Aligning Language Models with Selective Prediction

Gaoxiang Luo [1] **Yifan Wu** [2] **Sinian Zhang** [3] **Aryan Deshwal** [1] **Ju Sun** [1]

https://sun-umn.github.io/RLSR

Abstract

Large language models (LLMs) are increasingly deployed as critical decision-making components in high-stakes real-world AI systems, rendering LLM reliability a foremost practical concern. In this paper, we focus on enhancing LLM reliability through selective prediction (SP), a strategy that allows an LLM to only predict for inputs where it is likely to be correct (i.e., *coverage*) and hence reduce the error rate (i.e., *risk*) on that portion of inputs—flagging the remaining inputs for future human discretion. In other words, SP improves LLM reliability by balancing the risk-coverage trade-off and enabling seamless human-AI collaboration. To integrate SP into LLMs, we focus on the LLM post-training alignment stage and propose to align LLMs with SP performance metrics, in contrast with existing LLM alignment methods that focus primarily on correctness or calibration metrics. Specifically, we propose a novel alignment framework, Reinforcement Learning for Selection Reward (RLSR), which targets the area under the risk-coverage curve (AURC)—a popular SP performance metric—as its alignment objective. RLSR achieves substantially better risk-coverage trade-off compared to multiple alignment baselines on both in-domain and out-of-domain tasks.

1. Introduction

In high-stakes applications such as finance, healthcare and law (Lai et al., 2024; Fan et al., 2025; Gaber et al., 2025), LLMs are increasingly used to make decisions. In these settings, the cost of making a mistake significantly outweighs that of deferring the case to a human. Thus, it is reasonable to allow and encourage the LLM to abstain when its answer is likely to be wrong. This is the goal of **selective prediction (SP)**, where the LLM answers only a selected subset of queries that it is likely to get right, to keep its error rate acceptably low on that subset. This paradigm stands in stark contrast to standard LLM training and evaluation, which requires LLMs to answer regardless of their capabilities, causing confident mistakes and hallucinations (Kalai et al., 2026). In this work, we propose aligning LLMs directly with SP in a principled manner.

Confidence calibration (CC) is not sufficient or necessary for selective prediction (SP). SP is often assumed, *explicitly or implicitly*, to be the logical next step after confidence calibration (CC) (Guo et al., 2017; Nixon et al., 2019; Fisch et al., 2022; Liang et al., 2024a). Indeed, assume that the prediction confidence is perfectly calibrated (i.e., aligned with the true posterior probability $p(y|x)$); then a selection threshold at 0.8 ensures that the error rate on the selected subset is less than $1 - 0.8 = 0.2$. While this is technically elegant and convenient, CC and SP are still disparate: CC stresses that the assigned confidence be consistent with the likelihood of correctness, whereas SP favors that correct predictions are ranked ahead of wrong ones—in terms of assigned confidence. As shown in Fig. 1 (Left), *perfect CC might not lead to perfect SP*, as any threshold we draw can still lead to accepted wrong answers and rejected right answers; conversely, as shown in Fig. 1 (Right), *perfect SP can be based on ill-calibrated confidence*, so long as all right answers are ranked ahead of wrong ones. Their difference is also reflected by the different evaluation metrics used in practice: CC is often measured by the *expected calibration error* (ECE) or variants, whereas SP by the *area under the risk-coverage curve* (AURC) or variants. Thus, *if the goal is SP, it is technically sound to pursue it directly instead of through CC*.

Aligning LLMs with selective prediction. In this paper, we propose aligning pretrained LLMs directly with SP. To this end, we adopt the popular Reinforcement Learning with Verifiable Rewards (RLVR) (Shao et al., 2024) align-

[1]Department of Computer Science and Engineering [2]Bioinformatics and Computational Biology Program [3]Division of Biostatistics and Health Data Science, University of Minnesota Twin Cities, Minneapolis, MN, USA. Correspondence to: Gaoxiang Luo <luo00042@umn.edu>, Ju Sun <jusun@umn.edu>.

Published at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Copyright 2026 by the author(s).

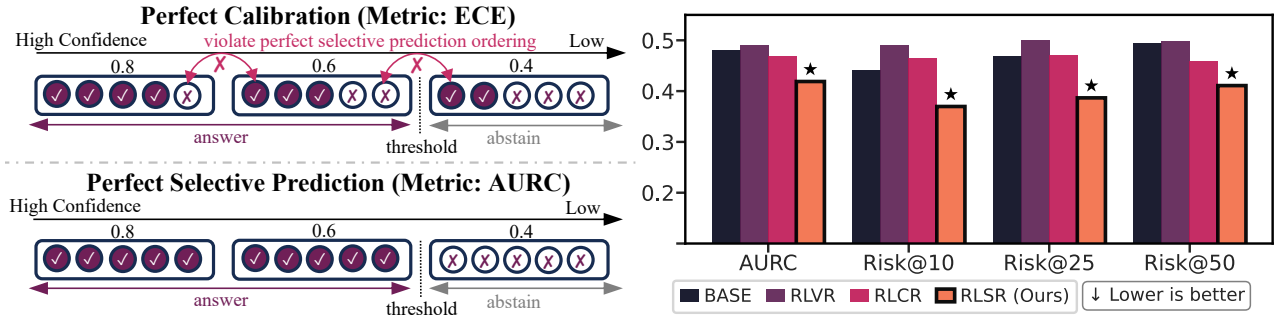

Figure 1. **(Left)** Perfect confidence calibration does not translate to perfect selective prediction, and vise versa; **(Right)** Our RLSR aligns LLMs with selective prediction (SP) and substantially improves upon competing methods in terms of SP metrics: AURC and Risk@ k.

ment framework, which uses verifiable rewards to improve LLM accuracy and reasoning. The recent work, Reinforcement Learning with Calibration Rewards (RLCR) (Damani et al., 2026), proposes augmenting the correctness reward in RLVR with CC-aware rewards (e.g., the Brier score) to promote LLM accuracy and CC. However, it does not necessarily optimize SP for LLMs, due to the misalignment of CC and SP discussed above. Thus, we propose replacing the correctness reward with SP-aware rewards. Toward this, the AURC, a standard SP metric that summarizes the whole spectrum of risk-coverage tradeoffs, seems a natural choice. However, we face two implementation and performance challenges: (1) the canonical AURC only penalizes errors but does not reward correctness, impairing learning; (2) AURC is a population-level metric that depends on the global ranking of sample confidence, not friendly for mini-batch stochastic gradient methods.

Our contributions. In this paper, we propose **Reinforcement Learning for Selection Reward (RLSR)** to directly align pretrained LLMs with SP. To tackle the challenges identified above, we develop a lifted AURC objective, and a batch training strategy based on a judicious choice of AURC reformulation. Our key contributions are:

1. We propose aligning pretrained LLMs with selection prediction (SP) to enhance their reliability for deployment. To the best of our knowledge, this is *the first work* proposing this;
2. We introduce RLSR, a new LLM alignment framework with an SP-aware reward. We base our learning algorithm on the Group Relative Policy Optimization (GRPO) framework (Shao et al., 2024) and develop a novel lifted AURC objective and a batch training strategy to tackle the key technical challenges in optimizing the AURC metric (Sec. 2.2);
3. We empirically demonstrate that RLSR outperforms strong baselines, including accuracy-oriented RLVR and calibration-oriented RLCR, achieving strong SP performance on multiple in-domain and out-of-domain tasks (Fig. 1 (Right), Secs. 4.1 and 4.2).

2. Method

2.1. Technical Background

LLM alignment via RL. Let $\mathcal{X}$ and $\mathcal{Y}$ be the prompt and response spaces, respectively, and $\mathcal{D}$ be the data distribution on $\mathcal{X} \times \mathcal{Y}$. Given a stochastic LLM π_θ that maps prompts to responses, a typical LLM alignment step is to improve LLM responses by optimizing

$$\max_\theta J(\theta) \doteq \mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbb{E}_{\widehat{y}\sim\pi_\theta(\cdot|x)} R(y, \widehat{y}), \tag{2.1}$$

where $\pi_\theta(\cdot|x)$ denotes the distribution on the response space $\mathcal{Y}$ conditioned on a prompt x—not a single response because π_θ is stochastic, and $R : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a chosen **reward function**. In other words, LLM alignment is to finetune the LLM parameters θ by maximizing the expected reward. To maximize using gradient ascent, one can apply the celebrated log-derivative trick, which is popularly used in policy gradient methods for reinforcement learning (RL), to derive the expected gradient as (see Sec. A.1 for details)

$$\nabla_\theta J(\theta) = \mathbb{E}_{(x,y)}\mathbb{E}_{\widehat{y}}\left[\nabla_\theta \log \pi_\theta(\widehat{y}|x) R(y, \widehat{y})\right], \tag{2.2}$$

where we omit the $\mathcal{D}$ and $\pi_\theta(\cdot|x)$ under expectation signs to simplify the notation. In practice, the expected gradient is approximated by mini-batch gradient: one estimates the gradient by sampling a batch of (x, y) pairs from $\mathcal{D}$, and for each x sampling a batch of $\widehat{y}$, i.e., rollouts, from $\pi_\theta(\cdot|x)$.

In practice, various techniques have been proposed to improve the performance of the above framework. We follow the popular **GRPO** algorithm (Shao et al., 2024), which is essentially the classic **proximal policy optimization (PPO)** in RL with a twist on the advantage function. PPO changes the optimization objective in Eq. (2.1) in three aspects: **(1) rewards into group-relative advantages** by replacing $R(y, \widehat{y})$ with prompt-level advantage $A(y, \widehat{y})$ that gives more advantages to "breakthrough" responses that stand out from the rest; **(2) per-iteration surrogation** by approximating the objective by

$$J(\theta; \theta_{\text{old}}) \doteq \mathbb{E}_{(x,y)}\mathbb{E}_{\widehat{y}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \frac{\pi_\theta(\widehat{y}|x)}{\pi_{\theta_{\text{old}}}(\widehat{y}|x)} A(y, \widehat{y}), \tag{2.3}$$

where θ_{old} denotes the model parameters from the previous iteration. This helps decouple response sampling from an old model $\pi_{\theta_{\text{old}}}$ and gradient estimation for the current model π_θ; **(3) range clipping on density ratios** to stabilize training by modifying Eq. (2.3) as

$$r_{x,\widehat{y}} \to \min(1+\varepsilon, r_{x,\widehat{y}})\mathbb{I}\{A(y,\widehat{y})>0\} + \max(1-\varepsilon, r_{x,\widehat{y}})(1-\mathbb{I}\{A(y,\widehat{y})>0\}) \quad (2.4)$$

where $r_{x,\widehat{y}} \doteq \pi_\theta(\widehat{y}|x)/\pi_{\theta_{\text{old}}}(\widehat{y}|x)$, i.e., avoid excessively positive and negative advantages so that the update to θ becomes less biased toward any particular $(x,\widehat{y})$ pair. GPRO, as a variant of PPO, works with the special advantage function

$$A_{\text{GRPO}}(y,\widehat{y}_i) = \frac{R(y,\widehat{y}_i) - \text{mean}\,\{R(y,\widehat{y}_j)\}_{j=1}^G}{\text{std}\,\{R(y,\widehat{y}_j)\}_{j=1}^G}, \quad (2.5)$$

where j indexes the sampled responses (i.e., rollouts) for the same prompt. This advantage function effectively performs prompt-level zero-mean-unit-variance normalization to the rewards. The gradient of the modified objective in GRPO can be derived using the log-derivative trick, analogous to the process for Eq. (2.1).

RLVR & RLCR. RLVR has recently emerged as the gold-standard LLM alignment framework for mathematical and coding tasks (Guo et al., 2025; Wen et al., 2025). It typically assumes an external programmatic verifier to judge the correctness of the response and gives a binary reward

$$R_{\text{RLVR}}(y,\widehat{y}) \doteq \mathbb{I}\{y=\widehat{y}\}. \quad (2.6)$$

After RLVR training, recent reasoning models often return verbalized confidence (Kumaran et al., 2026; Xiong et al., 2024) alongside their responses, explicitly communicating the model's internal uncertainty to the user (see Sec. C.1). Although R_{RLVR} explicitly incentivizes accurate prediction, RLVR-finetuned LLMs are often found to be overconfident, i.e., not well calibrated (Leng et al., 2025). To promote both correct prediction and calibrated confidence, RLCR (Damani et al., 2026) proposes adding a Brier score penalty to the RLVR reward, leading to

$$R_{\text{RLCR}}(y,\widehat{y},s) \doteq \mathbb{I}\{y=\widehat{y}\} - (s - \mathbb{I}\{y=\widehat{y}\})^2, \quad (2.7)$$

where s denotes the verbalized confidence returned by the LLM alongside $\widehat{y}$.

Selective prediction (SP). For a general input-output space $\mathcal{X}' \times \mathcal{Y}'$ and an associated data distribution $\mathcal{D}'$ on $\mathcal{X}' \times \mathcal{Y}'$, SP augments a predictive model $f: \mathcal{X}' \to \mathcal{Y}'$ with a binary selector $g: \mathcal{X}' \to \{0,1\}$, so that the final output by the pair (f,g) is:

$$(f,g)(x') = \begin{cases} f(x') & \text{if } g(x')=1, \\ \text{abstain} & \text{if } g(x')=0 \end{cases} \quad \forall x' \in \mathcal{X}'. \quad (2.8)$$

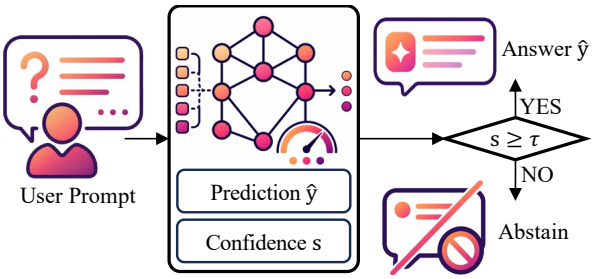

Figure 2. SP for LLMs in deployment.

The selector g typically takes the form

$$g_{s,\tau}(x') \doteq \mathbb{I}\{s(x') \geq \tau\}, \quad (2.9)$$

where $s: \mathcal{X}' \to [0,1]$ is a confidence scorer and $\tau \in [0,1]$ is a selection threshold. The selector chooses to keep only the most confident predictions based on the confidence scorer s with a confidence cutoff τ. To evaluate the performance of SP, two metrics are of particular interest: the **coverage** is the portion of predictions *effectively* covered by the selector, i.e., not rejected by the selector

$$\phi_{s,\tau}(f) \doteq \mathbb{E}_{(x',y')\sim\mathcal{D}'}[g_{s,\tau}(x')]; \quad (2.10)$$

the **selective risk** is the risk over the covered subpopulation

$$R_{s,\tau}(f) \doteq \mathbb{E}_{(x',y')\sim\mathcal{D}'}[\mathbb{I}\{f(x')\neq y\}\, g_{s,\tau}(x')]/\phi_{s,\tau}(f), \quad (2.11)$$

where the binary loss $\mathbb{I}\{f(x')\neq y\}$ could be replaced by other appropriate losses. For any fixed (f,s), there is typically a tradeoff between coverage and selective risk determined by τ. So, one can vary τ over $[0,1]$ to plot the entire tradeoff curve, called the **risk-coverage curve**. A popular summary metric for quantifying SP performance is the **area under the risk-coverage curve** (AURC). For a finite dataset of size N, the coverage varies by $1/N$ (assuming that there is no tie in the confidence). So, the empirical AURC is computed as

$$\overline{\text{AURC}} \doteq \frac{1}{N}\sum_{i=1}^N R_{(i)}, \quad \text{where } R_{(i)} = \frac{1}{i}\sum_{j=1}^i \mathbb{I}\{f(x_{(j)})\neq y_{(j)}\} \quad (2.12)$$

i.e., $R_{(i)}$ denotes the selective risk computed on the subset of the most confident samples i (the subscript $\cdot_{(j)}$ indicates the sample with the j-th highest confidence score). Clearly, many of the indicator terms are repeated across different $R_{(i)}$'s. The equivalent form of Zhou et al. (2025) collects the repeated indicator terms together so that each sample appears inside the summation exactly once, i.e., as a weighted empirical loss

$$\overline{\text{AURC}}^{\text{w}} = \frac{1}{N}\sum_{i=1}^N \widehat{\alpha}_i \cdot \mathbb{I}\{f(x_i)\neq y_i\}, \quad \text{with } \widehat{\alpha}_i = H_N - H_{N-r_i} = \sum_{j=1}^{r_i}\frac{1}{N-j+1}. \quad (2.13)$$

Here, r_i denotes the ranking of the i-th sample in ascending order of confidence, and H_N represents the N-th harmonic number. This weighted form is our primary focus below thanks to its conciseness.

2.2. Our Method: Reinforcement Learning for Selection Reward (RLSR)

As discussed in Sec. 1, although RLVR and RLCR could improve LLM accuracy and perhaps also confidence calibration, they are not sufficient to boost LLM SP performance. Our goal here is to directly align pretrained LLMs with SP based on the GRPO framework described in Sec. 2.1. To implement GRPO, it is natural to design the reward to be consistent with $\overline{\mathrm{AURC}}^{\mathrm{w}}$ defined in Eq. (2.13). For SP, we choose to work with **verbalized confidence** as an exemplar confidence, following RLCR (Damani et al., 2026) and due to its simplicity and recent popularity (Xiong et al., 2024; Kumaran et al., 2026; Geng et al., 2024; Xia et al., 2025).

Two challenges with AURC. The negative version of $\overline{\mathrm{AURC}}^{\mathrm{w}}$ in Eq. (2.13) can serve as a natural reward at the population-level. However, two fundamental challenges arise when we try to integrate it into GRPO:

(1) **Sparse rewards.** $\overline{\mathrm{AURC}}^{\mathrm{w}}$ is a weighted sum of binary indicators of the form $\mathbb{I}\{f(x_i) \neq y_i\}$, which produces sparse one-sided learning signals with two deficiencies: **(i) No signals for correct predictions.** Since $\overline{\mathrm{AURC}}^{\mathrm{w}}$ only penalizes errors and does not assign learning signals to correct predictions, this could slow down learning or even impair it; **(ii) No margin enforcement.** Since correct predictions are not rewarded regardless of their confidence levels, a confident correct prediction and a lucky guess are treated equally. This would allow the confidence margin between correct and wrong predictions to be arbitrarily close—the alignment process has no incentives to promote it, which can in turn hurt generalization performance;

(2) **Rewards based on ranking.** The definition of $\overline{\mathrm{AURC}}^{\mathrm{w}}$ in Eq. (2.13) depends on $\widehat{\alpha}_i$, which depends on the ranking of the i-th sample over the entire dataset. This means that $\overline{\mathrm{AURC}}^{\mathrm{w}}$ is a population-level metric that depends on the global ranking of all samples. However, mini-batch stochastic gradient descent (SGD) applied to the GRPO framework clearly relies on mini-batch rewards in each step, which cannot be decoupled from the global ranking. But global reranking per iteration step would defeat the purpose of mini-batch SGD in avoiding computation over the whole dataset.

Lifted AURC to address the first challenge. We introduce a reformulation called *Lifted AURC*. The idea is to shift the binary indicator $\mathbb{I}\{\widehat{y}_i \neq y_i\} \in \{0, 1\}$ to a signed variable $2\mathbb{I}\{\widehat{y}_i \neq y_i\} \in \{0, 1\} - 1 \in \{-1, +1\}$ directly on Eq. (2.13), resulting in

$$\overline{\mathrm{AURC}}^{\mathrm{w}}_{\mathrm{lift}} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \widehat{\alpha}_i \cdot \big(2\mathbb{I}\{\widehat{y}_i \neq y_i\} - 1\big). \quad (2.14)$$

Since $\frac{1}{N}\sum_{i=1}^{N} \widehat{\alpha}_i \cdot \big(2\mathbb{I}\{\widehat{y}_i \neq y_i\} - 1\big) = \frac{2}{N}\sum_{i=1}^{N} \widehat{\alpha}_i \cdot \mathbb{I}\{\widehat{y}_i \neq y_i\} - \frac{1}{N}\sum_{i=1}^{N} \widehat{\alpha}_i = \frac{2}{N}\sum_{i=1}^{N} \widehat{\alpha}_i \cdot \mathbb{I}\{\widehat{y}_i \neq y_i\} - 1$, we have $\overline{\mathrm{AURC}}^{\mathrm{w}}_{\mathrm{lift}} = 2\overline{\mathrm{AURC}}^{\mathrm{w}} - 1$. So, minimizing $\overline{\mathrm{AURC}}^{\mathrm{w}}_{\mathrm{lift}}$ is equivalent to minimizing $\overline{\mathrm{AURC}}^{\mathrm{w}}$.[1] This lifted version $\overline{\mathrm{AURC}}^{\mathrm{w}}_{\mathrm{lift}}$ provides two-sided rewards that actively reward correct predictions and penalize incorrect ones. The opposite signs of the nonzero rewards for correct and wrong predictions, respectively, create a "push-pull" dynamic that drives correct and wrong predictions apart in their confidence rankings, thereby improving the confidence margin. This margin enforcement is critical for SP, which relies on a sufficient separation of confidence between correct and wrong predictions.

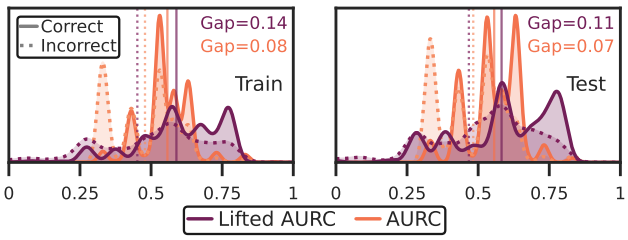

Figure 3. Normalized confidence distributions of correct and incorrect predictions. Empirically, lifted AURC as a reward achieves larger separation (margin) between correct and incorrect predictions on both train and test sets (see Sec. C.7).

In Fig. 3, we empirically confirm the margin benefit of using the lifted AURC as a reward, compared to the plain AURC. Consistent with $\overline{\mathrm{AURC}}^{\mathrm{w}}_{\mathrm{lift}}$, we define the per-sample RLSR reward as

$$R_{\mathrm{RLSR}}(y_i, \widehat{y}_i, s_i) = \begin{cases} +\widehat{\alpha}_i & \text{if } \widehat{y}_i = y_i \\ -\widehat{\alpha}_i & \text{if } \widehat{y}_i \neq y_i \end{cases}. \quad (2.15)$$

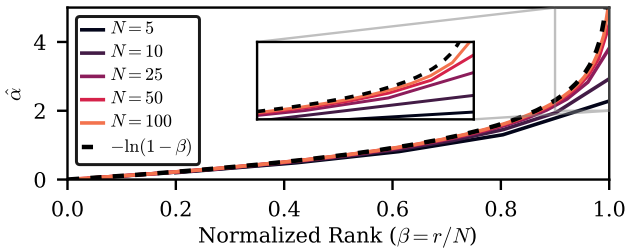

Figure 4. Behavior of the rank-based weights $\widehat{\alpha}$.

To confirm that this definition of the per-sample reward makes intuitive sense, in Fig. 4, we plot the weights $\widehat{\alpha}$

[1]The equivalence is in the sense that they have the same set of global solutions. But due to the significant overparameterization of LLMs, the global solution is unlikely to be unique and they may have different levels of generalization.

against the normalized rank $\beta = r/N$ (in ascending order of confidence) for varying batch sizes $N \in \{5, \ldots, 100\}$. The empirical weights converge to the theoretical curve $-\log(1-\beta)$ (Zhou et al., 2025). Moreover, this weighting scheme places much more emphasis on high-confidence predictions than on low-confidence ones (the asymptotic slope of the weight curve moving from low-confidence to high-confidence is $1/(1-\beta)$). This is sensible, as high-confidence ones are likely to persist in the selected subset when one gradually increases the selection threshold τ, than low-confidence ones that tend to be rejected early. Thus, attending more to the high-confidence ones helps improve the overall SP performance, aligned with our goal.

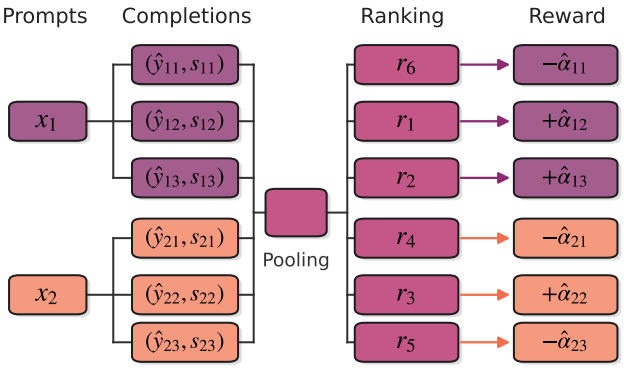

Figure 5. **Illustration of implementation.** We pool rollouts from B inputs, each generating G samples, into an effective batch of size $B \times G$ (e.g., $B = 2$, $G = 3$). Rewards are computed based on ranking over the effective batch. The subsequent relative advantage computation is identical to standard GRPO within each input group.

Algorithm 1 Our algorithm: GRPO with R_{RLSR} (Eq. (2.15))

Require: LLM π_θ, data $\mathcal{D}$, generations G, clipping ε, regex parser $\mathcal{P}$ for answer and verbalized confidence

for step $= 1, \ldots, T$ **do**
  Sample batch $\{(x_b, y_b)\}_{b=1}^{B}$ from $\mathcal{D}$
  $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  for $b = 1, \ldots, B$ **do**
    Sample G outputs from $\pi_{\theta_{\text{old}}}(\cdot \mid x_b)$
    $(\widehat{y}_{b,g},\ s_{b,g}) \leftarrow \mathcal{P}(\pi_{\theta_{\text{old}}}(\cdot \mid x_b))$ for $g = 1, \ldots, G$
  end for
  // Pool $n{=}B{\times}G$ rollouts for batch-level ranking
  Sort all n samples ascendingly; assign $r_i \in \{1, \ldots, n\}$
  $\widehat{\alpha}_i \leftarrow H_n - H_{n-r_i}$ $\quad$ {where $H_k \doteq \sum_{j=1}^{k} 1/j$}
  $R_i \leftarrow \begin{cases} +\widehat{\alpha}_i & \widehat{y}_i = y_i \\ -\widehat{\alpha}_i & \widehat{y}_i \neq y_i \end{cases}$
  // Group-relative advantages (GRPO)
  for $b = 1, \ldots, B$ **do**
    $A_{b,g} \leftarrow (R_{b,g} - \text{mean}_g(R_{b,\cdot}))\,/\,\text{std}_g(R_{b,\cdot})$
  end for
  // Clipped policy update (per-token)
  $\rho_{i,t} \leftarrow \pi_\theta(o_{i,t} \mid x, o_{i,<t})\,/\,\pi_{\theta_{\text{old}}}(o_{i,t} \mid x, o_{i,<t})$
  $\mathcal{L} \leftarrow -\mathbb{E}_{i,t}\big[\min\big(\rho_{i,t}A_i,\ \text{clip}(\rho_{i,t}, 1{\pm}\varepsilon)A_i\big)\big]$
  Update π_θ by $\nabla_\theta \mathcal{L}$
end for
return π_θ

Batch training to address the second challenge. Although R_{RLSR} provides a reasonable reward per-sample, computing the weight $\widehat{\alpha}_i$ still requires ranking across all samples. We address this by the following batch training strategy (Fig. 5). For each mini-batch of B samples, we independently generate G rollouts for each sample and then pool all these $B \times G$ rollouts into an effective batch. We then compute ranks r_i and weights $\widehat{\alpha}_i$ across this pooled set, approximating the population-level AURC. The batch-level ranking step introduces negligible overhead and remains compatible with standard GRPO throughout. The pseudocode of the entire algorithm can be found in Alg. 1.

Comparison with RLVR. When rollouts for an input are all correct or wrong, RLVR assigns the same reward to all, yielding zero advantage and no learning signal. However, this *sparse reward* issue is less likely for RLSR because it differentiates rollouts by confidence ranks, providing a unique reward to each rollout (if no ties or ties are broken by averaging $\widehat{\alpha}_i$). This allows the model to separate low-confidence lucky guesses from high-confidence predictions and penalize confident errors more heavily than uncertain mistakes.

3. Related Work

Confidence scorers for LLMs. In the SP literature, confidence scorers are often derived from prescribed rules (e.g., these based on output logits) or learned confidence predictors (Liang et al., 2024a). For LLMs, rule-based confidence scorers proposed in the literature are numerous, but the community has not converged to a standard (Rabanser & Papernot, 2025). Recent work (Chen et al., 2023; Mao et al., 2025) has also developed learned confidence scorers for LLMs. Our RLSR framework is compatible with both types of confidence scorers, as our reward only requires a confidence scorer that ranks the samples alongside a correctness verifier.

Direct learning with AURC. In machine learning, learning with the AURC objective is still rare. Recently, Franc et al. (2023); Zhou et al. (2025) have explored this for classification problems, but their focuses are mostly on the effective computation and approximation of AURC for learning and optimization purposes. Our method in Sec. 2.2 hinges on the key equivalent definition of AURC as weighted empirical risk (i.e., Eq. (2.13)) developed in Zhou et al. (2025). However, the two technical challenges we address in Sec. 2.2 are unique to the GRPO framework (and RL problems in general) and our RLSR algorithm, and therefore not considered in Franc et al. (2023); Zhou et al. (2025).

We include more discussion of remoter related work in Sec. B.

4. Experiments

To verify the efficacy of our RLSR alignment framework, we benchmark RLSR against RLVR and RLCR in terms of risk-coverage tradeoff on both in-domain and out-of-domain tasks in Sec. 4.1. We further evaluate these alignment frameworks under a risk-controlled deployment setting in high-stakes domains in Sec. 4.2.

4.1. Benchmarking in Risk-Coverage Tradeoff

Models and datasets. We consider both Qwen2.5-7B (Qwen et al., 2025) and Llama-3.1-8B (Grattafiori et al., 2024) as our base LLM models. We perform alignment training on two datasets: HotPotQA-Modified (Yang et al., 2018) and BigMath (Albalak et al., 2025). HotPotQA-Modified is a multi-hop question answering dataset that tests reasoning under incomplete or distracting evidence, while BigMath is a multi-step mathematical reasoning dataset where uncertainty accumulates across steps. *To diversify the evaluation sets and cover different uncertainty types other than those in the training datasets, we include* 6 *additional datasets:* SimpleQA and TriviaQA (Wei et al., 2024; Joshi et al., 2017) probe overconfidence on rare factual knowledge; GPQA, Math500, and GSM8K (Rein et al., 2024; Hendrycks et al., 2021; Cobbe et al., 2021) assess multi-step reasoning; and CommonsenseQA (Talmor et al., 2019) examines ambiguous, implicit reasoning scenarios. To construct the in-domain (ID) and out-of-domain (OOD) evaluation sets, we organize the datasets as follows: for HotPotQA-aligned models, ID evaluation only uses HotPotQA, and OOD evaluation includes the remaining 6 additional datasets; for BigMath-aligned models, Math500, GSM8K, and BigMath are for ID evaluation, while the rest are for OOD evaluation.

Competing methods, confidence scorers, and evaluation metrics. **Competing methods**. The key baseline methods we include are (1) BASE, no alignment, (2) RLVR, the standard correctness-only alignment, and (3) RLCR, the recent correctness+calibration alignment. We also compare against non-alignment alternatives such as supervised finetuning in Sec. D.3; **Confidence scorers**. To elicit verbalized confidence and reasoning about uncertainty, we include a format reward that encourages the model to output an analysis and a confidence score after the chain-of-thought answer (see Sec. C.4.2). Since our method is agnostic to the choice of the confidence scorer, we also experiment with other confidence scorers for the selector (see Sec. D.4); **Evaluation metrics**. Our primary metric is the standard SP metric AURC, which directly measures risk-coverage tradeoff. In addition, we also report the accuracy (i.e., $1-\text{risk}$) at the 10%, 25%, and 50% coverage levels, respectively, as in practical deployment the SP regime with low coverage but high accuracy is often the most valuable, other than the whole risk-coverage tradeoff summarized by AURC. For completeness, we also report ECE, the standard calibration metric. But note that ECE is not a direct measure of SP performance and is not optimized by RLSR.

Training and inference details. For each prompt, we generate responses with temperature $T = 0.7$ and use an effective batch size of $B \times G = 1536$ samples. We then rank the samples inside the effective batch by their confidence scores to compute the RLSR reward (Alg. 1). To elicit structured outputs, we adapt the RLCR system prompts for HotPotQA and BigMath, which induce `<think>`, `<answer>`, `<analysis>`, and `<confidence>` tags. The maximum response length is 1536 tokens for HotPotQA and 4096 tokens for BigMath. All competing methods share identical training configurations and operate under equal inference budgets for fair comparison; full hyperparameter details are in Sec. C.4. Following Turtel et al. (2025), we omit the std normalization when calculating the GRPO advantage; an ablation study confirms the difference is minor (HotPotQA ID AURC: 0.46 with std normalization vs. 0.44 without)). In evaluation, all results use greedy decoding ($T = 0$) for reproducibility; statistical significance over 5 seeds at $T = 0.3$ is reported in Sec. D.2.

Superior in-domain performance. From Tab. 1, it is evident that RLSR consistently outperforms RLVR and RLCR in terms of both AURC and accuracy-at-coverage across all three coverage levels for ID evaluation. For example, for models trained on HotPotQA, RLSR achieves an AURC of 0.44 compared to 0.56 for RLVR and 0.51 for RLCR; and at 10% coverage, RLSR reaches 66.0% accuracy versus 45.5% for RLVR and 53.5% for RLCR. Models trained on BigMath exhibit similar gains across all metrics. Notably, RLCR achieves the best ECE but lags behind RLSR in AURC, confirming that calibration and SP are fundamentally different objectives as discussed in Sec. 1: a model optimized for calibration does not necessarily lead to better SP performance, and vise versa. So, RLSR is a better choice if selection is the goal.

RLSR's SP capability transferable to novel domains. RLSR's SP performance remains strong on OOD evaluation, as shown in Tab. 1. It uniformly improves over all baseline methods, and particularly by large margins on the HotPotQA group. Surprisingly, RLVR always degrades the OOD SP performance relative to the base method without any alignment, and RCLR degrades the OOD SP performance in HotPotQA, but improves the performance in BigMath. We suspect that such performance degradation is partially due to overfitting the ID distribution during the alignment process, which makes the model less robust to distribution shifts. Notably, RLSR's SP performance advantage remains consistent despite the distribution shifts.

Table 1. Comparison of SP performance on in-domain (ID) (2 for HotPotQA and 3 for BigMath) and out-of-domain (OOD) evaluation datasets (6 for HotPotQA and 5 for BigMath) with the Qwen2.5-7B model. Orange highlights the best result per metric within each training setting. RLSR also outperforms all baselines on the Llama-3.1-8B model; see Tab. 4.

	In-Domain					Out-of-Domain				
Method	AURC↓	ECE↓	Acc@10↑	Acc@25↑	Acc@50↑	AURC↓	ECE↓	Acc@10↑	Acc@25↑	Acc@50↑
HotPotQA Training										
BASE	0.60	0.57	46.0%	40.6%	37.8%	0.44	0.40	59.1%	57.4%	54.9%
RLVR	0.56	0.56	45.5%	42.0%	41.8%	0.47	0.46	52.6%	52.9%	53.1%
RLCR	0.51	0.05	53.5%	49.0%	50.4%	0.46	0.23	53.5%	54.3%	55.3%
RLSR	0.44	0.24	66.0%	63.2%	57.2%	0.41	0.28	62.0%	60.7%	59.5%
BigMath Training										
BASE	0.41	0.40	58.9%	61.5%	57.9%	0.49	0.44	54.7%	51.1%	49.9%
RLVR	0.28	0.25	72.2%	70.7%	71.5%	0.53	0.49	46.4%	43.9%	46.7%
RLCR	0.26	0.12	76.7%	75.5%	73.6%	0.47	0.21	57.3%	53.8%	52.1%
RLSR	0.24	0.18	78.8%	78.4%	76.4%	0.45	0.34	58.5%	56.3%	54.2%

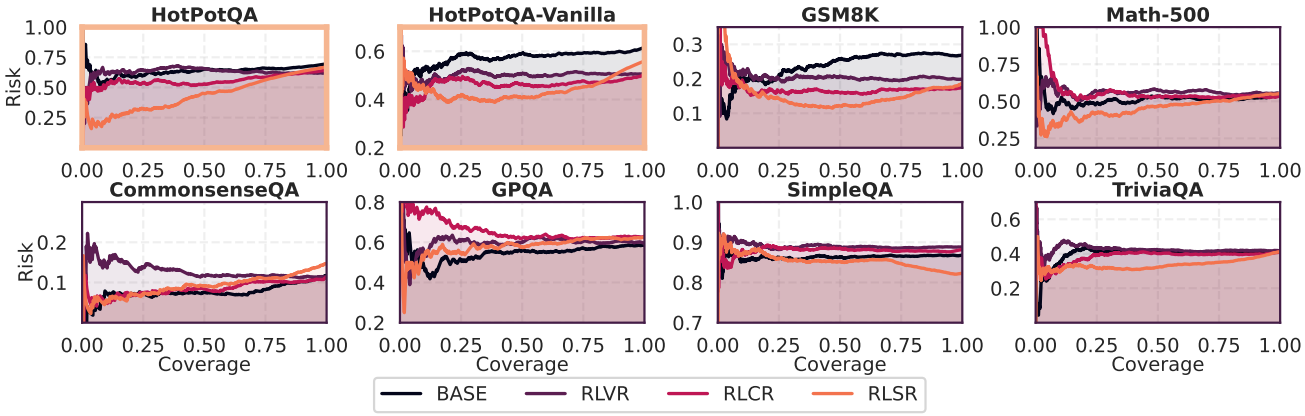

Figure 6. The risk-coverage curves of the Qwen2.5-7B model aligned on HotPotQA. The in-domain datasets are framed in orange while the remaining are out-of-domain datasets.

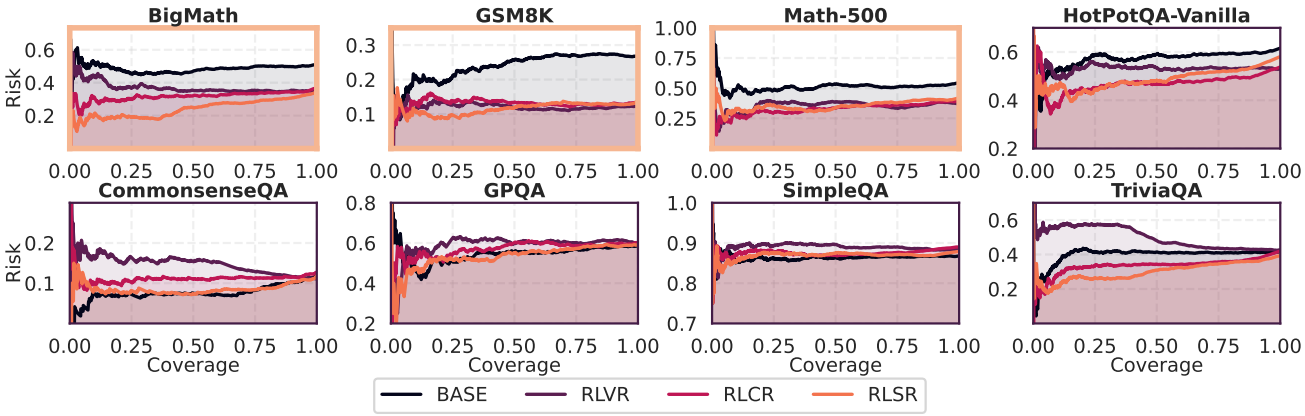

Figure 7. The risk-coverage curves of the Qwen2.5-7B model aligned on BigMath. The in-domain datasets are framed in orange while the remaining are out-of-domain datasets.

Table 2. Controlled confidence gap analysis on HotPotQA ID (Qwen2.5-7B). We restrict to cases where all 3 methods predict correctly or incorrectly, controlling accuracy as a confounder. $\Delta c = \mathbb{E}[c|\text{correct}] - \mathbb{E}[c|\text{incorrect}]$ measures how well each method separates correct from incorrect predictions by confidence.

Method	$\mathbb{E}[c \mid \text{correct}]$	$\mathbb{E}[c \mid \text{incorrect}]$	$\Delta c \uparrow$
RLVR	1.00	0.98	0.01
RLCR	0.45	0.39	0.06
RLSR	**0.89**	**0.52**	**0.37**

Discussion. To understand where RLSR's performance gains originate, we examine the detailed behavior of the risk-coverage curves in Figs. 6 and 7. A revealing pattern emerges: at full coverage, RLSR typically achieves comparable (or even slightly higher) risk than RLVR and RLCR, indicating similar overall accuracy. However, as coverage decreases, the RLSR's risk drops substantially faster, yielding a lower AURC. This behavior directly confirms the benefit of our alignment objective: by optimizing a lifted AURC reward, RLSR learns to assign higher confidence to correct predictions and lower confidence to incorrect ones (see Tab. 2 for empirical separation), precisely the ranking needed for effective selection. In contrast, RLCR shows a more gradual decline, and RLVR's curve often remains flat or even increases at low coverage, implying inconsistent confidence assignment for totally ignoring confidence in RLVR.

Abalation study. Since RLSE approximates the global ranking through batch ranking, we study the effect of batch size on the final SP performance. For this, we fix $G = 32$ and test batch size $B = 48, 32, 16$ respectively for Qwen2.5-7B training. In evaluation, the HotPotQA ID AURC is $0.44, 0.44, 0.45$ respectively, and the OOD AURC is $0.41, 0.47, 0.48$ respectively. In general, SP performance degrades gracefully as batch size decreases.

4.2. Benchmarking in Risk-Controlled Deployment

To show the effectiveness of RLSR in high-stakes deployment, we apply the models trained on HotPotQA to MedQA (Jin et al., 2021), a medical question-answering benchmark derived from various professional medical board and licensing examinations. This represents a challenging OOD setting where risk control is critical.

Experimental setup. Since all alignment methods on MedQA achieve full-coverage accuracy between 57% and 59% (see Fig. 8), we set a target accuracy at 75%, much higher than the full-coverage accuracy. We focus on this low-coverage, low-risk regime with a controlled risk level to emulate practical high-stakes deployment scenarios. To achieve the target accuracy, we find the cutoff threshold τ for each method on a validation set with samples $1,272$, and then apply the threshold to the test set with samples $1,273$.

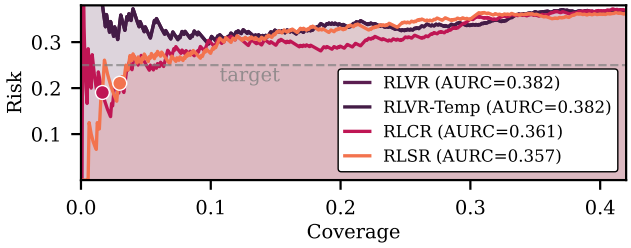

Figure 8. The risk-coverage curves of the three alignment methods on the MedQA validation set. The dashed line indicates the 25% target risk (i.e., 75% accuracy), and the dots mark the actual operating point of each method—not exactly on 25% risk level due to threshold selection on the finite validation set.

Table 3. SP performance of the three methods on MedQA (target accuracy: 75%). Temperature scaling (`RLVR-Temp`) compresses RLVR's confidence range to $[0.38, 0.62]$, so $\tau = 0.62$ is the maximum achievable threshold; $(\cdot)^\dagger$ highlights methods not able to achieve the target accuracy.

Method	τ	**Val. Cov.**	**Val. Acc.**	**Test Cov.**	**Test Acc.**
RLVR†	1.0	23.0%	66.6%	–	–
RLVR+Temp†	0.62	23.0%	66.6%	–	–
RLCR	0.95	1.7%	81.0%	1.4%	72.2%
RLSR	1.0	3.0%	78.9%	3.0%	**78.9%**

Results and discussion. As shown in Tab. 3 and Fig. 8, RLVR fail to achieve the target accuracy 75% throughout the coverage range, due to their overly high and ill-calibrated confidence regardless of the prediction correctness—the very issue that RLCR aims to address (Damani et al., 2026; Leng et al., 2025). Post-hoc temperature scaling (Guo et al., 2017; Wang et al., 2024) (see Sec. C.6.5) over RLVR (RLVR-Temp) does not improve or even change the SP performance, as it simply performs monotonic rescaling of the confidence score and hence preserves confidence ranking. This underscores the importance of optimizing the confidence scorers in RLCR and RLSR. RLSR's advantage is most pronounced in the low-coverage, low-risk regime (see Fig. 8): RLSR's risk-coverage curve is not only almost uniformly lower than that of RLCR's, but reaches very low risk—RLCR's reaches a significant risk floor ($\sim 15\%$) first and then bounces back as coverage level decreases. For error control (see Tab. 3), only RLSR and RLCR achieve the 75% target accuracy on the validation set but have very different generalization behaviors: while RCLR's test accuracy drops below the target (81.0% validation $\rightarrow$ 72.2% test), RLSR has the same test accuracy and coverage as the validation ones. Overall, RLSR obtains simultaneous higher accuracy and coverage than RLCR's by large margins on the test set, demonstrating RLSR's strong error-controlled SP performance.

5. Conclusion and Limitation

In this paper, we improve LLM reliability by aligning them with selective prediction (SP). Our alignment method, RLSR, directly optimizes the area under the risk-coverage curve (AURC), a standard SP metric. Our experiments confirm that RLSR significantly outperforms other competing alignment frameworks, including RLVR and RLCR, in terms of both AURC and particularly error-controlled deployment scenarios, which are most relevant for real-world applications. A major limitation is that our current alignment focuses on the holistic AURC metric, in contrast to the more useful error-control setting. Knowing the target risk, one could formulate risk-constrained coverage maximization alignment objectives.

Impact Statement

This work focuses on the refusal capability of LLMs so that the retained subset can be more accurate. It operates purely on ranking and thereby does not rely on confidence scores that are calibrated or normalized; therefore, it is applicable to plenty of uncertainty quantification methods (e.g., logit-based, consistency-based, etc) in LLMs. Although this work is evaluated on single-turn benchmarks only, the idea of aligning with selective prediction can be applied to selective tool-calling, and the post-training methods can be extended from single-turn to multi-turn for agentic tasks.

Acknowledgments and Disclosure of Funding

This work was supported in part by a UMN DSI-MnDRIVE PhD Graduate Assistantship, the National Institutes of Health (NIH) under award No. R01CA287413, R01NS131314, R01MH138929, U01FD008720, the National Science Foundation (NSF) under award No. 2435911. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors acknowledge the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported within this paper.

References

Albalak, A., Phung, D., Lile, N., Rafailov, R., Gandhi, K., Castricato, L., Singh, A., Blagden, C., Xiang, V., Mahan, D., and Haber, N. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models, 2025. URL https://arxiv.org/abs/2502.17387.

Chen, J., Yoon, J., Ebrahimi, S., Arik, S. O., Pfister, T., and Jha, S. Adaptation with self-evaluation to improve selective prediction in LLMs. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=haPIkA8aOk.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Cole, J., Zhang, M., Gillick, D., Eisenschlos, J., Dhingra, B., and Eisenstein, J. Selectively answering ambiguous questions. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 530–543, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.35. URL https://aclanthology.org/2023.emnlp-main.35/.

Damani, M., Puri, I., Slocum, S., Shenfeld, I., Choshen, L., Kim, Y., and Andreas, J. Beyond binary rewards: Training LMs to reason about their uncertainty. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ASQ649zdHm.

Dao, T. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Fan, T., Yang, Y., Jiang, Y., Zhang, Y., Chen, Y., and Huang, C. Ai-trader: Benchmarking autonomous agents in real-time financial markets, 2025. URL https://arxiv.org/abs/2512.10971.

Fisch, A., Jaakkola, T. S., and Barzilay, R. Calibrated selective classification. *Transactions on Machine Learning Research*, 2022.

Franc, V., Prusa, D., and Voracek, V. Optimal strategies for reject option classifiers. *Journal of Machine Learning Research*, 24(11):1–49, 2023. URL http://jmlr.org/papers/v24/21-0048.html.

Gaber, F., Shaik, M., Allega, F., Bilecz, A. J., Busch, F., Goon, K., Franke, V., and Akalin, A. Evaluating large language model workflows in clinical decision support for triage and referral and diagnosis. *npj Digital Medicine*, 8(1):263, May 2025. ISSN 2398-6352. doi: 10.1038/s41746-025-01684-1. URL https://doi.org/10.1038/s41746-025-01684-1.

Geng, J., Cai, F., Wang, Y., Koeppl, H., Nakov, P., and Gurevych, I. A survey of confidence estimation and calibration in large language models. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6577–6595, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.366. URL https://aclanthology.org/2024.naacl-long.366/.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel,

M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Xu, H., Ding, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Chen, J., Yuan, J., Tu, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., You, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Zhou, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, Sep 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hu, J., Zhang, Y., Han, Q., Jiang, D., Zhang, X., and Shum, H.-Y. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025. URL https://arxiv.org/abs/2503.24290.

Jin, D., Pan, E., Oufattole, N., Weng, W.-H., Fang, H., and Szolovits, P. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14), 2021. ISSN 2076-3417. doi: 10.3390/app11146421. URL https://www.mdpi.com/2076-3417/11/14/6421.

Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Barzilay, R. and Kan, M.-Y. (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Kalai, A. T., Nachum, O., Vempala, S. S., and Zhang, E. Evaluating large language models for accuracy incentivizes hallucinations. *Nature*, Apr 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10549-w. URL https://doi.org/10.1038/s41586-026-10549-w.

Kumaran, D., Conmy, A., Barbero, F., Osindero, S., Patraucean, V., and Veličković, P. How do LLMs compute verbal confidence? *arXiv preprint arXiv:2603.17839*, 2026.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Lai, J., Gan, W., Wu, J., Qi, Z., and Yu, P. S. Large language models in law: A survey. *AI Open*, 5:181–196, 2024. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2024.09.002. URL https://www.sciencedirect.com/science/article/pii/S2666651024000172.

Lamb, T. A., Ivanova, D. R., Torr, P., and Rudner, T. G. J. Semantic calibration of LLMs through the lens of temperature scaling. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*, 2025. URL https://openreview.net/forum?id=22UNXHzBgt.

Leng, J., Huang, C., Zhu, B., and Huang, J. Taming overconfidence in LLMs: Reward calibration in RLHF. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=l0tg0jzsdL.

Liang, H., Peng, L., and Sun, J. Selective classification under distribution shifts. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. URL https://openreview.net/forum?id=dmxMGW6J7N.

Liang, X., Wang, H., Wang, Y., Song, S., Yang, J., Niu, S., Hu, J., Liu, D., Yao, S., Xiong, F., et al. Controllable text generation for large language models: A survey. *arXiv preprint arXiv:2408.12599*, 2024b.

Lyu, Q., Shridhar, K., Malaviya, C., Zhang, L., Elazar, Y., Tandon, N., Apidianaki, M., Sachan, M., and Callison-Burch, C. Calibrating large language models with sample consistency. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(18): 19260–19268, Apr. 2025. doi: 10.1609/aaai.v39i18.34120. URL https://ojs.aaai.org/index.php/AAAI/article/view/34120.

Mao, Y., Durand, T., Mehrasa, N., He, J., and Ester, M. Calibrating LLMs for selective prediction: Balancing coverage and risk. In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=ZVZGjtP5VB.

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., and Tran, D. Measuring calibration in deep learning. In *CVPR workshops*, volume 2, 2019.

Quach, V., Fisch, A., Schuster, T., Yala, A., Sohn, J. H., Jaakkola, T. S., and Barzilay, R. Conformal language modeling. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pzUhfQ74c5.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Rabanser, S. and Papernot, N. What does it take to build a performant selective classifier? In *Advances in Neural Information Processing Systems*, 2025.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Saab, K., Tu, T., Weng, W.-H., Tanno, R., Stutz, D., Wulczyn, E., Zhang, F., Strother, T., Park, C., Vedadi, E., Chaves, J. Z., Hu, S.-Y., Schaekermann, M., Kamath, A., Cheng, Y., Barrett, D. G. T., Cheung, C., Mustafa, B., Palepu, A., McDuff, D., Hou, L., Golany, T., Liu, L., baptiste Alayrac, J., Houlsby, N., Tomasev, N., Freyberg, J., Lau, C., Kemp, J., Lai, J., Azizi, S., Kanada, K., Man, S., Kulkarni, K., Sun, R., Shakeri, S., He, L., Caine, B., Webson, A., Latysheva, N., Johnson, M., Mansfield, P., Lu, J., Rivlin, E., Anderson, J., Green, B., Wong, R., Krause, J., Shlens, J., Dominowska, E., Eslami, S. M. A., Chou, K., Cui, C., Vinyals, O., Kavukcuoglu, K., Manyika, J., Dean, J., Hassabis, D., Matias, Y., Webster, D., Barral, J., Corrado, G., Semturs, C., Mahdavi, S. S., Gottweis, J., Karthikesalingam, A., and Natarajan, V. Capabilities of gemini models in medicine, 2024. URL https://arxiv.org/abs/2404.18416.

Schulman, J. and Lab, T. M. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

Sellergren, A., Kazemzadeh, S., Jaroensri, T., Kiraly, A., Traverse, M., Kohlberger, T., Xu, S., Jamil, F., Hughes, C., Lau, C., Chen, J., Mahvar, F., Yatziv, L., Chen, T., Sterling, B., Baby, S. A., Baby, S. M., Lai, J., Schmidgall, S., Yang, L., Chen, K., Bjornsson, P., Reddy, S., Brush, R., Philbrick, K., Asiedu, M., Mezerreg, I., Hu, H., Yang, H., Tiwari, R., Jansen, S., Singh, P., Liu, Y., Azizi, S., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Riviere, M., Rouillard, L., Mesnard, T., Cideron, G., bastien Grill, J., Ramos, S., Yvinec, E., Casbon, M., Buchatskaya, E., Alayrac, J.-B., Lepikhin, D., Feinberg, V., Borgeaud, S., Andreev, A., Hardin, C., Dadashi, R., Hussenot, L., Joulin, A., Bachem, O., Matias, Y., Chou, K., Hassidim, A., Goel, K., Farabet, C., Barral, J., Warkentin, T., Shlens, J., Fleet, D., Cotruta, V., Sanseviero, O., Martins, G., Kirk, P., Rao, A., Shetty, S., Steiner, D. F., Kirmizibayrak, C., Pilgrim, R., Golden, D., and Yang, L. Medgemma technical report, 2025. URL https://arxiv.org/abs/2507.05201.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Burstein, J., Doran, C., and Solorio, T. (eds.), *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)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., Finn, C., and Manning, C. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

Turtel, B., Franklin, D., Skotheim, K., Hewitt, L., and Schoenegger, P. Outcome-based reinforcement learning to predict the future. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=bbhdeL8EUX.

Varshney, N. and Baral, C. Post-abstention: Towards reliably re-attempting the abstained instances in QA. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 967–982, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.55. URL https://aclanthology.org/2023.acl-long.55/.

Wang, C., Szarvas, G., Balazs, G., Danchenko, P., and Ernst, P. Calibrating verbalized probabilities for large language models, 2024. URL https://arxiv.org/abs/2410.06707.

Wei, J., Karina, N., Chung, H. W., Jiao, Y. J., Papay, S., Glaese, A., Schulman, J., and Fedus, W. Measuring short-form factuality in large language models, 2024. URL https://arxiv.org/abs/2411.04368.

Wen, X., Liu, Z., Zheng, S., Ye, S., Wu, Z., Wang, Y., Xu, Z., Liang, X., Li, J., Miao, Z., et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Xia, Z., Xu, J., Zhang, Y., and Liu, H. A survey of uncertainty estimation methods on large language models. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21381–21396, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1101. URL https://aclanthology.org/2025.findings-acl.1101/.

Xiao, C., Zhang, M., and Cao, Y. Bnpo: Beta normalization policy optimization, 2025. URL https://arxiv.org/abs/2506.02864.

Xiong, M., Hu, Z., Lu, X., LI, Y., Fu, J., He, J., and Hooi, B. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Zhang, M., Huang, M., Shi, R., Guo, L., Peng, C., Yan, P., Zhou, Y., and Qiu, X. Calibrating the confidence of large language models by eliciting fidelity. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2959–2979, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.173. URL https://aclanthology.org/2024.emnlp-main.173/.

Zhou, H., Landeghem, J. V., Popordanoska, T., and Blaschko, M. B. A novel characterization of the population area under the risk coverage curve (AURC) and rates of finite sample estimators. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=LBBUJkqkOM.

A. Derivations & Proofs

A.1. Gradient of the LLM alignment objective in Eq. (2.1)

First of all,

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathbb{E}_{\widehat{y}\sim\pi_\theta(\cdot|x)} R(y,\widehat{y}) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \nabla_\theta \mathbb{E}_{\widehat{y}\sim\pi_\theta(\cdot|x)} R(y,\widehat{y}), \tag{A.1}$$

so we can focus on deriving $\nabla_\theta \mathbb{E}_{\widehat{y}\sim\pi_\theta(\cdot|x)} R(y,\widehat{y})$. Now

$$\begin{aligned}
\nabla_\theta \mathbb{E}_{\widehat{y}\sim\pi_\theta(\cdot|x)} R(y,\widehat{y}) &= \nabla_\theta \sum\nolimits_{\widehat{y}} \pi_\theta(\widehat{y}|x) R(y,\widehat{y}) && \text{(A.2)}\\
&= \sum\nolimits_{\widehat{y}} \nabla_\theta \pi_\theta(\widehat{y}|x) R(y,\widehat{y}) && \text{(A.3)}\\
&= \sum\nolimits_{\widehat{y}} \pi_\theta(\widehat{y}|x) \nabla_\theta \log \pi_\theta(\widehat{y}|x) R(y,\widehat{y}) \quad \text{(log-derivative trick)} && \text{(A.4)}\\
&= \mathbb{E}_{\widehat{y}\sim\pi_\theta(\cdot|x)} [\nabla_\theta \log \pi_\theta(\widehat{y}|x) R(y,\widehat{y})], && \text{(A.5)}
\end{aligned}$$

as claimed.

B. More Related Work

Selective algorithms for LLMs. While our paper focuses on LLM alignment and optimizing LLMs for SP, several other lines of orthogonal work focusing on other aspects of LLM deployment also contain selection components. For example, selective answering (Cole et al., 2023) deals with selecting among inherently ambiguous questions; post-abstention (Varshney & Baral, 2023) addresses the question of what to do after abstention, i.e., refine selective prediction; conformal LLM (Quach et al., 2024) adapts conformal prediction, which outputs uncertainty sets, for LLMs; and guided/constrained decoding (Liang et al., 2024b) modifies the LLM autoregressive generation process by selecting and filtering tokens to ensure its output respects prescribed constraints.

Confidence calibration (CC) for LLMs. Overconfidence in LLMs trained with RL with human feedback (RLHF) is widely recognized (Leng et al., 2025; Damani et al., 2026). So, CC has been a dominant line of research to around it (Tian et al., 2023), evidenced by a wide range of post-hoc CC methods (e.g., temperature scaling) shown to improve calibration metrics such as expected calibration error (ECE) (Lamb et al., 2025; Zhang et al., 2024; Lyu et al., 2025). But, our focus of this paper is selective prediction, as discussed in Sec. 1.

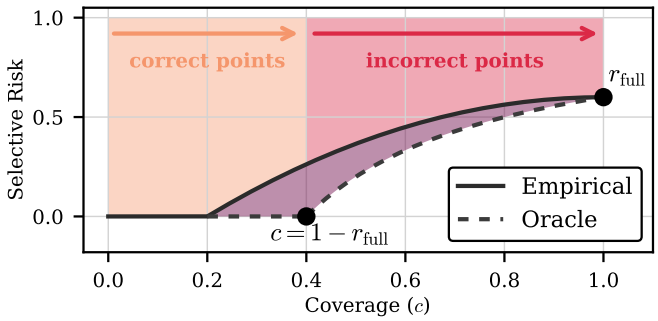

Figure 9. The dashed curve is the oracle under which coverage levels left of $c = 1 - r_{\text{full}}$ (orange) accept all correct predictions first and rank incorrect predictions last (red). r_{full} means full-coverage risk. The purple shaded area visualizes the SP gap.

Selective prediction (SP) gap. In SP, the ideal selector should mimic the *perfect-ordering oracle*, which ranks all correct predictions ahead of all incorrect ones in terms of confidence, thus minimizing the AURC. However, practical selectors typically fall short of this performance ceiling. Rabanser & Papernot (2025) formalizes this shortfall as the *SP gap* (see Fig. 9) and decomposes it into four major sources: Bayes noise, approximation limit, statistical noise, and ranking error. For LLMs, while the approximation limit and statistical noise can be collectively reduced with larger model size and data volume during training, the major obstacle to closing the gap comes from the ranking error, which is determined by the confidence scorer. The goal of our work is to improve the LLMs capabilities as simultaneous predictors and confidence scorers by aligning them with SP, thus improving the ranking error. Rabanser & Papernot (2025) also points out that many post-hoc calibration methods do not improve the ranking error because they are monotone transformations of the original scores and hence leave the ranking of samples intact.

C. Experiment Details

C.1. Prompt Templates

We use structured prompts that guide the LLM to generate both answers and confidence scores. Following Damani et al. (2026), we prompt the LLM to produce outputs with four components: (1) a thinking trace enclosed in `<think>` tags, (2) a final answer in `<answer>` tags, (3) an uncertainty analysis in `<analysis>` tags, and (4) a numerical confidence score in `<confidence>` tags:

Example Response

`<think>` The problem is asking for the person who established the Journal of Libertarian Studies as well as [...] but their connection to the establishment of the Journal of Libertarian Studies is not clear.`</think>`
`<answer>`**Murray Newton Rothbard**`</answer>`
`<analysis>` One uncertainty arises from the fact that while [...], the information about him writing over 20 books on political theory is less direct. It would be better if [...].`</analysis>`
`<confidence>`**0.85**`</confidence>`

We use two variants of prompts. For Big-Math Digits, we use the *Simple Prompt*; for HotPotQA and HotPotQA-Modified, we use the *Long Prompt*, which provides additional guidelines for uncertainty analysis.

System Prompt (Simple)

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 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. The final format that must be followed is: `<think>` reasoning process here `</think><answer>` final answer here `</answer><analysis>` analysis about confidence and uncertainty here `</analysis><confidence>` confidence level here (number between 0 and 1) `</confidence>`

System Prompt (Long)

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 `</answer><analysis>` analysis about confidence and uncertainty here `</analysis><confidence>` confidence level here (number between 0 and 1) `</confidence>`

C.2. Verifiers

Verifiers are used during training to compute rewards and during evaluation to assess correctness. We employ three types of verifiers, following Damani et al. (2026):

1. **Exact-Match.** The predicted answer must exactly match the ground truth answer string. This verifier is used for HotPotQA and HotPotQA-Modified.
2. **Math-Verify.** We use `math-verify`, a robust mathematical expression evaluation system that checks semantic equivalence of mathematical expressions. This verifier is used for Math-500, GSM8K, and Big-Math Digits.
3. **LLM-as-a-Judge.** We use Llama-3.1-8B-Instruct with greedy decoding (temperature=0) as the judge model. The judge is provided with the question, the groundtruth answer, and the predicted answer, and is prompted to respond with "YES" or "NO" based on correctness. Importantly, we do not condition the judge on thinking traces to avoid potential biases. This verifier is used for TriviaQA, SimpleQA, CommonsenseQA, GPQA, and MedQA.

LLM-as-a-Judge Prompt

You are given a question, a ground truth answer, and a predicted answer. Determine whether the predicted answer is correct based on the ground truth. Respond with only "YES" if the predicted answer is correct, or "NO" if it is incorrect.
Question: {question}
Ground Truth Answer: {ground_truth}
Predicted Answer: {predicted_answer}
Is the predicted answer correct?

C.3. LLM Response Parser

All responses use the `<think>`, `<answer>`, `<analysis>`, and `<confidence>` tags. The answers are extracted from the `<answer>` tags and confidence scores from the `<confidence>` tags. For models that do not follow the expected format, we append "Thinking time ended. My final answer is" to the output and prompt the model again to extract the answer. Similarly, if a valid confidence score cannot be extracted, we append "Thinking time ended. My verbalized confidence in my answer as a number between 0 and 1 is equal to" and prompt the model again. *Note that models trained with format rewards (see Sec. C.4.2) rarely trigger these fallbacks*.

C.4. Training

C.4.1. Training Datasets

1. **HotPotQA-Modified.** We adopt the HotpotQA-Modified dataset introduced by Damani et al. (2026), which is derived from the original HotPotQA distractor dataset (Yang et al., 2018). This modified version systematically varies the availability of supporting evidence by removing zero, one, or both of the key paragraphs required to answer each question. The questions are evenly distributed across these three conditions, and each instance contains 8 paragraphs in total. This construction creates varying levels of information completeness, making it particularly suitable for developing and evaluating uncertainty reasoning capabilities. The training set comprises $20,000$ examples, and correctness is evaluated by exact string matching. HotPotQA is distributed under CC BY-SA 4.0 License and HotPotQA-Modified is under MIT License.
2. **BigMath.** We use BigMath (Albalak et al., 2025), a large-scale curated dataset for reinforcement learning that contains over $250,000$ mathematical problems. Following Damani et al. (2026), we retain only problems with LLaMA-8B solve rates between 0% and 70% to maintain appropriate difficulty, and further restrict to problems with numerical answers to enable reliable automatic verification—resulting in the subset referred to as BigMath Digits. The final training set contains $15,000$ problems, with correctness evaluated using `math-verify`. It is distributed under the MIT License.

C.4.2. Training Details

Base LLM models & GRPO setup. We use Qwen2.5-7B (Qwen et al., 2025) under the Apache 2.0 License and Llama3.1-8B (Grattafiori et al., 2024) under the Llama 3.1 Community License. Following recent work on RL for reasoning (Guo et al., 2025; Hu et al., 2025), we initialize RL directly from the base model without supervised finetuning and set $\beta = 0$ (no KL regularization). We use GRPO without standard deviation normalization in the advantage calculation (Turtel et al., 2025) *(an ablation confirms that the difference is minor: HotPotQA ID AURC 0.46 with std normalization vs. 0.44 without)*, along

with the BNPO loss function (Xiao et al., 2025), which aggregates token-level losses using the number of active tokens in the local training batch. For parameter-efficient training, we apply LoRA (Hu et al., 2022) with rank $r = 1$, $\alpha = 32$, and no dropout to all linear layers, motivated by *LoRA Without Regret* (Schulman & Lab, 2025).

Reward functions. The total reward combines a task-specific reward with a format reward:

$$R_{\text{total}} = R_{\{\text{RLVR,RLCR,RLSR}\}} + R_{\text{format}}. \tag{C.1}$$

The method-specific reward is defined in Eq. (2.6) for RLVR, Eq. (2.7) for RLCR, and Eq. (2.15) for RLSR. The format reward $R_{\text{format}} \in \{0, 1\}$ is a binary indicator: 1 if the output follows the expected tag structure (`<think>`, `<answer>`, `<analysis>`, `<confidence>`) and contains a valid confidence score $c \in [0, 1]$, and 0 otherwise. *Note that the format reward converges to near-perfect values early in the training processes for all methods*.

Key hyperparameters. We generate 32 responses per prompt with temperature 0.7. **For HotPotQA**, we use a learning rate of 1×10^{-5} with a constant schedule and 5% warmup, training for 1 epoch with an effective batch size of 1536 (4 GPUs $\times$ 3 per-device batch $\times$ 128 gradient accumulation steps). The maximum prompt and completion lengths are 3072 and 1536 tokens, respectively. **For BigMath**, we use a learning rate of 5×10^{-5} with a linear decay schedule and 20% warmup, training for 0.5 epochs with an effective batch size of 1536 (4 GPUs $\times$ 6 per-device batch $\times$ 64 gradient accumulation steps). The maximum prompt and completion lengths are 1024 and 4096 tokens, respectively. All methods (RLVR, RLCR, RLSR) use identical hyperparameters to ensure fair comparison. All training uses the Transformers Reinforcement Learning (TRL) library, with vLLM as an efficient rollout engine (Kwon et al., 2023). Important system optimizations include `bfloat16` (`bf16`) precision, FlashAttention-2 (FA2) (Dao, 2024) and Triton kernels shipped with TRL.

Training time. Training runs on 4xNVIDIA H100 GPUs, and each run takes approximately 62 hours (HotPotQA) or 57 hours (BigMath).

C.5. Evaluation

We follow RLCR (Damani et al., 2026) for setting up the evaluation protocol. All models are evaluated with greedy decoding (temperature=0) and a maximum token budget of 4096 (sufficient for all selected evaluation benchmarks). We generate a single response per question and use the same system prompts as during training: the simple prompt (Sec. C.1) for Big-Math Digits and the long prompt for HotPotQA.

C.5.1. Evaluation Datasets

1. **HotPotQA.** (CC BY-SA 4.0 License) We use 1,000 validation samples from the original HotPotQA distractor dataset (Yang et al., 2018), with 2 irrelevant paragraphs removed from each question. Thus, each question contains 8 paragraphs with both supporting paragraphs present. Correctness is measured using exact-match.
2. **HotPotQA-Modified.** (MIT License) We use 500 held-out validation samples from the modified dataset introduced by Damani et al. (2026). Correctness is measured using exact-match.
3. **Math-500.** (MIT License) We use the MATH-500 dataset, a subset of problems from the original MATH dataset (Hendrycks et al., 2021). Correctness is measured using `math-verify`.
4. **GSM8K.** (MIT License) We use 1,319 problems from the test set of the Grade School Math 8K dataset (Cobbe et al., 2021). Correctness is measured using `math-verify`.
5. **Big-Math Digits.** (MIT License) We use 1,000 held-out validation samples from the filtered Big-Math dataset (Albalak et al., 2025). Correctness is measured using `math-verify`.
6. **TriviaQA.** (Apache 2.0 License) We use 2,000 samples from the TriviaQA (Joshi et al., 2017) validation set, specifically the no-context split, to test the factual accuracy. Correctness is measured using LLM-as-a-judge.
7. **SimpleQA.** (MIT License) We use the full dataset consisting of 4,326 factual questions (Wei et al., 2024). Correctness is measured using LLM-as-a-judge.
8. **CommonsenseQA.** (MIT License) uses 1,220 problems from the CommonsenseQA (Talmor et al., 2019) validation set, a multiple-choice question answer dataset requiring various types of commonsense knowledge. Correctness is measured using LLM-as-a-judge.
9. **GPQA.** (MIT License) We use the main dataset (gpqa_main) that contains 448 multiple-choice questions written by experts in the fields of biology, physics, and chemistry (Rein et al., 2024). Correctness is measured using LLM-as-a-judge.
10. **MedQA.** (MIT License) We use the 4-option version of MedQA (Jin et al., 2021), consistent with its original assessment and usage by technical reports of frontier medical LLMs (Saab et al., 2024; Sellergren et al., 2025). Correctness is

measured using LLM-as-a-judge.

C.5.2. Evaluation Metrics

1. **AURC (↓).** The *area under risk-coverage curve* measures cumulative selective risk as a function of coverage with sorted prediction confidence. We refer the reader to Sec. 2.1 for the formal definition. Lower values means better SP performance.
2. **ECE (↓).** The *expected calibration error* measures the alignment between confidence and correctness likelihood:

$$\text{ECE} = \sum_{m=1}^{M} |B_m|/N \cdot |\text{acc}(B_m) - \text{conf}(B_m)| \tag{C.2}$$

   where M is the number of bins, B_m is the set of samples in bin m, and N is the total number of samples. We use $M = 10$ bins.
3. **Accuracy@$k\%$ (↑).** Accuracy computed on the top $k\%$ most confident predictions, measuring how well the model's confidence identifies its correct predictions. We report results at coverage levels $k \in \{10, 25, 50\}$.

C.6. Mimicking Selective Prediction in Deployment

C.6.1. The MedQA Dataset

We use the 4-option version of MedQA (Jin et al., 2021), consistent with its original assessment and usage by frontier medical LLMs (Saab et al., 2024; Sellergren et al., 2025). Each question presents a clinical vignette followed by four answer choices (A, B, C, D). We partition the dataset as detailed in the following table:

Split	Samples	Usage
Train	10,178	alignment training
Validation	1,272	threshold selection
Test	1,273	deployment evaluation

The prediction correctness is checked by exact matching by the letter of the answer, i.e., among $\{A, B, C, D\}$. When the prediction does not follow the expected format, we apply robust answer extraction using multiple regex patterns: direct letter match, letter prefix (e.g., "D. Nitrofurantoin"), "The answer is" prefix, and pure-option-text-without-option-letter matching.

C.6.2. Inference Settings

We use the same generation settings as those of the in-domain evaluation (Sec. C.5): temperature set to 0, maximum token budget of 4096, and with the long system prompt (Sec. C.1). We use vLLM (Kwon et al., 2023) for efficient inference. The models are loaded from HotPotQA training checkpoints.

C.6.3. Threshold Selection—A Key Step in Error-Controlled SP Deployment

Given a target accuracy α^* and the validation set D_{val}, we fine the threshold τ via

$$\tau = \min\{t \in [0,1] : \text{acc}(\{(x,y) \in D_{\text{val}} : s(x) \geq t\}) \geq \alpha^*\}, \tag{C.3}$$

where $s(x)$ is the confidence score for the input x. The corresponding coverage is computed as:

$$\text{coverage}(\tau) = |\{x \in D_{\text{val}} : s(x) \geq \tau\}|/|D_{\text{val}}|. \tag{C.4}$$

We set $\alpha^* = 75\%$.

C.6.4. Evaluation Metrics

After threshold selection above, we apply the threshold directly to the test set and report the following metrics:

- **Validation Accuracy**: the prediction accuracy on validation samples with confidence $\geq \tau$
- **Validation Coverage**: the fraction of validation samples with confidence $\geq \tau$
- **Test Accuracy**: the prediction accuracy on test samples with confidence $\geq \tau$
- **Test Coverage**: the fraction of test samples with confidence $\geq \tau$

C.6.5. Post-hoc Calibration Baseline

To evaluate whether post-hoc calibration can rescue poorly-calibrated confidence for SP, we apply temperature scaling (Guo et al., 2017) to RLVR's confidence output. Since our setting does not generate native logits, we follow Wang et al. (2024) which proposes to calibrate a verbalized probability vector $(p_1, \ldots, p_K)$ via an invert-softmax trick: assume that the pre-softmax logits take the form $z \doteq (\log p_1 + b, \ldots, \log p_K + b)$ for an unknown constant b, the idea is to perform the temperature scaling $\mathrm{softmax}(z/T)$. For our single output case, we treat it as a binary classification problem on the $\{\text{correct}, \text{incorrect}\}$ classes, i.e., $K = 2$ so that the effective class probabilities are $(s, 1 - s)$. So:

$$s_T \doteq \mathrm{softmax}(z/T)_{\text{correct}} = \frac{e^{(\log s + b)/T}}{e^{(\log s + b)/T} + e^{(\log(1-s)+b)/T}} = \frac{1}{1 + e^{\frac{1}{T}\log(1-s)/s}} \tag{C.5}$$

$$= \mathrm{sigmoid}(\tfrac{1}{T} \log \tfrac{s}{1-s}), \tag{C.6}$$

yielding our temperature scaling operator. We find an appropriate T by training it over the binary calibration set $\{(s_i, y_i)\}_{i=1}^N$ (s_i: verbalized confidences; $y_i \in \{0, 1\}$: groundtruth correctness label):

$$T^* = \arg\min_{T>0} -\sum_{i=1}^N y_i \log s_T(s_i) + (1 - y_i) \log(1 - s_T(s_i)). \tag{C.7}$$

The optimal $T^* \approx 47.9$ on the MedQA indicates severe overconfidence of the original prediction: it squashes the full probability domain $[0, 1]$ to $\approx [0.38, 0.62]$, and the confidence values the model actually emits, $\{0.5, \ldots, 1.0\}$, collapse into the even narrower band $\approx [0.50, 0.62]$.

It is easy to verify that S_T is a monotonically increasing function of s. Therefore, *it maintains the confidence ranking of the predictions—leaving the ranking-based risk-coverage curve unchanged*, as empirically confirmed in Fig. 8. The same holds for any monotonic scalar recalibration (e.g. Platt or isotonic regression). So, while confidence calibration does not necessarily improve SP performance, improving the confidence ranking is the key.

C.7. Experiment Details of Margin Analysis for the Lifted AURC Versus Canonical AURC

To compare the effect of lifted AURC versus AURC as the reward on the *margin*—the separation of correct and incorrect predictions in terms of confidence, we finetune the base model Qwen2.5-7B using an identical setting except for the reward function: one with lifted AURC (awarding correct predictions and penalizing incorrect predictions) and one with AURC (only penalizing incorrect predictions) on $6,000$ training samples of the HotPotQA dataset—enough for the format reward to converge for faithful parsing of confidence scores. We evaluate both models on the same $6,000$ training samples and 500 held-out test samples. For each sample, we extract the model's predicted answer, its correctness (determined by exact match), and its verbalized confidence score. We then partition the samples into correct and incorrect predictions and estimate their respective *normalized* distributions using kernel density estimation with a bandwidth of 0.15. Finally, we report the confidence gap, i.e., the difference between the mean confidence values of the two distributions. Fig. 4 visualizes these distributions, with solid lines denoting correct predictions and dotted lines denoting incorrect predictions.

D. Complementary Experiment Results

D.1. RLSR Works with Different Modal Families

We follow the same training recipe and evaluation protocol as Tab. 1 for the Llama-3.1-8B model. The conclusion remains the same: RLSR achieves the best AURC (lower the better) performance averaged over in-domain (HotPotQA, HotPotQA-Vanilla) and out-of-domain (SimpleQA, TriviaQA, GPQA, Math500, GSM8K, CommonsenseQA) benchmarks.

Table 4. The Llama-3.1-8B results trained on HotPotQA. Orange highlights the best result.

Method	In-Domain AURC↓	Out-of-Domain AURC↓
BASE	0.54	0.41
RLVR	0.46	0.42
RLCR	0.43	0.40
RLSR	0.41	0.38

D.2. Statistical Significance of RLSR over RLCR and RLVR

Table 5. The AURC (mean $\pm$ std) of Qwen2.5-7B trained on HotPotQA over 5 seeds at $T = 0.3$. Orange highlights the best result.

Method	In-Domain AURC↓	Out-of-Domain AURC↓
RLVR	0.537 ± 0.008	0.453 ± 0.006
RLCR	0.509 ± 0.012	0.443 ± 0.009
RLSR	0.478 ± 0.006	0.412 ± 0.009

Our main results in Tab. 1 are reported using greedy decoding (temperature=0), where it is (almost) deterministic. To obtain variances and verify that the improvement of RLSR over the baselines is not due to sampling noise, we re-run all methods with 5 seeds at temperature 0.3 on HotPotQA. The non-overlapping confidence intervals in Tab. 5 confirm the statistical significance of RLSR's improved performance.

D.3. Comparison with Supervised Finetuning (SFT) Baseline

To isolate the effect of RL training, we fine-tune Qwen2.5-7B on (question, answer) pairs from HotPotQA without chain-of-thought or confidence output because there is no groundtruth chain-of-thought and confidence. Since the SFT model cannot output verbalized confidence (VC), we use self-consistency (SC) with temperature=0.7 and samples $K = 8$ as a post-hoc confidence selector, where the fraction of completion agreement serves as the confidence score. We observe that SFT + SC (0.51) matches RLCR and outperforms RLVR, confirming that SC is a strong post-hoc selector. However, RLSR still outperforms it by 0.07 in AURC with a single inference pass, demonstrating the advantage of end-to-end RL training over supervised alternatives.

Table 6. Non-RL baseline comparison on HotPotQA with Qwen2.5-7B. SFT uses self-consistency (SC) as selector; RL methods use verbalized confidence (VC). Orange highlights the best result.

Method	Selector	ID AURC↓
Base	VC	0.60
SFT-Answer-Only	SC (K=8)	0.51
RLVR	VC	0.56
RLCR	VC	0.51
RLSR	VC	0.44

D.4. RLSR is Agnostic to the Choice of Confidence Selector

Table 7. Confidence scorer ablation on HotPotQA using Qwen2.5-7B. Orange highlights the best result.

Train Selector	Inference Selector	In-Domain AURC↓
VC	VC	0.44
Log-prob	Log-prob	0.50
Base with VC		0.60
SFT-Answer-Only with SC (K=8)		0.51

As the community has not converged to a standard for LLM confidence quantification (Geng et al., 2024; Xia et al., 2025; Rabanser & Papernot, 2025) and *the choice of confidence scorers is orthogonal to our goal of optimizing for AURC*, we have focused on verbalized confidence (VC) for its simplicity and popularity. However, it does not imply that RLSR is bound to VC. Tab. 7 shows that the use of length-normalized log-probability (Log-prob) can also meaningfully improve AURC compared to baselines.

VC can be seen as a learning-based scorer, because it is part of the sampled response and is hence trained end-to-end by the same policy gradient. It is more lightweight than Chen et al. (2023); Mao et al. (2025) because it does not require additional components added to the model architecture. External post-hoc confidence scorers such as those based on logits (e.g., length normalized log-probability) or Monte Carlo simulations (e.g., self-consistency) (Geng et al., 2024; Xia et al., 2025) do not enjoy this property.

D.5. Per-Benchmark Performance

Tab. 1 reports the average performance on the 8 benchmark datasets. Here, we include the performance per-benchmark for Qwen2.5-7B trained on HotPotQA (in-domain in Tab. 8 and out-of-domain in Tab. 9) and BigMath (in-domain in Tab. 10 and out-of-domain in Tab. 11) for completeness.

Table 8. The in-domain results of Qwen2.5-7B trained on HotPotQA.

Dataset	Method	AURC↓	ECE↓	Acc@10↑	Acc@25↑	Acc@50↑
HotPotQA	BASE	0.64	0.60	44.0	40.0	34.8
	RLVR	0.63	0.62	38.0	36.0	34.4
	RLCR	0.55	0.03	48.0	46.4	47.2
	RLSR	0.44	0.20	76.0	68.0	55.2
HotPotQA-Vanilla	BASE	0.57	0.54	48.0	41.2	40.8
	RLVR	0.50	0.51	53.0	48.0	49.2
	RLCR	0.46	0.07	59.0	51.6	53.6
	RLSR	0.44	0.27	56.0	58.4	59.2

Table 9. The out-of-domain results of Qwen2.5-7B trained on HotPotQA.

Dataset	Method	AURC↓	ECE↓	Acc@10↑	Acc@25↑	Acc@50↑
GSM8K	BASE	0.24	0.22	78.8	78.8	74.2
	RLVR	0.21	0.20	78.8	80.0	79.2
	RLCR	0.17	0.33	80.3	83.6	84.1
	RLSR	0.16	0.11	81.8	85.8	87.9
MATH-500	BASE	0.51	0.50	50.0	51.2	46.4
	RLVR	0.56	0.55	42.0	43.2	42.4
	RLCR	0.56	0.06	42.0	45.6	46.8
	RLSR	0.45	0.32	66.0	57.6	53.2
CommonsenseQA	BASE	0.08	0.01	92.7	92.8	92.5
	RLVR	0.13	0.11	86.2	86.3	88.4
	RLCR	0.08	0.40	95.1	92.5	91.7
	RLSR	0.09	0.14	94.3	92.8	90.8
GPQA	BASE	0.55	0.48	53.3	49.1	44.2
	RLVR	0.60	0.60	44.4	39.3	39.7
	RLCR	0.66	0.15	24.4	32.1	37.9
	RLSR	0.58	0.35	46.7	45.5	42.4
SimpleQA	BASE	0.86	0.81	14.5	13.8	13.6
	RLVR	0.89	0.89	9.7	11.7	11.1
	RLCR	0.88	0.36	11.8	11.4	11.5
	RLSR	0.85	0.50	12.9	14.1	14.8
TriviaQA	BASE	0.40	0.37	65.0	58.6	58.6
	RLVR	0.42	0.42	54.5	56.8	57.8
	RLCR	0.38	0.09	67.5	60.8	60.0
	RLSR	0.34	0.27	70.5	68.4	67.8

Table 10. The in-domain results of Qwen2.5-7B trained on BigMath.

Dataset	Method	AURC↓	ECE↓	Acc@10↑	Acc@25↑	Acc@50↑
BigMath	BASE	0.49	0.46	48.0	54.4	53.0
	RLVR	0.37	0.32	56.0	63.2	64.6
	RLCR	0.32	0.07	69.0	69.6	68.0
	RLSR	0.24	0.19	79.0	80.8	75.2
GSM8K	BASE	0.24	0.22	78.8	78.8	74.2
	RLVR	0.12	0.10	88.6	87.3	88.2
	RLCR	0.13	0.27	87.1	86.4	86.5
	RLSR	0.12	0.07	89.4	90.3	88.0
MATH-500	BASE	0.51	0.50	50.0	51.2	46.4
	RLVR	0.36	0.34	72.0	61.6	61.6
	RLCR	0.33	0.03	74.0	70.4	66.4
	RLSR	0.36	0.29	68.0	64.0	66.0

Table 11. The out-of-domain results of Qwen2.5-7B trained on BigMath.

Dataset	Method	AURC↓	ECE↓	Acc@10↑	Acc@25↑	Acc@50↑
HotPotQA-Vanilla	BASE	0.57	0.54	48.0	41.2	40.8
	RLVR	0.53	0.53	51.0	44.8	46.6
	RLCR	0.47	0.12	63.0	55.2	53.2
	RLSR	0.49	0.36	55.0	55.6	51.8
CommonsenseQA	BASE	0.08	0.01	92.7	92.8	92.5
	RLVR	0.15	0.07	83.7	83.7	84.6
	RLCR	0.11	0.29	90.2	89.9	88.9
	RLSR	0.09	0.10	91.1	91.8	92.6
GPQA	BASE	0.55	0.48	53.3	49.1	44.2
	RLVR	0.59	0.59	42.2	37.5	40.2
	RLCR	0.57	0.18	48.9	45.5	39.7
	RLSR	0.53	0.35	53.3	48.2	45.1
SimpleQA	BASE	0.86	0.81	14.5	13.8	13.6
	RLVR	0.89	0.86	10.9	10.6	10.4
	RLCR	0.88	0.44	12.9	12.2	13.1
	RLSR	0.87	0.65	14.5	12.5	12.4
TriviaQA	BASE	0.40	0.37	65.0	58.6	58.6
	RLVR	0.50	0.40	44.0	43.0	52.0
	RLCR	0.33	0.04	71.5	66.4	65.6
	RLSR	0.30	0.22	78.5	73.6	69.1