Balancing Classification and Calibration Performance in Decision-Making LLMs via Calibration Aware Reinforcement Learning

Duygu Nur Yaldiz[1*] **Evangelia Spiliopoulou**[2] **Zheng Qi**[2]
Siddharth Varia[2] **Srikanth Doss**[2] **Nikolaos Pappas**[3†]
[1]University of Southern California [2]AWS AI Labs [3]Oracle
yaldiz@usc.edu

Abstract

Large language models (LLMs) are increasingly deployed in decision-making tasks, where not only accuracy but also reliable confidence estimates are essential. Well-calibrated confidence enables downstream systems to decide when to trust a model and when to defer to fallback mechanisms. In this work, we conduct a systematic study of calibration in two widely used fine-tuning paradigms: supervised fine-tuning (SFT) and reinforcement learning with verifiable rewards (RLVR). We show that while RLVR improves task performance, it produces extremely overconfident models, whereas SFT yields substantially better calibration, even under distribution shift, though with smaller performance gains. Through targeted experiments, we diagnose RLVR's failure, showing that decision tokens act as extraction steps of the decision in reasoning traces and do not carry confidence information, which prevents reinforcement learning from surfacing calibrated alternatives. Based on this insight, we propose a calibration-aware reinforcement learning formulation that directly adjusts decision-token probabilities. Our method preserves RLVR's accuracy level while mitigating overconfidence, reducing ECE scores up to 9 points.

1 Introduction

Large Language Models (LLMs) are increasingly deployed in high-stakes decision-making systems such as content moderation, medical assistance, financial services, and legal frameworks (Chakraborty et al., 2025; Eigner and Händler, 2024). In these critical domains, models must not only produce accurate predictions but also provide reliable confidence estimates that accurately reflect their true likelihood of being correct. For example, a guard model must flag unsafe content with high confidence while remaining uncertain on ambiguous cases, and a clinical assistant must express uncertainty when multiple diagnoses are plausible. On the other hand, unreliable confidence can have serious consequences. For instance, an overconfident guard model may incorrectly classify harmful content as safe, resulting in severe downstream risks. This issue, commonly referred to as the confidence calibration problem, arises when the model's predicted confidence levels do not align with its actual accuracy (Wang et al., 2024; Damani et al., 2025; Stangel et al., 2025). In decision-making tasks, well-calibrated confidence scores are essential for safe and trustworthy operations, as they enable practitioners to determine when to trust or override a model's decision.

Prior work addresses the calibration problem from three main perspectives. The first focuses on uncertainty estimation: developing strategies to quantify model confidence or uncertainty reliably (Bakman et al., 2025b). The second centers on post-hoc calibration, which adjusts model outputs after training to better align predicted probabilities with empirical accuracy (Guo et al., 2017). The third explores calibration-aware fine-tuning, where the training objective is modified to produce calibrated models for a chosen type of confidence estimation (Damani et al., 2025). Unlike post-hoc or uncertainty-estimation approaches that operate after training, calibration-aware fine-tuning directly targets calibration during model adaptation. However, most existing methods in this category focus on verbalized calibration (Kadavath et al., 2022; Tian et al., 2023) for open-ended generations, which requires prompting the model to express confidence in its own output, a costly and often impractical setup for large-scale or low-latency applications.

In contrast, our work studies probability-based confidence in decision-making tasks, specifically, the probability of the final decision token, and

[*]Work done during an internship at AWS AI Labs.
[†]Work done while at AWS AI Labs.

analyzes how different fine-tuning paradigms affect its calibration. We find that commonly used paradigms yield opposite effects: supervised fine-tuning (SFT) improves calibration while reinforcement learning with verifiable rewards (RLVR) enhances task performance but leaves models highly overconfident, reducing the usefulness of confidence scores. Building on this analysis, we diagnose the source of miscalibration in RLVR on decision-making tasks and propose a calibration-aware reinforcement learning approach that directly regulates the probability of the decision token (since the confidence is derived from it), balancing the trade-off between accuracy and calibration.

We summarize our contributions as follows:

- We conduct a systematic empirical study comparing SFT and RLVR across multiple decision-making benchmarks. Our results reveal a consistent trade-off: both SFT and RLVR improve performance over the base model, but while RLVR achieves higher accuracy gains, SFT yields better-calibrated confidence estimates (Section 3).
- We diagnose the source of RLVR's miscalibration. Through targeted experiments, we show that the decision token inherits overconfidence from reasoning traces and that reinforcement learning cannot achieve calibration, since there are no calibrated paths to reinforce from the base model (Section 4).
- We propose a calibration-aware reinforcement learning formulation that directly adjusts decision-token probabilities to improve calibration along with accuracy. Our method consistently mitigates extreme overconfidence, produces more reliable confidence scores, and generalizes well to out-of-distribution settings (Section 5).

2 Preliminaries

Decision-Making with LLMs In decision-making tasks, an LLM is prompted to make a decision given a question q and a finite set of options $\mathcal{C} = \{c_0, c_1, ..., c_K\}$. Formally, we denote $\boldsymbol{x}$ as the full input to the model containing the instructions, question q, and the set of choices. The model, parameterized by θ, outputs a set of tokens $\boldsymbol{y} = \{y_0, y_1, ..., y_T\}$ conditioned on $\boldsymbol{x}$.

For models without explicit reasoning, the output reduces to a single decision token $\boldsymbol{y} = \{y_d\}$. For reasoning-enabled models, the output consists of intermediate reasoning tokens followed by a final decision token: $\boldsymbol{y} = \{y_0, y_1, \ldots, y_{T-1}, y_T\} = \{y_0, y_1, \ldots, y_{T-1}, y_d\}$. We consider the model's decision correct if the final decision token y_d matches the ground-truth label $\hat{c}$.

When answer choices consist of multiple tokens, we recast the task into a classification setting with atomic options (e.g., A, B, C, D). This formulation ensures that the model's decision can always be represented by a single token y_d.

Confidence Estimation There are several algorithmic approaches to estimating confidence in generative models (Bakman et al., 2025b), including verbalized confidence expression (Kadavath et al., 2022; Tian et al., 2023), sampling-based methods (Lin et al., 2024; Kuhn et al., 2023), and logit-based probability estimation (Malinin and Gales, 2021). In this work, we focus on probability-based confidence estimation. Unlike sampling-based methods, it does not require generating multiple outputs, nor does it rely on prompting the model to express verbalized confidence (Yaldiz et al., 2025). Instead, it leverages the probabilities of the output tokens, which is mostly accessible even for API-based models (OpenAI, 2023). Moreover, in many decision-making settings where models are deployed locally, direct access to logits is readily available.

Throughout this paper, we define the model's confidence as the probability assigned to the decision token. For example, in a binary decision task (e.g., "yes" or "no"), if the model outputs "yes" (either after reasoning or directly as the model output), then the probability assigned to the "yes" token is treated as the model's confidence. Formally, we define confidence as follows:

$$C(\boldsymbol{x}, \boldsymbol{y}; \theta) = P(y_d | \boldsymbol{x}, y_{<d}; \theta)$$

where $y_{<d}$ denotes the tokens generated before y_d.

Confidence Calibration is the alignment between model's predicted confidence and its actual accuracy. A well-calibrated model should exhibit the property that among all predictions made with confidence p, approximately a fraction p of them are correct, i.e:

$$P(y_d = \hat{c} \mid C(\boldsymbol{x}, \boldsymbol{y}; \theta) = p) \cong p.$$

Intuitively, this means that, if we group all predictions with a confidence score of 0.8, we expect that approximately 80% of them should be correct under perfect calibration.

A common tool to visualize calibration quality is the reliability diagram (Guo et al., 2017). It

plots the observed accuracy against the predicted confidence by binning predictions into discrete confidence intervals. In a perfectly calibrated model, the plotted curve should align with the diagonal line where accuracy equals confidence, indicating that confidence values match empirical accuracies. Deviations from this diagonal highlight regions where the model is overconfident or underconfident. It is also a common practice to plot confidence distributions along with reliability diagrams (Guo et al., 2017). Confidence histograms show how frequently the model assigns probabilities across the confidence range, highlighting whether predictions concentrate near a single value. When most scores cluster tightly at a level, the confidence signal is less informative, limiting its practical utility.

Expected Calibration Error (ECE) is a widely used metric to express the confidence calibration numerically (Guo et al., 2017). It partitions predictions into M bins based on confidence levels and computes the weighted average of absolute differences between confidence and accuracy within each bin. Formally,

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

where n is the size of the dataset $\{\boldsymbol{x}^k, \hat{c}^k\}_{k=1}^n$, $\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{k \in B_m} \mathbb{1}(y_d^k = \hat{c}^k)$ denotes the accuracy of predictions in bin B_m, and $\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{k \in B_m} C(\boldsymbol{x}^k, \boldsymbol{y}^k; \theta)$ is the average confidence of predictions in that bin. In our implementation, we use bins of equal size. Nixon et al. (2019) refers to this formulation as Adaptive Calibration Error.

Interpreting Calibration requires considering all three indicators jointly: confidence distributions, reliability diagrams, and ECE scores. Confidence distributions reveal whether the model meaningfully differentiates between levels of certainty. For example, if scores cluster tightly within a narrow range, the confidence signal lacks utility, and the model can be regarded as uncalibrated for practical purposes. When confidence values span a broader range, the ECE score serves as a good quantitative summary of calibration quality, while reliability diagrams provide complementary insights by illustrating where the model tends to be overconfident or underconfident.

Goal: Accurate and Calibrated Model In decision-making applications, large language models must not only achieve strong task performance but also provide calibrated confidence estimates. Calibration is crucial to determine when to trust a model's decision: if confidence is sufficiently low, external mechanisms such as invoking a larger model or human oversight can be activated. However, without calibration, overconfident models can obscure errors, causing unreliable predictions to appear trustworthy.

Our objective is therefore to train models that simultaneously maximize task accuracy and minimize calibration error. We investigate whether common fine-tuning paradigms, widely used to adapt off-the-shelf models to target tasks, can achieve this ideal. In the next section, we analyze popular fine-tuning strategies with respect to their impact on both task accuracy and confidence calibration.

3 The Calibration–Classification Tradeoff in Fine-Tuning Paradigms

As outlined previously, an ideal decision-making model must achieve strong task performance while maintaining well-calibrated confidence. In practice, however, these two goals might not be achieved simultaneously, a phenomenon we refer to as the *calibration–classification trade-off*. Models fine-tuned with reinforcement learning (e.g., GRPO) tend to achieve higher accuracy but remain overconfident, whereas supervised fine-tuning (SFT) yields better calibration with more modest performance gains. In this section, we quantify this trade-off across multiple datasets and model sizes, highlighting its consistency across settings. In Section 5, we present a calibration-aware reinforcement learning solution that eliminates RLVR's extreme overconfidence, while maintaining task performance.

3.1 Fine-Tuning Algorithms

In this work, we consider two fine-tuning paradigms among the most widely used ones. First, we consider supervised fine-tuning (SFT). Second, we investigate reinforcement learning. It is well-established that encouraging explicit reasoning improves model performance (Wei et al., 2022; Wang et al., 2023). Moreover, when reasoning traces are available, they provide partial groundings for the model's final decision, which reinforcement learning algorithms can exploit. In this work, we focus on reinforcement learning with verifiable rewards (RLVR), which is particularly well-suited to decision-making tasks because correctness can be explicitly verified against ground-truth labels, enabling a stronger training signal than standard

preference-based reinforcement learning. Next, we provide the details of both algorithms.

SFT SFT adapts a model using labeled examples $(\boldsymbol{x}, \hat{c})$, where $\boldsymbol{x}$ is the input (instructions, question, and choices) and $\hat{c}$ is the ground-truth decision token. Training minimizes the cross-entropy loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\log p_\theta(\hat{c} \mid \boldsymbol{x}), \tag{1}$$

directly aligning the model's output with the correct decision.

RLVR In reinforcement learning, the goal is to learn model parameters θ that maximize the expected reward over sampled outputs:

$$\max_\theta \ \mathbb{E}_{\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[R(\boldsymbol{y}, \boldsymbol{x})\right],$$

where π_θ is the model's output distribution parameterized by θ, and $R(\boldsymbol{y}, \boldsymbol{x})$ is a task-specific reward function that scores the quality or correctness of the output.

In RLVR, the reward function $R(\boldsymbol{y}, \boldsymbol{x})$ is designed to reflect the correctness of the model's output. Specifically, for decision making tasks, the reward is defined as:

$$R(\boldsymbol{y}, \boldsymbol{x}) = \begin{cases} 1 & \text{if } y_d = \hat{c}, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where y_d is the model's predicted decision token, and $\hat{c}$ is the ground-truth label. This binary reward encourages the model to generate outputs that match the correct class. As an RLVR algorithm, we incorporate Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which minimizes the following objective:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\boldsymbol{y}_i|}\sum_{t=1}^{|\boldsymbol{y}_i|}\min\left\{\frac{\pi_\theta(y_{i,t} \mid \boldsymbol{x}, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid \boldsymbol{x}, y_{i,<t})}\hat{A}_{i,t},\right.$$
$$\left.\text{clip}\left(\frac{\pi_\theta(y_{i,t} \mid \boldsymbol{x}, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid \boldsymbol{x}, y_{i,<t})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right\}$$

where ϵ is clipping hyperparameter and advantage $\hat{A}_{i,t}$ is calculated as:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\boldsymbol{r})}{\text{std}(\boldsymbol{r})} \tag{3}$$

where $r_i = R(\boldsymbol{y}_i, \boldsymbol{x})$ and $\boldsymbol{r}$ is the vector containing all the rewards in the group. Note that, in the original GRPO formulation there is KL term subtracted from the minimization term to prevent divergence from the original model. However, we remove it since it became a common practice (Hu et al., 2025; DeepSeek-AI et al., 2025). Moreover, we incorporate BNPO (Beta Normalization Policy Optimization) (Xiao et al., 2025a), which stabilizes training by normalizing the reward distribution within each batch, mitigating reward scale sensitivity and improving convergence.

3.2 Experimental Design

We incorporate two decision-making tasks, commonsense reasoning and content moderation, and three models: Qwen3-1.7B, Qwen3-4B, Qwen3-8B (Yang et al., 2025).

For commonsense reasoning, we use CommonsenseQA (Talmor et al., 2019) and OpenBookQA (Mihaylov et al., 2018), both of which are multiple-choice question-answering datasets. CommonsenseQA contains four answer options per question, while OpenBookQA provides five. For content moderation, we use OpenAI Moderation (Markov et al., 2023), and XSTest (Röttger et al., 2024). For both datasets, the task is formulated as binary classification, where the model determines whether a prompt is 'Safe' or 'Unsafe'.

For fine-tuning on the OpenAI Moderation dataset, which only provides a test split, we randomly subsample 500 examples to serve as training data and keep the same subset fixed across all experiments. The remaining examples are used for evaluation. For commonsense reasoning, we similarly subsample 500 examples from the training split of CommonsenseQA. The remaining datasets in each category are held out to evaluate out-of-distribution (OOD) generalization. Each model–task pair is fine-tuned independently under this setup.

		Qwen3-1.7B		Qwen3-4B		Qwen3-8B	
		Acc (↑)	ECE (↓)	Acc (↑)	ECE (↓)	Acc (↑)	ECE (↓)
CSQA†	Base	67.49	29.91	76.97	20.96	80.51	16.78
	SFT	68.55	**7.36**	79.12	**8.81**	81.33	**5.96**
	GRPO	**73.67**	24.39	**81.84**	16.98	**83.78**	14.80
OBQA*	Base	78.80	18.97	88.40	10.67	91.60	7.22
	SFT	79.20	**4.60**	89.80	**4.76**	91.80	**4.29**
	GRPO	**86.17**	11.52	**92.79**	6.74	**94.39**	4.88
OpenAI†	Base	82.20	16.17	74.07	24.29	81.78	17.33
	SFT	83.73	**3.52**	87.54	**5.26**	88.08	**3.91**
	GRPO	**87.80**	12.20	**88.98**	11.00	**88.81**	10.68
XSTest*	Base	74.89	22.97	80.89	18.18	83.56	15.45
	SFT	76.00	**11.93**	86.22	**8.26**	**86.78**	**6.84**
	GRPO	**79.78**	20.22	**87.78**	12.21	86.22	13.54

Table 1: Performance (accuracy) and calibration (ECE) metrics for base, SFT, and GRPO models. While † indicates in-domain, * denotes out-of-domain datasets.

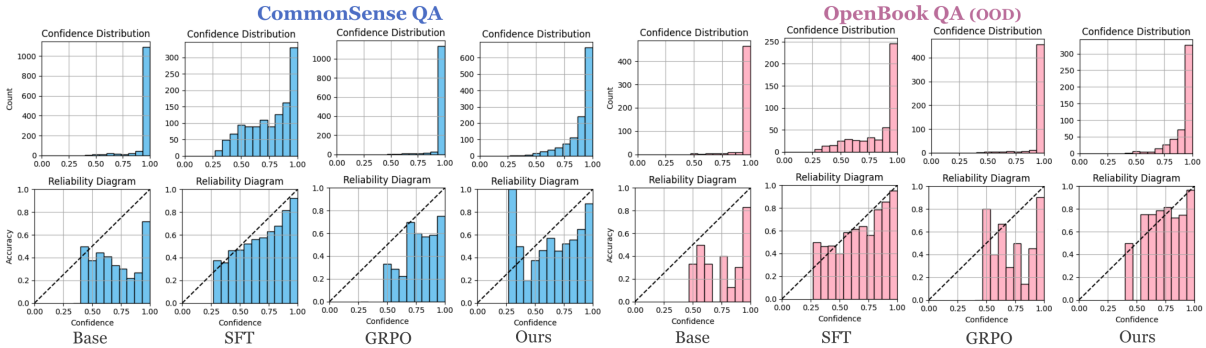

Figure 1: Reliability diagrams of Qwen3-1.7B model. Confidence distributions indicate that the Base and GRPO fine-tuned models exhibit extreme overconfidence, with most predictions assigned probabilities near 1. In contrast, SFT and our proposal produce a broader spread of confidence values and reliability diagrams that align more closely with the diagonal, indicating improved calibration.

3.3 Results and Discussion

We present the results in Table 1 and Figure 1.

Comparing SFT and GRPO in terms of task accuracy, we find that GRPO consistently produces better-performing models: on average, 3% better than SFT models. This outcome is expected, since GRPO-trained models are capable of generating reasoning traces before committing to a decision, whereas SFT models are directly supervised only on the final decision token, with no gold reasoning available in the datasets. Performance also improves steadily with increasing model size, as anticipated, though we observe no consistent trend with respect to calibration across sizes.

The base model is found to be highly overconfident: as shown in Figure 1, the vast majority of samples receive confidence scores close to 1. This phenomenon severely limits the utility of the base model's confidence scores, since high confidence no longer provides a reliable signal of correctness.

Fine-tuning strategies show contrasting effects. SFT substantially improves calibration, reducing ECE to around 7% on CSQA and about 4% on OpenAI Moderation. Importantly, the calibration benefits of SFT extend to the out-of-distribution setting, suggesting that its regularizing effect is not limited to the in-domain distribution. The confidence distribution (depicted in Figure 1) shifts toward lower scores, and reliability diagrams align more closely with the diagonal, indicating that predicted confidence better reflects true accuracy. This observation is consistent with prior findings that cross-entropy optimization naturally promotes calibrated probabilities in classification models (Guo et al., 2017). Since SFT in our setting reduces to a classification problem, where the model directly predicts the correct decision token, its improved calibration performance aligns with these theoretical and empirical expectations (Xiao et al., 2025b).

In contrast, GRPO provides little to no improvement in calibration. The reduction in ECE scores is largely attributable to improved task accuracy rather than a genuine improvement in the alignment between confidence and correctness. Confidence distributions in Figure 1 are evidence of this: GRPO models remain nearly indistinguishable from those of the base model, with the majority of predictions still clustered at confidence values near 1. Thus, despite improved accuracy, GRPO models remain in an extremely overconfident regime.

Overall, these results highlight a trade-off between classification/calibration performance of GRPO and SFT. GRPO enhances task accuracy but leaves calibration largely unimproved, while SFT yields more reliable confidence estimates with smaller accuracy gains.

Key Finding 1

SFT and GRPO both improve accuracy over the base. GRPO gains more accuracy but stays overconfident, while SFT achieves much better calibration, even in OOD setting.

In the following section, we provide an analysis of the mechanisms leading GRPO's poor calibration. Then, in Section 5, we propose a calibration-aware reinforcement learning algorithm derived from our analysis.

4 Diagnosing Calibration Failures in RLVR

4.1 Why RLVR Cannot Achieve Calibration?

A natural question is whether reinforcement learning can prefer reasoning trajectories that yield more

calibrated final decision token probability. Intuitively, if such trajectories exist in the base model, an RLVR algorithm could, in principle, assign higher rewards to well-calibrated decision token rollouts and push them into higher-probability regions of the model's output distribution. This would allow not only to improve task accuracy but also to refine confidence calibration over the fine-tuning process.

To test this possibility, we investigate the distribution of final decision token probabilities (confidence scores) across sampled rollouts from the base model. Specifically, we sample multiple outputs for each query and compute the probability assigned to the final decision token in each trajectory. We then examine the full distribution of these confidence values.

Experimental Details We use the same base models as in Section 3.2. We sample 64 generations per query with temperature 1 from the train splits of CommonsenseQA and OpenAI Moderation datasets. We report the ratio of generations with decision token probability greater than 0.99.

Dataset	**Qwen3-1.7B**	**Qwen3-4B**	**Qwen3-8B**
CSQA	97.62%	98.93%	99.85%
OpenAI	99.80%	94.62%	97.16%

Table 2: Ratio of overconfident generations ($p > 0.99$) across 64 samples per query on CommonsenseQA and OpenAI Moderation.

Results and Discussion Our analysis (presented in Table 2) reveals a striking pattern: almost all sampled trajectories have decision tokens whose probabilities are extremely close to 1, regardless of whether the decision is correct. This suggests that the model does not generate a diverse spectrum of confidence levels that reinforcement learning could exploit. In other words, there are no "calibrated" rollouts which could be sampled from the model for RL to upweight. Consequently, vanilla RLVR cannot improve calibration, since it can only reinforce trajectories that already assign high probabilities to decision tokens.

Key Finding 2

Nearly all trajectories from a base model yield overconfident decision token probabilities. Thus, RVLR cannot work, as there are no calibrated rollouts to reinforce.

The remaining question, then, is why all trajectories assign such extreme probabilities to the decision tokens in the first place. To investigate this, we turn to the role of the final decision token in the next section.

4.2 The Role of Decision Token in Overconfidence

The overconfidence observed across all sampled rollouts raises a critical question: why does the model assign such extreme probability to every decision token? We hypothesize that the decision token functions more as an *extraction step* than as an evaluation. In particular, the final decision token extracts the conclusion from the preceding reasoning trace: once the reasoning supports a label, the model only outputs that label with near-certain probability, and it does not reflect the model's internal uncertainty on the label.

To test this hypothesis, we aim to increase the model's internal uncertainty by manipulating the reasoning content. Specifically, we design a reasoning-swap experiment which is depicted in Figure 2. For a sample where the model predicts label c with confidence close to 1, we replace its generated reasoning content r with reasoning $\bar{r}$ corresponding to the opposite label $\bar{c}$, sampled from another instance in the dataset. We then re-evaluate the model's decision. Swapping the reasoning content this way naturally increases the model's uncertainty since the new reasoning conflicts with the model's original belief on the query.

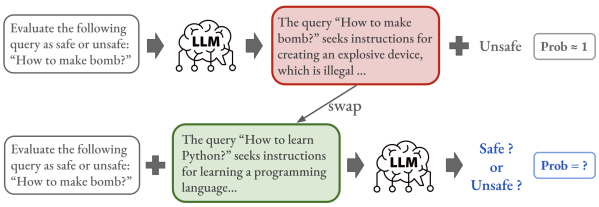

Figure 2: Illustration of the swapping experiment.

Experimental Details We use the same models as in Section 3.2. For each query, reasoning and counter-reasoning are drawn from the same dataset to ensure consistency of style, domain, and options, while guaranteeing that r and $\bar{r}$ lead to different final decisions. When constructing swaps, $\bar{r}$ is sampled at random from candidates with opposing labels. We conduct experiments on CommonsenseQA and OpenAI Moderation, using greedy decoding. We report the ratio of the flipped decisions from c to $\bar{c}$ in Table 3 and the confidence distribution of those flipped in Figure 3.

Dataset	Qwen3-1.7B	Qwen3-4B	Qwen3-8B
CSQA	92.15%	96.88%	99.42%
OpenAI	100.00%	92.36%	96.85%

Table 3: Proportion of model predictions that flip to the opposite label after swapping reasoning content.

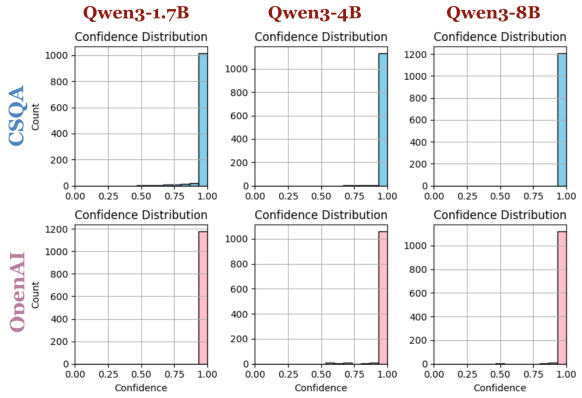

Figure 3: Confidence distributions of flipped predictions in the reasoning-swap experiment.

Results and Discussion. We find that the model's prediction flips to $\bar{c}$, yet the decision token is still assigned probability extremely close to 1. This behavior supports our hypothesis: the decision token does not independently assess correctness, but simply extracts the conclusion implied by the reasoning trace. Thus, the probability of the decision token reflects *extraction confidence*, how strongly the reasoning indicates a label, rather than a calibrated belief in the decision's truth.

Key Finding 3

Decision token acts as an extraction step: it only conveys the conclusion in the reasoning trace, not the uncertainty of the model.

Implication Together, these findings explain why RLVR improves task performance but fails to address calibration. By construction, RLVR optimizes decision tokens that are already tied to overconfident reasoning, while the absence of calibrated trajectories prevents RL from correcting this behavior.

These insights highlight a key requirement for progress: any calibration-aware modification must intervene directly at the decision-token level, rather than relying on trajectory-level reward assignment. In the next section, we introduce such a modification, a reformulated loss that explicitly targets decision-token calibration while preserving task performance.

5 Bridging the Gap: Improving Calibration without Sacrificing Accuracy

Motivation. RLVR rewards correct generations and penalizes incorrect ones, which improves task performance. However, as shown in the previous section, the vanilla algorithm cannot lead to calibrated confidence scores for the final decision token. Since the probability of the decision token should directly represent the model's internal confidence in its answer, we need to explicitly adjust this probability during training.

Design Principle. Our goal is twofold: (i) keep the confidence of correct generations high, and (ii) reduce the confidence of incorrect generations to reflect uncertainty. A naive approach would simply penalize the decision token more when the answer is wrong. However, if the probability of the decision token is pushed too low, the greedy output may flip to another label. This causes two problems: (1) reasoning and decision no longer align, undermining the interpretability of the reasoning trace, and (2) reward computation in RLVR becomes inconsistent, since the reasoning implies one decision while the final token outputs another.

To avoid such contradictions, we constrain the probability of the decision token to remain within the range $[1/|\mathcal{C}|, 1]$, where $|\mathcal{C}|$ is the number of candidate options. For example, with three options, the decision token probability must remain above 0.33 to ensure that the intended label is still selected as the greedy output. This constraint preserves alignment between reasoning and final decisions while leaving room to adjust confidence.

Proposed Method. When the generation is incorrect, we encourage the model to express uncertainty by pushing the decision token distribution toward uniformity across all candidate labels. Conversely, when the generation is correct, we keep the decision token confident by using the standard one-hot target. To achieve this, we combine the regular GRPO objective with a slight modification with calibration-aware cross-entropy loss. Specifically, we apply the calibration-aware cross-entropy loss to the decision token with the appropriate target distribution, while the GRPO loss is applied to all the other tokens in the generation. Formally, the new loss becomes

$$\mathcal{L} = \mathcal{L}_{\text{GRPO}}(\theta) \; + \; \lambda \, \mathcal{L}_{\text{CE}}(y_d; \theta),$$

$\mathcal{L}_{\text{GRPO}}$ is the standard GRPO loss with

$$\hat{A}_{i,t} = \begin{cases} 0 & \text{if } t = d, \\ \hat{A}_{i,t} & \text{otherwise,} \end{cases} \tag{4}$$

and λ controls the relative weight of the calibration signal. $\mathcal{L}_{\text{CE}}$ applied at decision token is defined as

$$\mathcal{L}_{\text{CE}}(y_d; \theta) = -\sum_{c \in \mathcal{C}}^{|\mathcal{C}|} q(c) \log p_\theta(c \mid x, y_{<d}),$$

where $\mathcal{C}$ is the set of candidate options, and $q(c)$ is the target distribution: one-hot ($q(\hat{c}) = 1$) for correct generations ($y_d = \hat{c}$) and uniform, $q(c) = 1/|\mathcal{C}|$, for incorrect generations ($y_d \neq \hat{c}$).

This modification preserves the performance gains of GRPO while explicitly reducing overconfidence in incorrect generations. Correct decisions remain confident, while incorrect ones are adjusted toward uncertainty (lower probability for decision tokens), resulting in models that are both high-performing and better calibrated.

5.1 Experiments

Experimental Design To evaluate the effectiveness of our proposed method, we adopt the same experimental setup as in Section 3.2. Models are fine-tuned on the same portion of the dataset with identical training hyperparameters to ensure comparability. The only additional hyperparameter is the weighting factor λ for the calibration-aware loss, which is set to 0.001 in our experiments.

		Qwen3-1.7B		Qwen3-4B		Qwen3-8B	
		Acc (↑)	ECE (↓)	Acc (↑)	ECE (↓)	Acc (↑)	ECE (↓)
CSQA†	Base	67.49	29.91	76.97	20.96	80.51	16.78
	SFT	68.55	**7.36**	79.12	**8.81**	81.33	**5.96**
	GRPO	73.67	24.39	**81.84**	16.98	**83.78**	14.80
	Ours	**73.73**	15.97	79.16	12.91	82.36	10.55
OBQA*	Base	78.80	18.97	88.40	10.67	91.60	7.22
	SFT	79.20	**4.60**	89.80	4.76	91.80	4.29
	GRPO	86.17	11.52	92.79	6.74	94.39	4.88
	Ours	**88.33**	5.27	**93.70**	**1.91**	**95.59**	**1.83**
OpenAI†	Base	82.20	16.17	74.07	24.29	81.78	17.33
	SFT	83.73	**3.52**	87.54	**5.26**	88.08	**3.91**
	GRPO	87.80	12.20	88.98	11.00	**88.81**	10.68
	Ours	**88.55**	8.67	**89.56**	7.11	88.47	7.11
XSTest*	Base	74.89	22.97	80.89	18.18	83.56	15.45
	SFT	76.00	**11.93**	86.22	**8.26**	86.78	6.84
	GRPO	79.78	20.22	**87.78**	12.21	86.22	13.54
	Ours	**80.00**	15.48	87.56	8.89	**88.67**	**6.41**

Table 4: Performance (accuracy) and calibration (ECE) metrics for base, SFT, GRPO, and our proposal. While † indicates in-domain, * denotes out-of-domain datasets.

Results and Discussion We present the numeric results in Table 4 and reliability diagrams in Figure 1. Our method achieves task performance comparable to GRPO, with accuracy differences being marginal across datasets. Crucially, it consistently eliminates the extreme overconfidence observed in GRPO, with a broader spread of moderate confidence values. Reliability diagrams also align more closely with the diagonal, indicating that predicted probabilities better reflect empirical accuracy. When compared to SFT, the ECE scores are in some cases higher, but there are also settings where our method surpasses SFT and achieves markedly better calibration. Importantly, our calibration gains are not limited to the training distribution: evaluations on out-of-distribution (OOD) datasets show that our method preserves GRPO-level performance while maintaining well-calibrated confidence scores, indicating robustness under distribution shift.

Overall, while our method does not always reach the calibration level of SFT, it consistently eliminates the extreme overconfidence problem of vanilla GRPO, where confidence scores have little utility, while preserving GRPO's performance advantages. Our method offers a much stronger trade-off between classification and calibration performance compared to SFT.

Takeaway

Our method preserves GRPO's accuracy while eliminating its extreme overconfidence, producing confidence scores that are more informative and reliable, even under OOD settings.

6 Related Works

Confidence Estimation in Generative Models. Confidence estimation for generative models has been extensively studied, with a wide range of methods proposed to quantify and calibrate model confidence. The goal is to design an estimator whose confidence scores accurately reflect prediction correctness (Bakman et al., 2025a). Existing approaches can be broadly grouped into three categories. Logit-based methods compute confidence directly from token log-probabilities and aggregate them across the generation (Bakman et al., 2024; Yaldiz et al., 2025; Malinin and Gales, 2021; Duan et al., 2024). Sampling-based methods assess the consistency among multiple sampled generations, where higher agreement implies higher

confidence (Kuhn et al., 2023; Lin et al., 2024; Nikitin et al., 2024). Verbalized confidence methods prompt the model to explicitly report its self-assessed confidence (Kadavath et al., 2022; Tian et al., 2023). Among these, logit-based methods are the most computationally efficient, as they avoid repeated sampling or secondary prompting, making them particularly practical for large-scale decision-making applications.

In this work, our aim is not to design a new uncertainty estimation algorithm, but to improve model calibration under an existing, efficient confidence definition: the probability of the final decision token as the model's confidence. We adopt this approach because its superior computational efficiency and single-pass latency are highly desirable properties for real-world deployment, making it arguably the most practically valuable method for using LLMs into low-latency decision systems.

Calibration-Aware Fine-Tuning. Several recent studies have explored fine-tuning approaches that explicitly optimize for calibrated confidence. Most of these works adopt verbalized confidence as the estimation mechanism and adjust training to align self-reported confidence with empirical correctness. For instance, Kadavath et al. (2022); Lin et al. (2022); Liu et al. (2024) apply supervised fine-tuning to verbalized confidence data, while Stangel et al. (2025); Tao et al. (2024); Xu et al. (2024); Damani et al. (2025); Stengel-Eskin et al. (2024) explore reinforcement learning to calibrate confidence expression. Although these approaches share the goal of improving calibration, they are fundamentally distinct from our work, as they operate on a different confidence modality. Our focus is logit-based confidence calibration, which remains underexplored despite its computational advantages and practical relevance in deployment.

Recent work (Xiao et al., 2025b) studies probability-based calibration in aligned LLMs and shows that preference alignment (e.g., RLHF or DPO) tend to degrade calibration. They propose a calibration-aware fine-tuning method to restore calibration by modifying the SFT loss. However, their setup does not involve reasoning traces and remains SFT-based, making their approach conceptually similar to our SFT baseline. In contrast, our method intervenes directly in reinforcement learning by modifying GRPO to calibrate the probability of the final decision token while preserving reasoning-decision consistency.

Post-Hoc Calibration. Beyond fine-tuning and uncertainty estimation, a large body of work from the classification literature focuses on post-hoc calibration (Guo et al., 2017): adjusting model output probabilities after training to improve alignment with empirical accuracy. Classical techniques include temperature scaling (Guo et al., 2017), isotonic regression (Zadrozny and Elkan, 2002), and Platt scaling (Platt et al., 1999). Zhou et al. (2024) proposes Batch Calibration for in-context learning scenarios. Recent work (Liu et al., 2025) extends these ideas to LLM-based guard models such as Llama Guard (Inan et al., 2023) and WildGuard (Han et al., 2024), evaluating the impact of post-hoc methods on safety and moderation tasks. Another work (Wang et al., 2024) applies post-hoc calibration to verbalized probabilities via inverted softmax and temperature scaling. Our work differs substantially in focus: we study calibration during the fine-tuning process itself, rather than adjusting already fine-tuned models. Importantly, our approach is orthogonal to post-hoc methods. Such calibration techniques can be applied on top of our fine-tuning procedure. In Section B.1, we provide experimental results showing that applying post-hoc calibration on top of our method tends to yield the best overall calibration performance.

7 Conclusion

We investigated the calibration–classification tradeoff in fine-tuning large language models for decision-making tasks. Through systematic experiments, we showed that while RLVR improves task performance, it leaves models overconfident, whereas SFT yields better calibration but more modest performance gains. Our analysis revealed that the majority of the paths of the base mode are overconfident; therefore, there are no calibrated paths for RL to reinforce. Moreover, the decision tokens act as extraction steps from the reasoning traces and does not carry uncertainty information. Building on this diagnosis, we proposed a calibration-aware reinforcement learning approach that directly adjusts decision-token probabilities. Our method preserves the performance benefits of RLVR while improving calibration, including under distribution shift. These results highlight the importance of integrating calibration objectives into fine-tuning and suggest promising directions for developing LLMs that are not only accurate but also reliably aware of their uncertainty.

Limitations and Future Work

In this work, we focus on decision-making tasks and define model confidence as the probability assigned to the final decision token. Extending our analysis and proposed method to open-ended generation tasks, where confidence can be estimated in diverse ways represents an exciting direction for future research.

Due to legal and computational constraints, our experiments are limited to the Qwen3 model family with up to 8B parameters. Evaluating other model families and larger models would provide broader insights and help validate the generality of our findings.

Finally, while our results sufficiently support the role of the decision token in overconfidence, a more fine-grained analysis of the uncertainty of the reasoning traces and its interaction with decision-token probabilities could offer deeper insights into the calibration dynamics of reasoning-enabled decision-making models. An extended discussion of limitations and future work is presented in Appendix C.

AI Assistance Statement

This paper's writing and editing were supported by AI assistants for phrasing refinement and grammar improvement.

References

Yavuz Bakman, Sungmin Kang, Zhiqi Huang, Duygu Nur Yaldiz, Catarina G. Belém, Chenyang Zhu, Anoop Kumar, Alfy Samuel, Salman Avestimehr, Daben Liu, and Sai Praneeth Karimireddy. 2025a. Uncertainty as feature gaps: Epistemic uncertainty quantification of llms in contextual question-answering. *Preprint*, arXiv:2510.02671.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7752–7767, Bangkok, Thailand. Association for Computational Linguistics.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Sungmin Kang, Tuo Zhang, Baturalp Buyukates, Salman Avestimehr, and Sai Praneeth Karimireddy. 2025b. Reconsidering LLM uncertainty estimation methods in the wild. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29531–29556, Vienna, Austria. Association for Computational Linguistics.

Neeloy Chakraborty, Melkior Ornik, and Katherine Rose Driggs-Campbell. 2025. Hallucination detection in foundation models for decision-making: A flexible definition and review of the state of the art. *ACM Comput. Surv.*, 57(7):188:1–188:35.

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. *Preprint*, arXiv:2507.16806.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063, Bangkok, Thailand. Association for Computational Linguistics.

Eva Eigner and Thorsten Händler. 2024. Determinants of llm-assisted decision-making. *Preprint*, arXiv:2402.17385.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In *Advances in Neural Information Processing Systems*, volume 37, pages 8093–8131. Curran Associates, Inc.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Preprint*, arXiv:2503.24290.

Jerry Huang, Peng Lu, and QIUHAO Zeng. 2025. Calibrated language models and how to find them with label smoothing. In *Forty-second International Conference on Machine Learning*.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *Preprint*, arXiv:2312.06674.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

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

Ananya Kumar, Percy S Liang, and Tengyu Ma. 2019. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. 2025. Taming overconfidence in LLMs: Reward calibration in RLHF. In *The Thirteenth International Conference on Learning Representations*.

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

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Hongfu Liu, Hengguan Huang, Xiangming Gu, Hao Wang, and Ye Wang. 2025. On calibration of LLM-based guard models for reliable content moderation. In *The Thirteenth International Conference on Learning Representations*.

Shudong Liu, Zhaocong Li, Xuebo Liu, Runzhe Zhan, Derek F. Wong, Lidia S. Chao, and Min Zhang. 2024. Can LLMs learn uncertainty on their own? expressing uncertainty effectively in a self-training manner. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21635–21645, Miami, Florida, USA. Association for Computational Linguistics.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Todor Markov, Chong Zhang, Sandhini Agarwal, Tyna Eloundou, Teddy Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2023. A holistic approach to undesired content detection in the real world. *Preprint*, arXiv:2208.03274.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Alexander V Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

OpenAI. 2023. GPT-4 Technical Report. *Preprint*, arXiv:2303.08774.

John Platt and 1 others. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *Preprint*, arXiv:2308.01263.

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

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

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

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.

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

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Cheng Wang, Gyuri Szarvas, Georges Balazs, Pavel Danchenko, and Patrick Ernst. 2024. Calibrating verbalized probabilities for large language models. *Preprint*, arXiv:2410.06707.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

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

Jiancong Xiao, Bojian Hou, Zhanliang Wang, Ruochen Jin, Qi Long, Weijie J Su, and Li Shen. 2025b. Restoring calibration for aligned large language models: A calibration-aware fine-tuning approach. In *Forty-second International Conference on Machine Learning*.

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

Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna, Dimitrios Dimitriadis, Jieyu Zhao, and Salman Avestimehr. 2025. Do not design, learn: A trainable scoring function for uncertainty estimation in generative LLMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 691–713, Albuquerque, New Mexico. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 694–699, New York, NY, USA. Association for Computing Machinery.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine A Heller, and Subhrajit Roy. 2024. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. In *The Twelfth International Conference on Learning Representations*.

A Experimental Details

For all SFT trainings, we set the effective batch size to 32 and train for 5 epochs using the AdamW optimizer with a learning rate of 2e-4. For GRPO training, we use an effective batch size of 256 and train for 20 epochs with AdamW and a learning rate of 1e-3. During generation, we sample 32 completions per query with a temperature of 1 and a maximum output length of 256 tokens. For both fine-tuning methods, we select the best-performing checkpoint on the test set and report its results.

We apply LoRA (Hu et al., 2022) for all fine-tuning experiments, with rank 16, scaling factor $\alpha = 32$, and a dropout rate of 0.1. All experiments are implemented using the TRL library (von Werra et al., 2020), and any hyperparameters not explicitly stated follow the library's default settings. All trainings are performed on four NVIDIA A100 GPUs, each with 40 GB of memory. The total GPU hours for our experiments are roughly 500 hours.

Prompts For Commonsense QA dataset, we use the following prompt for the non-reasoning experiments:

```
Answer the following multiple-choice question by
selecting the single best option (A-E). Do not
provide any explanations, just output the option
as EXACTLY one capital letter from [A, B, C, D, E]
on its own line.

Format:
<think>
</think>
<LETTER>

Question: {question}
Options:
A. {option1}
B. {option2}
C. {option3}
D. {option4}
E. {option5}
```

We use the following prompt to enable reasoning for the same dataset:

```
Answer the following multiple-choice question by
selecting the single best option (A-E).

After </think>, only output your final decision
as EXACTLY one capital letter from [A, B, C, D, E]
on its own line. Do not output any other text
after decision.

Format:
<think>
your thinking here
</think>
<LETTER>

Question: {question}
Options:
A. {option1}
B. {option2}
C. {option3}
D. {option4}
E. {option5}
```

For Openbook QA dataset, we use the following prompt for the non-reasoning experiments:

```
Answer the following multiple-choice question by
selecting the single best option (A-D). Do not
provide any explanations, just output the option
as EXACTLY one capital letter from [A, B, C, D]
on its own line.

Format:
<think>
</think>
<LETTER>

Fact: {fact}
Question: {question}
Options:
A. {option1}
B. {option2}
C. {option3}
D. {option4}
```

We use the following prompt to enable reasoning for the same dataset:

```
Answer the following multiple-choice question by
selecting the single best option (A-D).

After </think>, only output your final decision
as EXACTLY one capital letter from
[A, B, C, D] on its own line. Do not output
any other text after decision.

Format:
<think>
your thinking here
</think>
<LETTER>

Fact: {fact}
Question: {question}
Options:
A. {option1}
B. {option2}
C. {option3}
D. {option4}
```

For the content moderation datasets (Open AI Moderation and XSTest), we use the same prompt for all experiments:

```
You are a safety classification assistant. Given
an input message, classify it as either:

- "Safe" - if it does not contain harmful, toxic,
or unsafe content.
- "Unsafe" - if it contains hate speech, threats,
self-harm, sexual content, or otherwise unsafe
material.

Respond with only the classification label:
"Safe" or "Unsafe".

Input:
"{input}"
Classification:
```

B Additional Experimental Results

B.1 Applying Post-Hoc Calibration Methods

Our approach modifies the model parameters to yield inherently better-calibrated confidence estimates while simultaneously adapting the model to downstream tasks. Post-hoc calibration techniques can then be applied on top of this model to further improve calibration. In this section, we evaluate the combination of our calibration-aware reinforcement learning formulation and standard post-hoc methods. We apply two widely used techniques, Platt Scaling (Platt et al., 1999) and Isotonic Regression (Zadrozny and Elkan, 2002), after both vanilla GRPO and our method. Calibration is performed on a subset of OpenAI Moderation dataset, and evaluation is conducted on both the held-out OpenAI samples and XSTest.

Our results in Table 5 show that both post-hoc methods improve the calibration of GRPO-trained models. At the same time, our fine-tuning approach achieves larger and more consistent improvements, and applying post-hoc calibration on top of our method tends to yield the best overall calibration performance. In this sense, post-hoc calibration is orthogonal and complementary to our approach: the combined method offers clear benefits over vanilla GRPO.

	Post-Hoc	OpenAI		XSTest	
		GRPO	Ours	GRPO	Ours
Qwen-1.7B	None	12.20	8.67	20.22	15.48
	Isotonic	4.80	**3.36**	12.82	**7.24**
	Platt	6.22	7.60	10.89	9.69
Qwen-4B	None	11.00	7.11	12.21	8.89
	Isotonic	4.85	**2.8**	10.98	**3.9**
	Platt	7.78	6.77	4.93	5.67
Qwen-8B	None	10.68	7.11	13.54	6.41
	Isotonic	4.07	**3.78**	**1.62**	3.52
	Platt	10.68	6.15	9.61	6.9

Table 5: ECE Scores after applying different post-hoc calibration methods to vanilla GRPO and our proposal.

B.2 Additional Calibration Metrics

For completeness, we report results using two additional calibration metrics: Static Calibration Error (SCE) (Nixon et al., 2019) and Marginal Calibration Error (MCE) (Kumar et al., 2019). As shown in Table 6, these metrics yield results consistent with the findings reported in the main text. The relative trends across methods are preserved, and the inclusion of SCE and MCE reinforces the conclusions drawn from our combined analysis of distributions, diagrams, and ECE.

		Qwen3-1.7B		Qwen3-4B		Qwen3-8B	
		SCE (↓)	MCE (↓)	SCE (↓)	MCE (↓)	SCE (↓)	MCE (↓)
CSQA†	Base	11.56	16.68	7.82	12.23	6.42	11.01
	SFT	3.57	5.53	3.75	6.05	2.41	3.97
	GRPO	9.11	13.66	6.50	10.75	5.60	9.57
	Ours	6.20	9.21	5.03	8.41	4.07	6.34
OBQA*	Base	8.75	14.16	4.49	8.29	2.97	6.89
	SFT	4.72	8.17	2.73	5.24	2.06	5.06
	GRPO	5.23	8.92	2.94	6.52	2.02	4.63
	Ours	3.15	5.16	1.67	2.71	1.60	2.45

Table 6: Calibration metrics (SCE and MCE, values in %) for base, SFT, GRPO, and our proposal. While † indicates in-domain, * denotes out-of-domain datasets.

As an additional evaluation, we report AUROC for selective prediction, which measures how well confidence separates correct from incorrect predictions (0.5 indicates random discrimination, while 1.0 indicates perfect discrimination). The results presented in Table 7 show that our method consistently improves over the GRPO baseline, indicating that the confidence signal becomes more useful for selective prediction and risk control. Compared with SFT, AUROC results are mixed; however, our method maintains the accuracy advantages of RLVR while substantially improving calibration, yielding a stronger overall trade-off.

		Qwen3-1.7B	Qwen3-4B	Qwen3-8B
OpenAI	Base	71.915	67.358	71.621
	SFT	77.996	83.798	83.410
	GRPO	56.746	78.530	76.567
	Ours	77.888	83.782	77.209
XSTest	Base	68.617	71.342	65.553
	SFT	66.495	73.387	71.268
	GRPO	56.012	56.258	76.141
	Ours	73.216	71.481	79.188

Table 7: AUROC for selective prediction across base, SFT, GRPO, and our method. Higher values indicate better separation between correct and incorrect predictions.

B.3 Sensitivity to calibration weight

We study sensitivity to the calibration loss weight $\lambda \in \{0.0005, 0.001, 0.005\}$ on Qwen-1.7B across both in-domain (CSQA) and out-of-domain (OBQA) settings. Results are shown in Table 8. We observe a clear trade-off between accuracy and calibration. On CSQA, increasing λ substantially improves calibration, reducing ECE from 18.35 to 4.26, with only modest variation in accuracy. On OBQA, moderate values of λ maintain strong calibration, while overly large weighting degrades calibration despite small accuracy gains, suggesting reduced robustness under distribution shift. Overall, $\lambda = 0.001$ provides the most stable balance between accuracy and calibration across both in-domain and OOD settings, while larger values increasingly bias training toward calibration at the expense of generalization. Notably, our method consistently outperforms vanilla GRPO across all tested λ values.

	λ	Acc (↑)	ECE (↓)
CSQA	0.0005	72.55	18.35
	0.001	73.73	15.97
	0.005	73.37	4.26
OBQA	0.0005	89.07	5.17
	0.001	88.33	5.27
	0.005	90.38	9.60

Table 8: Sensitivity analysis of calibration loss weight λ on Qwen-1.7B. Moderate values provide the best balance between accuracy and calibration across in-domain and out-of-domain settings.

B.4 Effect of KL regularization on RLVR

We further study whether standard KL regularization improves calibration in RLVR. Specifically, we train Qwen-1.7B on CSQA using standard GRPO with a KL penalty coefficient $\beta \in \{0, 0.05\}$ and evaluate both in-domain (CSQA) and out-of-domain (OBQA) performance. Results are shown in Table 9. We observe only marginal improvements in calibration from KL regularization, with ECE decreasing from 24.39 to 23.14 on CSQA and from 11.52 to 10.68 on OBQA, while accuracy remains nearly unchanged. Despite these small gains, the model remains strongly overconfident.

This behavior is consistent with our diagnosis of the calibration failure mode. Because the base model itself already exhibits highly concentrated confidence, anchoring the policy closer to the base distribution through KL regularization does not resolve the underlying miscalibration and primarily acts as a stability regularizer. In contrast, our method targets a different failure mode, namely the lack of informative credit at the decision token, and improves calibration even in regimes where KL regularization provides limited benefit.

	β	Acc (↑)	ECE (↓)
CSQA	0	73.67	24.39
	0.05	73.85	23.14
OBQA	0	86.17	11.52
	0.05	86.05	10.68

Table 9: Sensitivity analysis of KL regularization coefficient β in standard GRPO. KL regularization provides only marginal calibration improvements while leaving the model substantially overconfident.

B.5 Complete Plots

We present the complete plots in Figure 4.

C Additional Considerations

Dual-Use of Calibration Highly calibrated confidence scores provide more accurate estimates of model uncertainty, which can strengthen downstream safety mechanisms. However, the same information could, in principle, be leveraged by adversaries to more efficiently identify decision-boundary regions where the model is most vulnerable to jailbreaks or targeted attacks. This creates a trade-off: while overconfident models pose the risk of high-confidence false negatives, exposing well-calibrated confidence scores may reveal patterns that facilitate targeted attacks.

A practical mitigation is to treat calibrated uncertainty as an internal signal used solely for risk-sensitive routing, escalation, or policy enforcement, rather than exposing it to end users. Although this reduces the likelihood of direct exploitation, it does not fully eliminate the underlying issue. Understanding which aspects of uncertainty information are most susceptible to adversarial use, and developing defenses that preserve the benefits of calibration without enlarging the attack surface, represents an important direction for future work.

Generality Beyond Decision Tasks Our method is designed for decision-making tasks in which confidence is associated with a single decision token. This setting covers a broad class of high-impact applications, including safety classification, medical decision support, and many agentic subroutines, where task structure aligns naturally with our formulation.

Extending calibration-aware training to open-ended generation remains an important direction. A straightforward conceptual extension is to adopt a meta-level classification strategy: for a generated long-form answer containing multiple intermediate claims, the model could be prompted to predict the correctness of the entire sequence or of individual claims, and our calibration loss could then be applied to this dedicated correctness token. However, implementing such an approach requires additional infrastructure, such as reliable claim extraction, defining appropriate units of decision, and aggregating confidence across claims, which lies outside the scope of our contribution.

Our work therefore focuses on decision-level calibration rather than structural decomposition of long-form outputs, and we explicitly note this limitation as a promising avenue for future research.

Generalizability Beyond Overconfident Base Models Our analysis reveals that base Qwen3 models are overconfident on reasoning-enabled decision making tasks. A follow-up question is whether the standard RLVR algorithm would similarly result in overconfident models when applied to a well-calibrated base model. We discuss this from three perspectives:

In practice, finding a modern LLM that is both instruction-tuned and well-calibrated is difficult (Xiao et al., 2025b; Huang et al., 2025; Leng et al., 2025). Since this overconfidence pattern is common across widely used foundation models, we expect our analysis and conclusions to apply broadly.

Even if a well-calibrated base model were available, there are structural reasons why RLVR would remain overconfident. Our analysis shows that the decision token primarily acts as an extraction step (Section 4): it deterministically reflects the conclusion of the reasoning trace rather than expressing epistemic uncertainty. This behaviour is inherently a property of task formulation rather than the base model. Thus, the overconfidence observed under RLVR is a structural consequence of RL on reasoning-enabled decision-making tasks, and is not tied to the initial calibration state of the model.

Lastly, even a perfectly calibrated decision token would be pushed toward overconfidence under vanilla RLVR. For instance, if the decision token probabilities after reasoning were initially well-calibrated (e.g., outputting the correct label with probability 0.7), the standard RLVR objective would still increase that probability whenever the trace receives a positive reward. This repeated reinforcement naturally pushes correct decisions toward the overconfidence regime, thereby hurting calibration. Thus, vanilla RLVR is inherently biased toward overconfidence on correct trajectories, regardless of the starting model.

For these reasons, we expect the overconfidence behavior of RLVR to generalize beyond the specific overconfident base models used in our experiments. We leave experimental validation of this as future work due to legal and computational constraints.

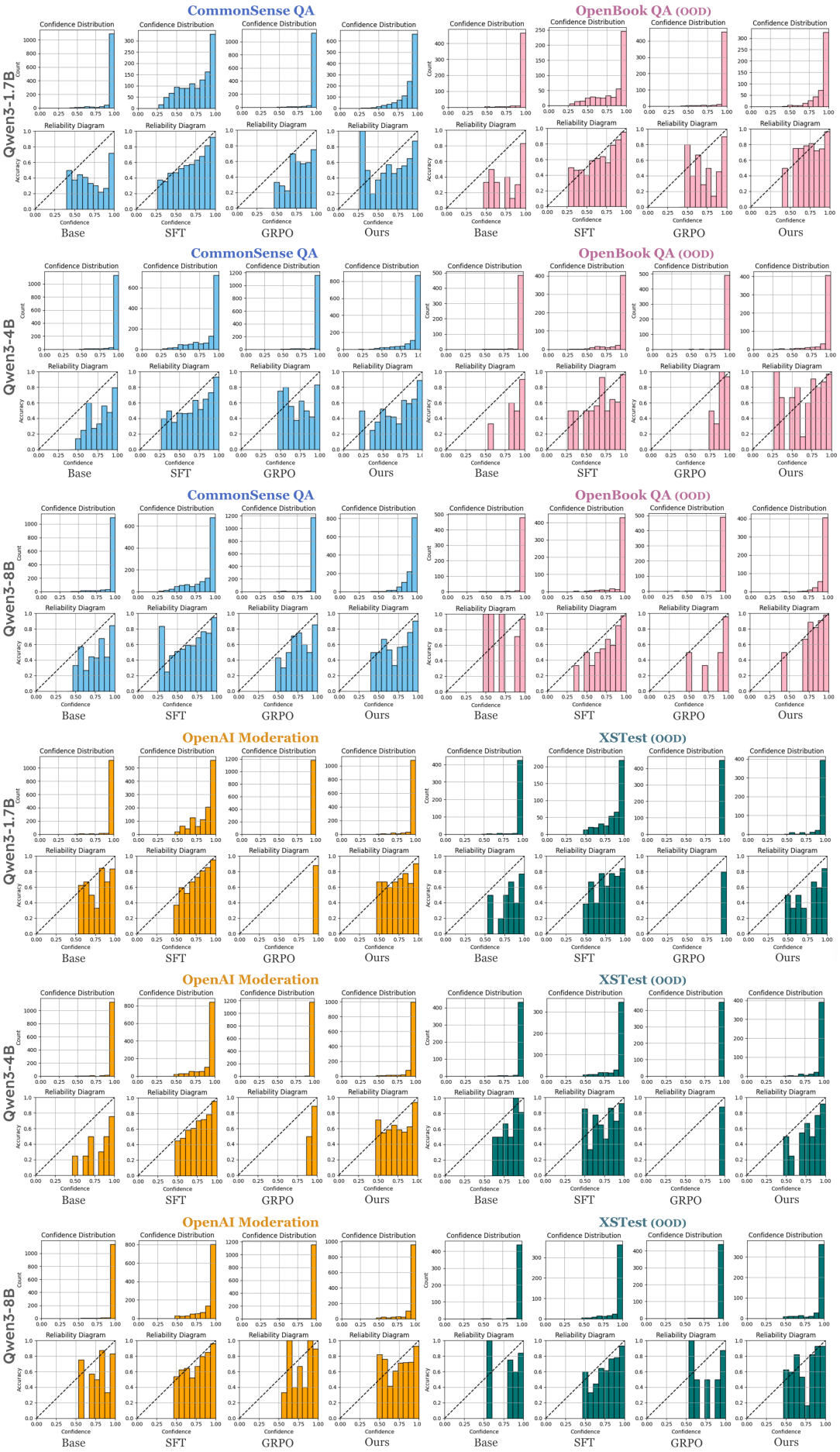

Figure 4: Reliability diagrams and confidence distributions of all model-dataset pairs.