ReCalibrate: RL for Uncertainty-Aware Reasoning in LLMs

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) have been observed to perform well when trained to output textual reasoning chains using reinforcement learning (RL). However, almost all successful applications of RL for reasoning use reward functions that are simply binary correctness checks. Because these rewards do not penalize guessing or low-confidence outputs, they often degrade calibration and increase hallucinations as side effects. We propose a new RL framework that jointly improves accuracy and calibrated confidence estimation by combining the correctness reward with the Brier score, a proper scoring rule that incentivizes truthful confidence reporting. Our designed reward provably encourages models to produce predictions that are both accurate and well calibrated. Across a variety of datasets, both in-domain and out-of-domain, our method dramatically improves calibration at no cost in accuracy, outperforming both RL training and classifiers trained *only* to assign confidence scores. While ordinary RL hurts calibration, our approach improves it. These results highlight the potential of calibrated RL for building more reliable and interpretable reasoning models.

1 Introduction

Reasoning models trained with reinforcement learning (RL) have remarkably advanced model capabilities such as math and coding (Guo et al., 2025). A key driver of this progress is the ability to scale thinking time—that is, generating longer chains-of-thought (CoT). Model performance consistently improves as the reasoning traces grow longer and more structured (Muennighoff et al., 2025; Guo et al., 2025).

The standard approach to training reasoning models uses RL with a simple binary correctness reward: $r_{\text{correctness}} = \textit{verify}(y, y_{\text{gold}})$, where *verify* checks whether the model's output y matches the ground-truth answer y_{gold}. While simple and incredibly effective for improving accuracy, this reward comes with a critical limitation: it incentivizes models to guess. Models are rewarded identically whether they are confidently correct or output a random response—and penalized the same whether they guess incorrectly or abstain when unsure.

Consistent with this concern, recent studies have found that reasoning models often exhibit worsened calibration and increased hallucination rates compared to base models (Kirichenko et al., 2025; Jaech et al., 2024). Even when initially well-calibrated, LLMs tend to become overconfident following RL, particularly when trained with reward signals that emphasize only correctness (Leng et al., 2024). This is a critical limitation in high-stakes domains such as healthcare or law, where models must not only be accurate but also communicate uncertainty when appropriate (Omar et al., 2024).

Motivated by the existing calibration challenges in reasoning models and inspired by progress in CoT reasoning, this work addresses two questions: **Can reasoning models be trained to improve both correctness and calibrated uncertainty? And can reasoning explicitly about uncertainty further enhance calibration?**

Our approach draws on statistical decision theory, specifically the theory of **proper scoring rules**: cost functions that are minimized when confidence scores reflect true probabilities (Gneiting & Raftery, 2007). A canonical example is the **Brier score**, $r_{\text{brier}} = -(p - c)^2$, where p is the model's stated confidence and c is a binary variable indicating the correctness of a model's output (Brier, 1950).

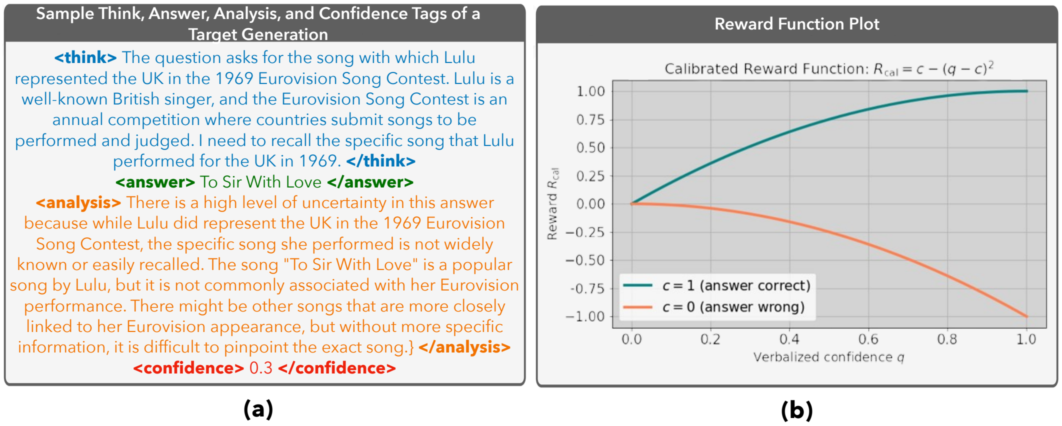

Figure 1: **(a)**: Sample CoT using <think>, <answer>, <analysis>, and <confidence> tags. **(b)**: A plot of the reward function changing as a function of verbalized confidence q.

While proper scoring rules are widely used in forecasting, they have seen limited application in training LLMs with RL. We bridge this gap by introducing a calibrated reward function that combines a binary correctness reward with a Brier score applied to a "verbalized confidence" within a model's own output. Our approach trains models to reason about both correctness and confidence in their reasoning chain. We prove that our designed reward function incentivizes truthful confidence reporting while maximizing task performance at the same time. Empirically, we demonstrate on a multi-hop reasoning benchmark that: (1) our reward formulation matches the task accuracy of models trained with binary correctness while significantly improving calibration; (2) our calibrated reasoning model achieves significantly better out-of-distribution calibration compared to a trained classifier; and (3) reasoning about uncertainty in CoT positively impacts calibration.

Other Approaches to Confidence Estimation. Confidence estimation in LLMs typically falls into three categories. Post-hoc verbalizations prompt models to state confidence after answering (Tian et al., 2023; Xiong et al., 2023; Yang et al., 2024), but these are often overconfident and weakly correlated with correctness. Sampling-based methods use response agreement (e.g., majority vote or best-of-N) as a proxy for confidence, but are costly and require clear ground truth (Kang et al., 2025). Internal probing extracts confidence from model features like token probabilities (Gupta et al., 2024), offering fine-grained scores but lacking generality. Recently, RL-based methods train models for calibrated verbal confidence (Xu et al., 2024; Stangel et al., 2025), but do not optimize accuracy. Optimizing exclusively for calibration can unintentionally degrade task accuracy—especially in larger models that might reward hack by deliberately outputting incorrect answers with 0 confidence to get perfect calibration. In contrast, our approach jointly optimizes correctness and calibration via a principled reward, focusing on the advantages of explicitly reasoning about uncertainty.

2 Method

We propose a new formulation where after reasoning about the solution and producing an answer, the model analyzes its uncertainty in the solution and outputs a verbalized confidence estimate q. Figure 1a illustrates a sample CoT that follows this formulation. Given the correctness label of the response $c = \texttt{verify}(y, y_{\text{gold}})$ and the model's verbalized confidence q, we design a reward function that jointly encourages correctness and calibration:

$$r_{\text{calibrated}}(q, c) = c - (q - c)^2$$

This reward incentivizes correctness but penalizes models when they output incorrect answers with high confidence or correct answers with low confidence. Figure 1b illustrates how the reward changes as a function of the verbalized confidence q. Notably, two properties of this reward make it particularly desirable (both proven in Appendix A.1) -

1. *Our calibrated reward incentivizes maximizing correctness:* Like the standard correctness reward, our calibrated reward is maximized when models select answers with the highest probability of being correct. This prevents collapse to degenerate policies where the

model learns to output incorrect answers with 0 confidence. In contrast, unbounded scoring rules (e.g., log loss) can incentivize such collapse (discussed in Appendix A.2).

2. *Our calibrated reward incentivizes truthful confidence reporting:* The expected reward is maximized when the model's reported confidence q matches its true belief about correctness. As a result, models are incentivized to report their genuine uncertainty.

Finally, we use a format reward to encourage adherence to a structured format. In addition to <think> and <answer> tags, we require an <analysis> tag to enclose uncertainty reasoning and a <confidence> tag for verbalized confidence. In Appendix B.6, we detail our system prompt, which provides guidelines for uncertainty reasoning.

3 Experiments

Dataset We use a modified HotPotQA distractor dataset with multi-hop questions and 10 paragraphs (2 relevant, 8 distractors) (Yang et al., 2018). To test uncertainty reasoning, *HotPotQA-Modified* removes 0, 1, or both relevant paragraphs, creating varying information completeness. The dataset is evenly split across these conditions, with 8 paragraphs per example. We train on 20,000 examples and evaluate using exact match.

Training Details We use GRPO as base RL algorithm with small modifications (see Appendix B.2). We initialize from Qwen2.5-7B base model, and do not use KL regularization.

Methods We evaluate the following methods:

1. **RL-calibrated (ours):** Training using our calibrated reasoning reward.
2. **RL-correctness:** Training using binary correctness reward with only <think> and <answer> tags. During evaluation, the model is asked to verbalize its confidence.
3. **Learned Classifier:** A classifier trained using responses from the RL-correctness model. The classifier is initialized from Qwen2.5-7B Base and trained with BCE loss. This method is expensive as it requires training and storing 2 models in memory.
4. **Qwen2.5-7B Base and Qwen2.5-7B Instruct:** Off-the-shelf models prompted to solve the question and reason about/verbalize their confidence.

Evaluation Metrics: We report **Accuracy (↑), AUROC (↑), Brier Score (↓), and Expected Calibration Error (ECE) (↓)**. Accuracy measures task performance; AUROC evaluates discrimination between classes. Brier and ECE quantify calibration, with lower values indicating better confidence alignment. More details about the metrics in Appendix B.5.

3.1 Results

In-Distribution Performance Our calibrated reasoning reward is the sum of 2 components: the binary correctness reward and the Brier score (negated). Fig. 3 shows the training curves for *RL-calibrated* (ours) and *RL-correctness*. Both the correctness and Brier reward for our method increase smoothly, indicating that the model is able to jointly improve accuracy and calibration. Table 1 shows results on 1,000 test examples from the original HotpotQA distractor dataset. RL-trained models outperform off-the-shelf models in multi-hop accuracy, confirming RL's effectiveness. Our method matches RL-correctness in accuracy, showing the calibration term doesn't hurt performance. Base, instruct, and RL-correctness models are highly overconfident and poorly calibrated—consistent with prior work. In contrast, our method and the classifier are much better calibrated, with our method slightly ahead.

Generalization We evaluate generalization performance on six diverse datasets covering factual, math, science, and commonsense: TriviaQA, SimpleQA, MATH500, GSM8K, CommonsenseQA, and GPQA. Table 1 presents the average performance across these datasets (individual results in Appendix C.2). The base model's accuracy closely matches that of the models trained with RL, indicating that RL training on HotpotQA does not enhance OOD reasoning. Consistent with in-distribution findings, off-the-shelf models and the RL-correctness model exhibit poor calibration. *In contrast, our calibrated RL approach achieves*

Method	HotpotQA				O.O.D Averaged			
	Acc.	AUROC	Brier	ECE	Acc.	AUROC	Brier	ECE
RL-cal (**ours**)	62.8%	**0.68**	**0.21**	**0.03**	**55.9%**	**0.68**	**0.21**	**0.21**
RL-correctness	**63.2**%	0.50	0.37	0.37	53.5%	0.50	0.46	0.47
Classifier	63.2%	0.65	0.22	0.07	53.5%	0.58	0.27	0.24
Qwen-7B Base	39.1%	0.55	0.53	0.54	53.3%	0.53	0.41	0.40
Qwen-7B Instruct	47.8%	0.64	0.40	0.41	48.4%	0.61	0.37	0.38

Table 1: Performance on HotpotQA and 6 out-of-distribution datasets.

substantial gains over all baselines across every calibration metric while maintaining (and even slightly overperforming) on task accuracy. Notably, the performance gap between the classifier and our method widens significantly in out-of-distribution settings. We hypothesize the differences could be due to -

1. **Uncertainty CoT:** Explicitly reasoning about uncertainty can improve calibration by allowing longer reflection on confidence, in line with recent work (Yoon et al., 2025).
2. **Training dynamics of RL.** During RL training, the model's confidence analysis and scores have to constantly adapt to the model's improving task performance. This non-stationarity might lead to more robust learning and better generalization.

Does reasoning about confidence help? Recent work has shown that CoT reasoning can be unfaithful, with generated CoTs that do not influence their final answers (Chen et al., 2025). This raises the possibility that the confidence analysis may not meaningfully inform the verbalized confidence score. To test this, we train two classifiers:

1. **Baseline classifier:** Trained on reasoning traces of the *RL-correctness* model.
2. **Analysis classifier:** Trained on reasoning traces of the *RL-calibrated* model with confidence scores (present within <confidence> tags) removed.

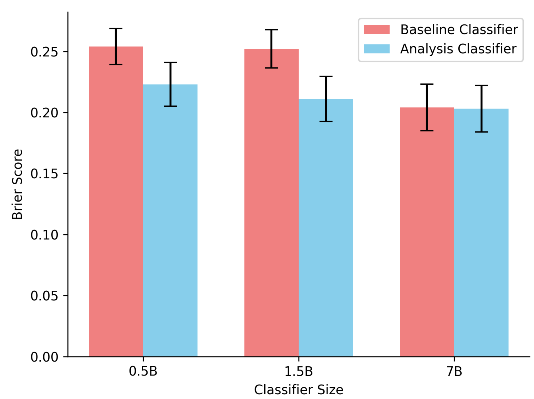

Figure 2: **Classifier performance across model sizes**.

As both RL models have comparable task accuracy, differences in classifier performance would indicate that uncertainty reasoning contains useful information for calibration. We train classifiers for 3 different model sizes of the Qwen-base model - 0.5B, 1.5B and 7B. Figure 2 shows Brier scores on HotPotQA-Modified. Interestingly, while 7B classifiers perform similarly, the *analysis classifier* outperforms the *baseline* at smaller sizes, suggesting classifier capacity is key. Large models can infer confidence from the solution trace alone, but smaller models get additional benefit from uncertainty reasoning in the CoT. This points to a deeper link between model size and CoT, which we leave to future work.

4 Conclusion

We show that incorporating proper scoring rules into RL enables reasoning models to improve both accuracy and calibration. Our approach trains models to reason about and verbalize uncertainty, preserving task performance while significantly improving calibration in- and out-of-distribution. These results highlight a path toward reasoning systems that are not only accurate, but also reliably communicate uncertainty.

References

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

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning models don't always say what they think, 2025. URL https://arxiv.org/abs/2505.05410.

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

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

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

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

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

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

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

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

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J Bell. Abstentionbench: Reasoning llms fail on unanswerable questions. *arXiv preprint arXiv:2506.09038*, 2025.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in llms: Reward calibration in rlhf. *arXiv preprint arXiv:2410.09724*, 2024.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

M Omar, BS Glicksberg, GN Nadkarni, and E Klang. Overconfident ai? benchmarking llm self-assessment in clinical scenarios. *medRxiv*, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

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

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

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

Changyi Xiao, Mengdi Zhang, and Yixin Cao. Bnpo: Beta normalization policy optimization. *arXiv preprint arXiv:2506.02864*, 2025.

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

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

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for llms, 2024. URL https://arxiv.org/abs/2412.14737.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Dongkeun Yoon, Seungone Kim, Sohee Yang, Sunkyoung Kim, Soyeon Kim, Yongil Kim, Eunbi Choi, Yireun Kim, and Minjoon Seo. Reasoning models better express their confidence. *arXiv preprint arXiv:2505.14489*, 2025.

A Proofs

A.1 Properties of our Calibrated Reward Function

Theorem 1. *Consider a task where the correctness of an answer given an input x is inherently stochastic, due to factors such as limited information or intrinsic randomness. Let $c \in 0, 1$ be a random variable denoting correctness, modeled as a Bernoulli random variable with success probability $p \in [0, 1]$. Let the model report a confidence $q \in [0, 1]$, and define the calibrated reward as:*

$$r_{calibrated}(c, q) = c - (q - c)^2,$$

Also define the correctness-only reward as $r_{correctness}(c) = c$. Then the following properties hold:

1. ***Calibration.*** *The model maximizes its expected calibrated reward if and only if it truthfully reports its confidence, $q = p$.*

2. ***Correctness Maximization.*** *For any fixed true probability p, the action that maximizes the expected calibrated reward also maximizes the expected correctness reward. Thus, maximizing calibration does not compromise correctness.*

Proof. For a given true probability p and reported confidence q, the expected calibrated reward V is:

$$V(p, q) = p \cdot \left(1 - (1 - q)^2\right) + (1 - p) \cdot (-q^2)$$

Simplifying this expression:

$$\begin{aligned} V(p, q) &= p \cdot \left(1 - (1 - q)^2\right) + (1 - p) \cdot (-q^2) \\ &= p \cdot (2q - q^2) - (1 - p)q^2 \\ &= (2p - q)q \end{aligned}$$

We begin by showing that for any given $p \in [0, 1]$, the expected reward $V(p, q)$ is maximized at $q = p$. Taking the derivative with respect to q:

$$\frac{\partial V}{\partial q} = 2p - 2q$$

Setting the derivative to zero to find the critical point:

$$\begin{aligned} 0 &= 2p - 2q \\ \Rightarrow \quad q &= p \end{aligned}$$

The second derivative is:

$$\frac{\partial^2 V}{\partial q^2} = -2$$

Since the second derivative is negative, this confirms that $q = p$ is a global maximum. Therefore, the reward function constitutes a proper scoring rule that incentivizes models to report their true confidence. This shows property (1), **Calibration**.

Next, we address property (2), **Correctness Maximization**. Evaluating the expected reward at the optimal confidence ($q = p$) gives:

$$\begin{aligned} V(p, p) &= p \cdot \left(1 - (1 - p)^2\right) + (1 - p) \cdot (-p^2) \\ &= p \cdot (1 - 1 + 2p - p^2) - (1 - p)p^2 \\ &= p \cdot (2p - p^2) - (1 - p)p^2 \\ &= 2p^2 - p^3 - p^2 + p^3 \\ &= p^2 \end{aligned}$$

Since the function p^2 is strictly monotonically increasing for $p \in (0, 1]$, it follows that $V(p, p)$ is also strictly monotonically increasing in p. This implies that among well-calibrated policies, those with higher correctness probabilities are always preferred under the calibrated reward function.

□

Remark. Note that our proof treats uncertainty as aleatoric, arising inherently from the stochastic nature of the task itself. Although language models might typically possess epistemic uncertainty (due to incomplete knowledge), from the model's perspective, all uncertainty regarding the correctness of its answer given input x can be effectively considered aleatoric, reflecting the model's internal probabilistic belief about correctness.

A.2 Combining log loss with correctness reward

Instead of brier score, another possible design choice is to combine the commonly used log-loss (a proper scoring rule) with correctness:

$$r_{\text{ce}}(q, c) = c + c \cdot \log(q) + (1 - c) \cdot \log(1 - q)$$

For a given probability p and reported confidence q, the expected reward can be written as:

$$V(p, q) = p \cdot (1 + \log(q)) + (1 - p) \cdot (\log(1 - q))$$

Since the log loss in a proper scoring rule, we know that $V(p, q)$ is maximized when $q = p$.

Theorem 2. *Log loss does not satisfy the correctness maximization property.*

Proof. The expected reward under the combined log-loss and correctness reward is

$$V(p, p) = p \cdot (1 + \log p) + (1 - p) \cdot \log(1 - p)$$

Let $f(p) = V(p, p)$. Differentiating with respect to p:

$$f'(p) = 1 + \log p - \log(1 - p)$$

Simplifying:

$$f'(p) = \log\left(\frac{p}{1 - p}\right) + 1$$

To find where $f'(p) < 0$, set:

$$\log\left(\frac{p}{1 - p}\right) < -1$$

Exponentiating both sides:

$$\frac{p}{1 - p} < \frac{1}{e}$$

Solving for p:

$$p < \frac{1}{e + 1}$$

Since $1/(e + 1) \approx 0.269$, it follows that $f(p)$ decreases on $(0, 0.269)$ and increases on $(0.269, 1)$, hence is not monotonic.

Thus, the combined reward fails to satisfy the correctness maximization property. □

B Experimental Setup

B.1 Training Dataset

We use a modified version of the HotPotQA distractor dataset, which contains factual questions requiring multi-hop reasoning. (Yang et al., 2018). Each example in this setting presents ten paragraphs, only two of which contain the information necessary to answer the question; the remaining eight paragraphs include closely related but irrelevant details. Consequently, solving this task requires the model to identify and reason over the pertinent passages. To more strongly develop uncertainty reasoning capability, we construct a new dataset, *HotPotQA-Modified*, in which we systematically remove either 0, 1, or both of the key paragraphs required to answer each question. This modification introduces varying levels of informational completeness that the model must reason over. We distribute questions across three equal groups: one-third have no relevant paragraphs (0/8), one-third have 1 relevant paragraph (1/7), and one-third have both relevant paragraphs (2/6). Each question consistently contains 8 total paragraphs. Our training dataset consists of 20,000 examples. We measure correctness using exact-match.

B.2 Training Details

We use GRPO as our base RL algorithm with some small modifications (Shao et al., 2024). Following Turtel et al. (2025), we remove the standard deviation division in the advantage, which might help with learning on examples where there are extreme miscalibrations. We use the BNPO loss function, which aggregates token level losses using the number of active tokens in the local training batch (Xiao et al., 2025). We generate 32 responses per prompt with a temperature of 0.7, and use an effective batch size of 2048. We train for 1 epoch, use a constant learning rate of $1e-6$ and a maximum response length of 1536. We initialize from the Qwen2.5-7B base model, and do not use any KL regularization. We use the *Long Analysis* system prompt for *RL-calibration* and the *Simple Generation* prompt for *RL-correctness* (see Appendix B.6). Finally, both format and calibration rewards are weighted equally.

B.3 Evaluation Datasets

We run evaluation on a large number of datasets:

1. **HotPotQA (Distractor):** We use 1000 validation examples from the original HotpotQA distractor dataset. We slightly modify the dataset and remove 2 non-relevant paragraphs from each question. Thus, each question has 8 paragraphs with both supporting paragraphs present. We measure correctness using exact-match (Yang et al., 2018).
2. **HotPotQA-Modified:** We evaluate on 500 held-out validation examples from the training dataset. We measure correctness using exact-match.
3. **TriviaQA:** We use 2000 examples from the validation set of the TriviaQA dataset (Joshi et al., 2017). We use the no-context split to purely test factual accuracy.We evaluate using LLM-as-a-judge.
4. **SimpleQA:** We use the full SimpleQA dataset consisting of 4326 factual questions (Wei et al., 2024). We evaluate using LLM-as-a-judge.
5. **Math-500** We use the popular MATH-500 dataset, which contains a subset of problems from the original MATH dataset (Hendrycks et al., 2021). We evaluate using *math-verify*, a mathematical expression evaluation system released by huggingface.
6. **GSM8K:** We use the test set (1319 problems) of the popular Grade School Math 8K dataset (Cobbe et al., 2021). We evaluate using *math-verify*.
7. **CommonSenseQA:** We use the validation set (1220 problems) of the CommonsenseQA dataset (Talmor et al., 2018), a multiple-choice question answering dataset that requires different types of commonsense knowledge to predict the correct answers. We evaluate using LLM-as-a-judge.

8. **GPQA:** We use the GPQA main dataset containing 448 multiple-choice questions written by experts in biology, physics, and chemistry (Rein et al., 2024).

B.4 Evaluation Details

All models are evaluated with temperature 0. For all datasets except Math and GSM8K, we use a maximum token budget of 2048. For Math and GSM8K, this is increased to 4096. The system prompt for evaluation and the pipeline to extract answer and confidence scores varies slightly based on the method we are evaluating:

1. **RL-calibrated (ours):** Our method trains model with the *Long Analysis Prompt* that guides them to use `<think>`, `<answer>`, `<analysis>` and `<confidence>` tags. They are evaluated with the same system prompt. We extract their answer from the `<answer>` tag and their confidence from the `<confidence>` tag.
2. **RL-correctness:** This baseline method is trained with the binary correctness reward with the *Standard Generation* system prompt that guides them to use the `<think>` and `<answer>`. They are evaluated with the same system prompt and we extract their answer from the `<answer>` tag. To obtain their verbalized confidence, we append *"Thinking time ended. My verbalized confidence in my answer as a number between 0 and 100 is equal to"* to their generated output.
3. **Classifier:** The classifier is conditioned on the question and the *RL-correctness* model's generation (without any verbalized score present in the input). The classifier method thus uses *RL-correctness* as a generator and their reported accuracies in the result tables are equal.
4. **Qwen2.5-7B Base:** The base model is not good at instruction following and is prompted with a simpler system prompt (*Simple Confidence Prompt*) that guides it to use `<think>`, `<answer>` and `<confidence>` tags. In case no valid confidence can be extracted, we append *"Thinking time ended. My verbalized confidence in my answer as a number between 0 and 100 is equal to"* to their output and call them again for confidence extraction.
5. **Qwen2.5-7B Instruct:** The instruct model is good at instruction following and we use the exact evaluation pipeline as our *RL-calibrated* method for it.

For all methods, if we unable to extract a valid answer from the `<answer>` tags, we append *"Thinking time ended. My final answer is"* to their output and call them again. The main goal of these custom pipelines is to be able to fairly extract an answer and a confidence level and minimize cases where incorrect formatting adversely affects performance. Note that because they are trained with format rewards, both the RL-trained models are nearly perfect in adhering to the desired format and require minimal interventions. However, off-the-shelf models (particularly the base model) benefits from this full extraction pipeline. **Importantly, once the answers and confidence scores have been extraced, all methods are evaluated identically and based on the dataset, exact-match, LLM-as-a-judge or math-verify is used.**

LLM-as-a-judge: We use Llama-3.1-8B-Instruct with temperature set to 0 as our judge. The judge is provided with the question, the ground truth answer and the answer extracted from the evaluation pipeline. It is prompted to respond with "YES" or "NO" based on the correctness of the answer. As the datasets we evaluate have short and objective answers, we do not condition the judge on the thinking traces which can add biases.

B.5 Evaluation Metrics

We use the following evaluation metrics:

1. **Accuracy (↑):** A measure of performance.
2. **Area under ROC curve (AUROC) (↑):** Measures ability of classifier to distinguish between positive/negative classes across thresholds. Higher is better.
3. **Brier Score (↓):** Squared difference between confidence and ground truth.

4. **Expected Calibration Error (ECE) (↓):** Calibration metric that groups confidences into bins and computes difference between the average correctness and confidence.

B.6 System Prompts

.

Long Analysis Prompt

”A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind, provides the user with the final answer, then analyzes its confidence about the solution and then provides the user with its confidence level. The confidence level is a number between 0 and 1 (inclusive) enclosed within `<confidence> </confidence>` tags. The final answer is enclosed between `<answer> </answer>` tags. The analysis about confidence and uncertainty is enclosed within `<analysis> </analysis>` tags. The assistant should reason about its confidence in the solution and its uncertainty in the solution within these tags. Here are some guidelines for the analysis:
1. Your task is to point out things where the model could be wrong in its thinking, or things where there might be ambiguity in the solution steps, or in the reasoning process itself.
2. You should not suggest ways of fixing the response, your job is only to reason about uncertainties.
3. For some questions, the response might be correct. In these cases, It is also okay to have only a small number of uncertainties and then explicitly say that I am unable to spot more uncertainties.
4. Uncertainties might be different from errors. For example, uncertainties may arise from ambiguities in the question, or from the application of a particular lemma/proof.
5. If there are alternate potential approaches that may lead to different answers, you should mention them.
6. List out plausible uncertainties, do not make generic statements, be as specific about uncertainties as possible.
7. Enclose this uncertainty analysis within `<analysis> </analysis>` tags.
The final format that must be followed is : `<think>` reasoning process here `</think> <answer>` final answer here `</analysis> <analysis>` analysis about confidence and uncertainty here `</analysis> <confidence>` confidence level here (number between 0 and 1) `</confidence>`)

.

Simple Confidence Prompt

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and analyzes its confidence about the solution and then provides the user with the final answer as well as its confidence level. The confidence level is a number between 0 and 1 (inclusive) enclosed within `<confidence> </confidence>` tags. The final answer is enclosed between `<answer> </answer>` tags. The final format that must be followed is : `<think>` reasoning process here `<//think><answer>` final answer here `</answer> <confidence>` confidence level here (number between 0 and 1) `</confidence>`.

.

Simple Generation Prompt

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think><answer> answer here </answer>.

C Results

C.1 Training Curves

Fig. 3 shows the training curves for *RL-calibrated* (ours) and *RL-correctness*. Both the correctness and Brier reward for our method increase smoothly, indicating that the model is able to jointly improve accuracy and calibration. The completion lengths of our method are significantly longer due to uncertainty analysis, and gradually increase during training.

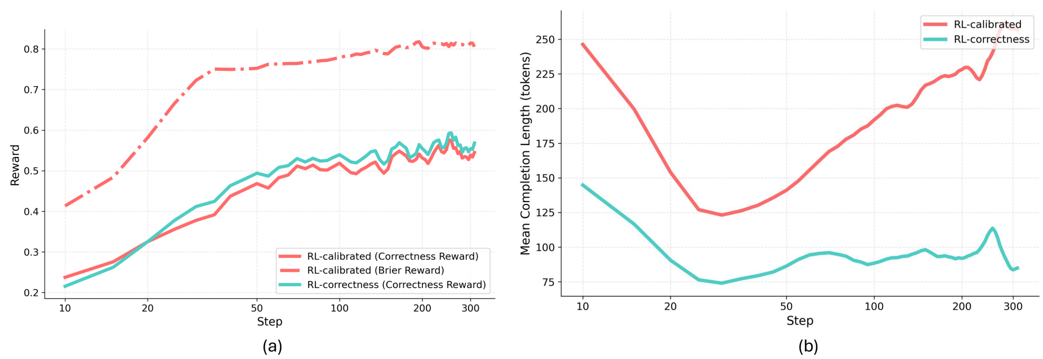

Figure 3: **(a) Reward curves for *RL-calibrated* (ours) and *RL-correctness* (baseline)**. Both the Brier reward ($-(\hat{p}-c)^2$) and correctness reward improve under our method, indicating joint gains in reasoning and calibration. The Brier reward is shifted upward by 1 for clarity. **(b) Completion lengths during training** The completion lengths of our method are significantly longer due to uncertainty analysis, and gradually increase during training.

C.2 Individual Dataset Results

Math Domains:

Method	MATH				GSM8K			
	Acc.	**AUROC**	**Brier**	**ECE**	**Acc.**	**AUROC**	**Brier**	**ECE**
RL-cal (**ours**)	43.0%	**0.710**	**0.257**	**0.209**	**86.6%**	**0.744**	**0.141**	0.198
RL-correctness	37.2%	0.501	0.626	0.626	81.0%	0.502	0.189	0.189
Classifier	37.2%	0.699	0.262	0.235	81.0%	0.580	0.157	**0.081**
Qwen-7B Base	**46.0%**	0.561	0.495	0.500	73.3%	0.526	0.242	0.217
Qwen-7B Instruct	41.2%	0.530	0.492	0.501	50.1%	0.552	0.388	0.375

Table 2: Performance on Math-500 and GSM8K. Best values bolded.

Factual Domains:

Method	SimpleQA				Trivia			
	Acc.	AUROC	Brier	ECE	Acc.	AUROC	Brier	ECE
RL-cal (**ours**)	11.8%	0.592	**0.242**	**0.340**	61.3%	**0.727**	**0.202**	**0.055**
RL-correctness	12.1%	0.501	0.878	0.878	**61.9%**	0.501	0.381	0.380
Classifier	12.1%	0.478	0.533	0.641	**61.9%**	0.567	0.257	0.153
Qwen-7B Base	**13.5%**	0.502	0.772	0.808	58.2%	0.518	0.377	0.367
Qwen-7B Instruct	8.4%	**0.736**	0.516	0.635	56.0%	0.638	0.354	0.355

Table 3: Performance on SimpleQA and Trivia datasets. Best values bolded.

Other Domains:

Method	CommonsenseQA				GPQA			
	Acc.	AUROC	Brier	ECE	Acc.	AUROC	Brier	ECE
RL-cal (**ours**)	**90.9%**	**0.723**	0.169	0.300	41.9%	**0.566**	**0.261**	**0.127**
RL-correctness	90.7%	0.500	**0.093**	0.093	37.9%	0.500	0.621	0.621
Classifier	90.7%	0.635	0.119	0.179	37.9%	0.526	0.277	0.159
Qwen-7B Base	88.9%	0.623	0.097	0.009	39.7%	0.468	0.499	0.501
Qwen-7B Instruct	88.5%	0.660	0.097	**0.017**	**46.0%**	0.548	0.401	0.390

Table 4: Performance on CommonsenseQA and GPQA. Best values bolded.

SimpleQA Example

Question

Who was awarded the Oceanography Society's Jerlov Award in 2018? (Ground Truth Answer: Annick Bricaud)

RL-Cal (ours)

<think> The question asks who was awarded the Oceanography Society's Jerlov Award in 2018. The Jerlov Award is given by the Oceanography Society, and it is named after the late Professor Gunnar Jerlov, a pioneer in the field of ocean optics. To find the winner of the 2018 award, I would need to look up the most recent information on the Oceanography Society's website or other reliable sources. </think>

<answer> Dr. David W. Schiedek </answer>

<analysis> The Oceanography Society's website or other reliable sources do not provide information on the winner of the 2018 Jerlov Award. It is possible that the information is not readily available or that the award was not given in 2018. Additionally, the name "Dr. David W. Schiedek" is not a well-known figure in the field of oceanography, which raises uncertainty about the accuracy of this answer. </analysis>

<confidence> 0.25 </confidence>

RL-Correctness

<think> The question asks for the person who was awarded the Oceanography Society's Jerlov Award in 2018. The Jerlov Award is given by the Oceanography Society to recognize outstanding contributions to the field of oceanography. The recipient for 2018 was Dr. Laurence C. Smith. </think>

<answer> Laurence C. Smith </answer>

<confidence> 100. </confidence>

Qwen-7B-Instruct

<think> To solve this problem, I need to recall the information about the Oceanography Society's Jerlov Award and identify the recipient for the year 2018. The Jerlov Award is given by the Oceanography Society to recognize outstanding contributions to the field of oceanography. I will need to search my knowledge base for the specific recipient in 2018. </think>

<answer> The Oceanography Society's Jerlov Award in 2018 was awarded to Dr. Lisa Levin. </answer>

<analysis> There is a potential uncertainty in the exact year the award was given, as the information might not be perfectly up-to-date in my knowledge base. Additionally, there could be a slight variation in the spelling of the recipient's name. However, based on the information I have, Dr. Lisa Levin is the correct recipient. </analysis>

<confidence> 0.95 </confidence>

Qwen-7B-Base

<think> To find out who was awarded the Oceanography Society's Jerlov Award in 2018, I will search for the most recent information on the award winners from the Oceanography Society's official website or a reliable source that covers scientific awards. </think>

<answer> The Oceanography Society's Jerlov Award in 2018 was awarded to Dr. Lisa Levin. </answer>

<confidence> 0.95 </confidence>

HotPotQA-Modified Example

All supporting information from the given question has been removed. The only way to answer correctly is if the model remembers the information in its weights.

Question and Supporting Information

Question: Jacques Sernas, actor in *Fugitive in Trieste*, was of what nationality? (Ground Truth Answer: Lithuanian-born French)
Your answer will be verified with exact match score. To ensure correct verification, only provide the answer within the <answer> </answer> tags. Do not put any sentences or reasoning process within the <answer> </answer> tags.

Supporting Information:
Paragraph 0
Man From 1997 is a time travel episode of the 1956–57 anthology television series *Conflict* directed by Roy del Ruth, produced by Roy Huggins, written by James Gunn from a story by Alfred Bester, and starring Jacques Sernas, Charles Ruggles, Gloria Talbott and James Garner. The music was written by David Buttolph and the cinematographer was Ted D. McCord. The show was originally telecast on November 27, 1956 and a kinescope of the broadcast currently exists.

Paragraph 1
Altair is a 1956 Italian romantic drama film directed by Leonardo De Mitri and starring Franco Interlenghi, Antonella Lualdi and Jacques Sernas.

Paragraph 2
Dieci canzoni d'amore da salvare (English title: "Ten Love Songs") is a 1953 Italian film directed by Flavio Calzavara. The plot concerns a songwriter, played by Jacques Sernas, who leaves his sweetheart and publisher when he learns that he is going blind. Supporting Sernas were Brunella Bovo, Franca Tamantini, and Enrico Viarisio.

Paragraph 3
Loving You Is My Sin (Italian: *Amarti è il mio peccato*) is a 1953 Italian melodrama film directed by Sergio Grieco and starring Jacques Sernas, Luisa Rossi and Elisa Cegani.

Paragraph 4
It's All Adam's Fault or *In Six Easy Lessons* (French: *C'est la faute d'Adam*) is a 1958 French comedy film directed by Jacqueline Audry and starring Dany Robin, Jacques Sernas and Mijanou Bardot.

Paragraph 5
Balearic Caper (Spanish: *"Zarabanda bing bing"*, Italian: *"Baleari Operazione Oro"*, French: *"Barbouze chérie"*, also known as *Operation Gold*) is a 1966 Spanish-Italian-French heist-Eurospy comedy film written and directed by José María Forqué and starring Jacques Sernas, Daniela Bianchi and Mireille Darc. It was shot in Ibiza.

Paragraph 6
Finishing School (Italian: *Fanciulle di lusso*) is a 1953 French-Italian comedy film directed by Bernard Vorhaus and starring Susan Stephen, Anna Maria Ferrero and Jacques Sernas. It was made at Cinecittà with sets designed by the art director Franco Lolli. It is also known by the alternative title of *Luxury Girls*.

Paragraph 7
Jump into Hell is a 1955 war film directed by David Butler. The film stars Jacques Sernas (billed as "Jack Sernas") and Kurt Kasznar. The first contemporary Hollywood war film of the war in Indochina, the story is a fictionalized account of the Battle of Dien Bien Phu.

RL-Cal

<think> The question asks for the nationality of Jacques Sernas, the actor in Fugitive in Trieste. The information provided mentions Jacques Sernas in several films, but does not specify his nationality. However, it does mention that he starred in Altair, a 1956 Italian romantic drama film, and that he was an actor in several French and Italian films. This suggests that Jacques Sernas might be of French or Italian nationality, as he starred in films produced in those countries. However, there is no direct information about his nationality in the given text. </think>

<answer> French </answer>

<analysis> The information provided mentions Jacques Sernas in several French and Italian films, and he starred in Altair, a 1956 Italian romantic drama film. However, there is no direct information about his nationality. While it is possible that he could be of French or Italian nationality, the lack of specific information makes it uncertain. Additionally, the fact that he starred in both French and Italian films could suggest that he might be of mixed nationality or have connections to both countries. Due to the ambiguity and lack of direct information, the confidence in this answer is relatively low. </analysis>

<confidence> 0.4 </confidence>

RL-Correctness

<think> The question asks for the nationality of Jacques Sernas, the actor in Fugitive in Trieste. Jacques Sernas was an American actor. </think>

<answer> American </answer>

<confidence> 100. </confidence>

Qwen-7B-Instruct

<answer> American </answer>

<analysis> There are several uncertainties in this problem. First, the supporting information does not explicitly state Jacques Sernas' nationality. Second, the films mentioned (Fugitive in Trieste, Man From 1997, Altair, Dieci canzoni d'amore da salvare, Loving You Is My Sin, It's All Adam's Fault, Balearic Caper, Finishing School, and Jump into Hell) are from different countries, which does not provide a clear indication of Sernas' nationality. Third, the information provided is primarily about Sernas' filmography and does not include any biographical details that could confirm his nationality. </analysis>

<confidence> 0.75 </confidence>

Qwen-7B-Base

<think> To determine Jacques Sernas' nationality, I will analyze the information provided in the supporting information.
From the information given, I can see that Jacques Sernas has appeared in various films from different countries, such as Italy, France, and Spain. However, there is no specific mention of his nationality in the provided information.
Since there is no direct information about Jacques Sernas' nationality, I will analyze the context of the question. The question is about Jacques Sernas' nationality in the context of the film "Fugitive in Trieste." Trieste is a city in Italy, and the film is an Italian romantic drama.
Based on this context, it is reasonable to assume that Jacques Sernas is of Italian nationality. However, without any direct information, I cannot be 100% confident in this assumption. </think>

<answer> Italian </answer>

<confidence> 0.8 </confidence>