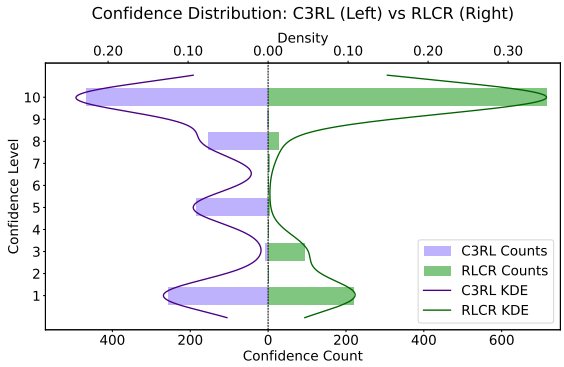


Figure 5: Confidence distribution (Count and Kernel Density Estimate) of C3RL and RLCR on in-domain test set D_{test}. RLCR confidence concentrates on $c = 1$ and $c = 10$, while C3RL confidence is more fine-grained to capture the nuanced uncertainty.

Text	In-domain			Agieval			Olympiad			MMLU		
Method	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓
Qwen2.5VL-7B-Instruct												
Base	45.3%	0.531	0.448	51.0%	0.543	0.375	20.3%	0.594	0.709	68.3%	0.587	0.188
SC	53.9%	0.762	0.088	53.7%	0.711	0.208	26.6%	0.783	0.090	69.9%	0.759	0.141
SFT+Ref	48.0%	0.720	0.098	49.1%	0.753	0.105	10.3%	0.732	0.141	62.1%	0.706	0.098
RLVR	**66.6%**	0.514	0.285	**58.1%**	0.534	0.377	**37.6%**	0.514	0.616	71.1%	0.602	0.239
SaySelf	55.0%	**0.807**	**0.075**	50.8%	**0.745**	**0.147**	28.3%	0.799	**0.071**	64.0%	**0.731**	0.129
RLCR	65.1%	0.751	0.109	57.5%	0.664	0.205	35.4%	**0.805**	0.147	70.7%	0.607	0.175
C3RL w/o Ref	64.9%	0.700	0.119	**58.1%**	0.682	0.197	35.9%	0.745	0.263	71.2%	0.648	0.111
C3RL (Ours)	64.4%	0.748	0.076	57.8%	0.688	0.154	37.5%	0.783	0.191	**71.4%**	0.653	**0.092**
Llama-3.2-3B-Instruct												
Base	37.2%	0.563	0.134	34.8%	0.557	0.423	8.3%	0.559	0.379	59.6%	0.586	0.271
SC	40.8%	0.787	0.091	28.6%	0.644	0.352	8.6%	0.631	0.120	59.9%	0.733	0.158
SFT+Ref	37.6%	0.572	0.168	32.1%	0.594	0.240	5.2%	0.583	0.170	52.8%	0.628	0.162
RLVR	45.9%	0.587	0.463	**37.3%**	0.522	0.468	**17.0%**	0.593	0.754	63.4%	0.537	0.208
SaySelf	13.1%	0.676	**0.060**	19.4%	0.611	**0.072**	3.3%	0.654	0.083	37.4%	**0.715**	0.110
RLCR	43.3%	**0.799**	0.072	34.4%	0.658	0.111	12.8%	**0.657**	**0.077**	62.3%	0.670	0.153
C3RL w/o Ref	47.3%	0.697	0.136	37.2%	0.652	0.218	16.2%	0.656	0.278	63.5%	0.688	**0.087**
C3RL (Ours)	**49.1%**	0.734	0.142	**37.3%**	**0.660**	0.246	15.8%	0.641	0.289	**64.0%**	0.68	0.089

Table 6: Qwen2.5VL-7B-Instruct and Llama-3.2-3B Instruct results across in-domain test set D_{test} , Agieval, Olympiad, and MMLU benchmarks. Among RL methods, the best values are highlighted in bold, and the suboptimal values are underlined. Our proposed C3RL method is shaded. All the methods except for Self Consistency are tested three times with seed=0,42,2025 separately and the results are averaged. The maximum standard deviation among all results is 0.9%, which is in a reasonable range.

A.4 Confidence Distribution Comparison of C3RL and RLCR

As shown in Figure 5, RLCR exhibits a highly polarized confidence distribution concentrated at the extremes ($c = 1$ and $c = 10$), effectively degenerating confidence estimation into a binary indicator. Although Brier-score-based training can align models on the semantic meaning of extreme confidence values (e.g., "confidence=10" denotes total certainty and "confidence=1" denotes total uncertainty), it does not provide a clear confidence boundary separating certain and uncertain cases. As a result, intermediate confidence levels are underutilized.

In contrast, C3RL explicitly introduces a confidence threshold ($t = 5$) through threshold-guided calibration reward, encouraging the model to learn a clearer distinction between certainty and uncertainty while still preserving fine-grained confidence expression. Consequently, C3RL produces a substantially smoother confidence distribution with richer intermediate confidence values. This comparison highlights the importance of learning an explicit confidence threshold instead of relying solely on binary confidence extremes.

A.5 Ablation Study to Determine α and β

We conduct ablation experiments to select optimal α and β. In these experiments, the correctness reward is always included, and the evaluation set is OOD (Text) specified in 4.1.1. We first ablate β by evaluating the models trained with different calibration rewards without reference accuracy reward. We try the β values from [0.05, 0.1, 0.2, 0.3, 0.4] and the result is shown in Figure 6a. We find that different β values share similar performance while $\beta = 0.1$ gives notably drop in ECE, which stands for best calibration.

After fixing $\beta = 0.1$, we then ablate the optimal α value from [0.1, 0.3, 0.5, 0.7, 0.9]. As shown in Figure 6b, $\alpha = 0.5$ reaches best performance and ECE at the same time, which displays its outstanding capability. So we choose $\beta = 0.1$, $\alpha = 0.5$ in our C3RL algorithm.

A.6 Parallel-Inference Version of CAS

Assume there have been $n < N$ generated responses. Based on the stopping mechanism, we

Method	MMMU			MathVision			MathVista			LogicVista		
	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓
Qwen2.5VL-7B-Instruct												
Base	0.473	0.571	0.381	0.242	0.508	0.680	0.543	0.623	0.284	0.368	0.541	0.493
SC	0.509	0.621	0.211	0.282	0.700	0.193	0.591	0.818	0.077	0.446	0.678	0.157
SFT+Ref	0.368	0.651	0.095	0.181	0.613	0.083	0.452	0.702	0.091	0.306	0.578	0.092
RLVR	0.520	0.525	0.417	0.246	0.512	0.738	0.614	0.595	0.304	**0.467**	0.522	0.495
SaySelf	0.331	**0.784**	0.160	0.184	**0.694**	**0.133**	0.431	**0.822**	0.094	0.261	**0.739**	**0.152**
RLCR	0.509	0.651	0.146	0.254	0.626	0.253	0.600	0.690	0.117	0.431	0.608	0.284
C3RL w/o Ref	0.524	0.613	0.173	0.273	0.607	0.382	**0.622**	0.669	0.139	0.462	0.591	0.268
C3RL (Ours)	**0.530**	0.655	**0.096**	**0.279**	0.627	0.290	**0.622**	0.682	**0.074**	0.446	0.596	0.158

Table 7: Qwen2.5VL-7B-Instruct results across multimodal reasoning benchmarks MMMU, MathVision, MathVista and LogicVista. Among RL methods, the best values are highlighted in bold, and the suboptimal values are underlined. Our proposed C3RL method is shaded and achieves superior calibration performance with the lowest ECE scores on MMStar and MathVista. All the methods except for Self Consistency are tested three times with seed=0,42,2025 separately and the results are averaged. The maximum standard deviation among all results is 1.0%, which is in a reasonable range.

Method	In-domain			Agieval			Olympiad			MMLU		
	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓
Base	37.2%	0.563	0.134	34.8%	0.557	0.423	8.3%	0.559	0.379	59.6%	0.586	0.271
SC	40.8%	0.787	0.091	28.6%	0.644	0.352	8.6%	0.631	0.120	59.9%	0.733	0.158
SFT+Ref	37.6%	0.572	0.168	32.1%	0.594	0.240	5.2%	0.583	0.170	52.8%	0.628	0.162
RLVR	45.9%	0.587	0.463	**37.3%**	0.522	0.468	**17.0%**	0.593	0.754	63.4%	0.537	0.208
SaySelf	13.1%	0.676	**0.060**	19.4%	0.611	**0.072**	3.3%	0.654	0.083	37.4%	**0.715**	0.110
RLCR	43.3%	**0.799**	0.072	34.4%	0.658	0.111	12.8%	**0.657**	**0.077**	62.3%	0.670	0.153
C3RL w/o Ref	47.3%	0.697	0.136	37.2%	0.652	0.218	16.2%	0.656	0.278	63.5%	0.688	**0.087**
C3RL (Ours)	**49.1%**	0.734	0.142	**37.3%**	**0.660**	0.246	15.8%	0.641	0.289	**64.0%**	0.68	0.089

Table 8: Llama-3.2-3B-Instruct results across in-domain test set D_{test}, Agieval, Olympiad, and MMLU benchmarks. Among RL methods, the best values are highlighted in bold, and the suboptimal values are underlined. Our proposed C3RL method is shaded.

can find the minimum additional inference time $k \in \{0, 1, ..., N-n\}$ such that the probability will exceed the threshold:

$$\text{Prob}(c_1 + k, c_2) \geq P_{\text{stop}}$$
$$\text{Prob}(c_1 + k - 1, c_2) < P_{\text{stop}}$$

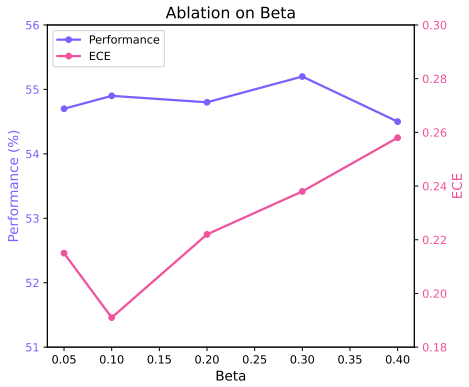
Then k additional responses are generated and n is updated to $n + k$. When current stopping probability $\text{Prob}(c_1, c_2) \geq P_{\text{stop}}$ (i.e. $k = 0$) or n reaches the maximum inference time N, the inference ends and a_M is returned as the final answer. The algorithm is described in Algorithm 2.

A.7 Advantage of Verbalized Confidence

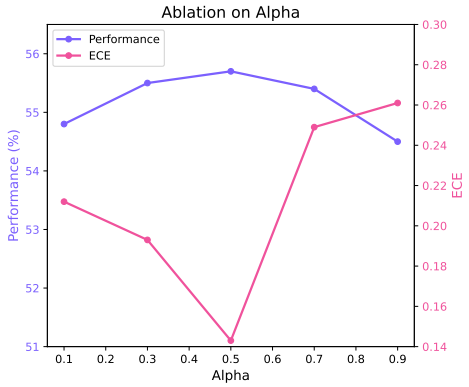
We select verbalized confidence instead of internal logits or probablities to express uncertainty due to the following three reasons. First, verbalized confidence can express epistemic uncertainty of the responses, while the internal logits only represent uncertainty over tokens instead of the claim

Algorithm 2 Confidence-based Adaptive Test Time Scaling (CAS) with Parallel Generation

Require: Question q, max sampling time N, threshold P_{stop}
Initialize $n \leftarrow 0$, $k \leftarrow 1$, $A_q \leftarrow \emptyset$, $C_q \leftarrow \emptyset$
while $n < N$ **and** stopping condition not met **do**
  Generate k additional samples, $n \leftarrow n+k$
  Update answer set A_q, confidence set C_q
  $c_1, c_2 \leftarrow$ max and second max of C_q
  Compute $\text{Prob} = \frac{1}{A}\int_0^{0.5} x^{c_2}(1-x)^{c_1}dx$
  if $\text{Prob} \geq P_{\text{stop}}$ **then break**
  end if
  Find minimum k for stopping criterion
end while
return a_M (answer with highest confidence sum c_1)



(a) Ablation of β



(b) Ablation of α when $\beta = 0.1$

itself (Lin et al., 2022). Second, verbalized confidence itself is typically better-calibrated than conditional probabilities (Tian et al., 2023). Third, we want to explore a model-agnostic method, which should be applicable to all kinds of LLMs without relying on the internal state of LLMs (Yang et al., 2024a).

We test the efficiency of different uncertainty quantification methods in the task of mitigating CHOKE (Certain Hallucinations Overriding Known Evidence) examples (Simhi et al., 2025). As mentioned in (Simhi et al., 2025), a CHOKE example needs to meet three requirements: (1) the LLM can originally answer the question correctly; (2) after modifying with the natural prompt, the LLM hallucinates; and (3) the LLM assigns a high certainty to the wrong answer. We conduct an evaluation experiment on the "all correct" + "partially correct" samples on the in-domain test set (808 among 1,073 samples). We use the model trained with C3RL to generate new responses following the identical settings in A.3.1. The CHOKE examples are those with wrong answers after the prompt is modified (we follow the "prompt_v7" modification in (Simhi et al., 2025)) and assigned a high confidence. We use probability difference as a baseline. The detailed results are shown in Table 9.

Metric	Verb. Conf. + C3RL	Prob. Diff. + C3RL
% of CHOKE	6.1%	22.8%
Avg. Conf.	0.716	0.952

Table 9: Performance comparison of verbalized confidence and probability difference on the task of reducing CHOKE examples.

Table 9 shows the considerable advantage of verbalized confidence over probability difference method in decreasing CHOKE examples from 22.8% to 6.1%. Also, the extremely high average confidence of probability difference demonstrates it suffers from overconfidence, while verbalized confidence gives a relatively uniform confidence distribution.

A.8 Selective-Answering Evaluation

To demonstrate the effectiveness of confidence calibration in calibration-critical scenarios, we additionally evaluate our method on the selective-answering benchmark ALCUNA (Yin et al., 2023). ALCUNA contains both answerable and unanswerable questions. Specifically, the benchmark includes 88 answerable questions, where sufficient information exists to infer the correct answer, and 158 unanswerable questions, where entities are fabricated and no relevant knowledge exists.

For answerable questions, we report the same metrics as in the main experiments, including Accuracy, AUROC, and ECE. For unanswerable questions, we focus on the Abstain Rate, defined as:

$$\text{Abstain Rate} = \frac{\#\text{abstentions}}{\#\text{samples}},$$

where responses with confidence ≤ 5 are regarded as abstentions.

Table 10 reports the evaluation results. As shown in the table, SaySelf achieves the highest abstention rate on unanswerable questions by aggressively suppressing confidence. However, this comes at the cost of severe degradation in both accuracy and calibration on answerable questions, indicating over-abstention even when answering is appropriate.

In contrast, C3RL achieves the best trade-off between accuracy and calibration on answerable samples while still maintaining a strong abstention capability on unanswerable samples. These results

Method	Answerable			Unanswerable
	Acc↑	AUROC↑	ECE↓	Abstain Rate↑
Base	0.977	0.689	0.014	0.247
RLVR	0.966	0.500	0.034	0.234
SaySelf	0.966	**0.859**	0.242	**0.962**
RLCR	**0.987**	0.483	0.015	0.665
C3RL w/o Ref	**0.989**	0.454	0.024	0.595
C3RL (Ours)	0.977	0.727	**0.002**	0.734

Table 10: Evaluation results on the ALCUNA selective-answering benchmark. For answerable questions, we report Accuracy, AUROC, and ECE. For unanswerable questions, we report Abstain Rate, where responses with confidence ≤ 5 are treated as abstentions.

Method	CommonSenseQA			StrategyQA		
	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓
SaySelf	0.725	**0.721**	0.183	0.496	0.589	**0.331**
RLCR	**0.786**	0.604	0.197	0.603	0.588	0.359
C3RL	0.785	0.608	**0.093**	**0.643**	**0.594**	0.356

Table 11: Evaluation results on canonical QA benchmarks CommonSenseQA and StrategyQA.

demonstrate that C3RL does not simply optimize abstention behavior, but instead learns a more reliable confidence boundary that distinguishes when to answer versus abstain.

A.9 Evaluation Results on Canonical QA Benchmarks

To evaluate the generalization capability of C3RL beyond reasoning-intensive tasks, we additionally conduct experiments on two canonical question-answering benchmarks: CommonSenseQA (Talmor et al., 2019) and StrategyQA (Geva et al., 2021). These datasets mainly focus on commonsense reasoning and implicit world knowledge, which are substantially different from our training corpus constructed from mathematics, science, and logical reasoning datasets.

As shown in Table 11, C3RL consistently achieves the best accuracy-calibration trade-off across both benchmarks. On CommonSenseQA, C3RL achieves comparable accuracy to RLCR while significantly reducing calibration error. On StrategyQA, C3RL surpasses all baselines in both accuracy and AUROC. These results demonstrate that the effectiveness of threshold-guided calibration generalizes beyond the reasoning-intensive training distribution and remains beneficial on canonical QA tasks.

Method	AGIEval			MMLU		
	Acc↑	AUROC↑	ECE↓	Acc↑	AUROC↑	ECE↓
SaySelf	0.510	**0.746**	**0.148**	0.642	**0.732**	0.128
RLCR	0.577	0.663	0.203	0.709	0.607	0.174
RLCR w.t.Ref	0.570	0.647	0.214	0.710	0.584	0.190
C3RL w/oRef	**0.580**	0.684	0.198	0.712	0.653	0.111
C3RL (Ours)	0.577	0.694	0.154	**0.716**	0.652	**0.089**

Table 12: Evaluation results on AGIEval and MMLU for RLCR with additional reference accuracy reward (RLCR w.t. Ref). The results show that simply adding the reference reward to RLCR does not recover the advantage of C3RL, demonstrating the effectiveness of threshold-guided calibration.

A.10 Ablation Experiment Evaluating RLCR + Reference Accuracy Reward

We additionally conduct an ablation experiment by incorporating the reference accuracy reward into RLCR, denoted as **RLCR w.t. Ref**. This experiment aims to verify whether the improvement of C3RL mainly comes from introducing additional reward terms or from the threshold-guided calibration formulation itself.

Table 12 reports the evaluation results on AGIEval and MMLU. Compared with RLCR, directly adding the reference accuracy reward to RLCR does not improve calibration performance and even slightly degrades AUROC and ECE on both benchmarks. In contrast, C3RL w/o Ref consistently outperforms RLCR and RLCR w.t. ref in calibration metrics, while the full C3RL achieves the best overall accuracy-calibration trade-off.

These results demonstrate that the improvement of C3RL does not simply arise from adding extra reward signals. Instead, the threshold-guided calibration objective plays the dominant role in preventing confidence collapse and enabling more fine-grained uncertainty estimation.

A.11 AI Using Statement

We employ copilot in writing code for experiments and drawing graphs. We guarantee that all the materials have been checked carefully manually.

A.12 Case Presentation

We present a case from MMLU anatomy questions in Table 13. All the responses are incorrect, while only the model trained by C3RL signals a low confidence, showing the effectiveness of C3RL in calibration.

Cases of different RL methods in calibration

Query: The muscles of the soft palate are innervated by branches of the A) trigeminal and glossopharyngeal nerves. B) facial and glossopharyngeal nerves. C) trigeminal and vagus nerves. D) facial and vagus nerves.
Ground Truth: C.

C3RL Model response: To determine the correct answer, let's analyze the function and innervation of the muscles of the soft palate.1. The soft palate is a part of the oral cavity that helps in various functions such as swallowing, speech, and maintaining airway closure during breathing.2. The muscles of the soft palate are innervated by branches of specific cranial nerves., let's consider each option:- **A) trigeminal and glossopharyngeal nerves**: The trigeminal nerve (CN V) innervates the sensory receptors in the mouth and throat, but it does not innervate the muscles of the soft palate. The glossopharyngeal nerve (CN IX) does innervate the muscles of the soft palate.- **B) facial and glossopharyngeal nerves**: The facial nerve (CN VII) innervates the muscles of the face but not the soft palate. The glossopharyngeal nerve (CN IX) does innervate the muscles of the soft palate.- **C) trigeminal and vagus nerves**: The trigeminal nerve (CN V) does not innervate the muscles of the soft palate. The vagus nerve (CN X) does innervate the muscles of the soft palate.- **D) facial and vagus nerves**: The facial nerve (CN VII) does not innervate the muscles of the soft palate. The vagus nerve (CN X) does innervate the muscles of the soft palate.the above analysis, the correct innervation for the muscles of the soft palate is by the glossopharyngeal nerve (CN IX). <ANSWER is: \boxed{A}>|<CONFIDENCE:1> .

Base model response: To determine which nerves innervate the muscles of the soft palate, we need to consider the anatomical innervation of this area.1. **Trigeminal Nerve (CN V)**: This nerve provides sensory function to the face and oral cavity. The branch involved in the soft palate is the greater petrosal nerve, which connects to the facial nerve (CN VII).2. **Glossopharyngeal Nerve (CN IX)**: This nerve provides both sensory and motor functions. For the soft palate, it innervates the tensor veli palatini muscle, which is essential for palate movement during swallowing.3. **Facial Nerve (CN VII)**: This nerve provides motor function to the muscles of facial expression and also innervates the tensor veli palatini muscle via the greater petrosal nerve.4. **Vagus Nerve (CN X)**: This nerve has extensive distribution throughout the body and does not specifically innervate the soft palate muscles.the above points, the muscles of the soft palate are primarily innervated by the glossopharyngeal nerve and the facial nerve, through their connection with the greater petrosal nerve. <ANSWER is: \boxed{B}>|<CONFIDENCE:8> .

RLVR Model response: To determine the correct answer, we need to identify which nerves innervate the muscles of the soft palate. The soft palate is part of the pharynx and is involved in various functions such as swallowing and speech.1. **Trigeminal Nerve (CN V)**: This nerve innervates the sensation of the skin on the face and the muscles of mastication. It does not innervate the muscles of the soft palate.2. **Glossopharyngeal Nerve (CN IX)**: This nerve innervates the muscles of the soft palate, the pharynx, and the tonsils. It also carries taste fibers from the posterior third of the tongue.3. **Facial Nerve (CN VII)**: This nerve innervates the muscles of facial expression but does not innervate the muscles of the soft palate.4. **Vagus Nerve (CN X)**: This nerve innervates the larynx, the heart, and the muscles of the digestive tract. It does not innervate the muscles of the soft palate.this information, the muscles of the soft palate are innervated by branches of the glossopharyngeal nerve (CN IX) and the vagus nerve (CN X)., the correct choice that matches the options provided is:) facial and glossopharyngeal nerves.since the correct choice should be B, the answer is: <ANSWER is: \boxed{B}>|<CONFIDENCE:8> .

SaySelf Model response: <think> The muscles of the soft palate are innervated by branches of the trigeminal nerve (CN V) and the glossopharyngeal nerve (CN IX). The trigeminal nerve innervates the anterior two-thirds of the tongue and the soft palate, while the glossopharyngeal nerve innervates the posterior one-third of the tongue and the soft palate. Therefore, the correct answer is A.</think> <ANSWER is: \boxed{A}>|<CONFIDENCE:10> .

RLCR Model response: <think> The muscles of the soft palate are innervated by branches of the trigeminal and glossopharyngeal nerves. The trigeminal nerve (CN V) innervates the musculature of the soft palate through the greater palatine nerve, and the glossopharyngeal nerve (CN IX) innervates it through the lesser palatine nerves. Therefore, the correct answer is A.</think> <ANSWER is: \boxed{A}>|<CONFIDENCE:10> .

Table 13: Responses of C3RL model, Base model, RLVR model, SaySelf model and RLCR model on a MMLU question