Calibrating LLMs with Semantic-level Reward

Fengfei Yu[1] * **Ruijia Niu**[1,2] * **Dongxia Wu**[3,‡] **Yian Ma**[2,1] **Rose Yu**[1,2,†]
[1]Department of Computer Science and Engineering, University of California San Diego
[2]Halıcıoğlu Data Science Institute, University of California San Diego
[3]Mohamed bin Zayed University of Artificial Intelligence

Abstract

As large language models (LLMs) are deployed in consequential settings such as medical question answering and legal reasoning, the ability to estimate when their outputs are likely to be correct is essential for safe and reliable use and requires well-calibrated uncertainty. Standard reinforcement learning with verifiable rewards (RLVR) trains models with a binary correctness reward that is indifferent to confidence, providing no penalty for confident but wrong predictions and thereby degrading calibration. Recent work addresses this by training models to produce verbalized confidence scores alongside answers and rewarding agreement with correctness. However, verbalized confidence is tied to surface text and thus varies across textual variations with the same semantic meaning. We propose **Calibration with Semantic Reward (CSR)**, a framework that calibrates language models directly in semantic space without a verbalized confidence interface. CSR is a training-time policy intervention: instead of eliciting a confidence score, it shapes the policy so that a fixed post-hoc semantic-uncertainty estimator becomes reliable. CSR combines the correctness reward with a novel semantic calibration reward that encourages exploitation among correct rollouts by promoting semantic agreement, and exploration among incorrect ones by discouraging spurious consistency. Across three model families and four open-ended QA datasets, one in-distribution and three out-of-distribution, CSR lowers four-dataset average ECE by up to 57% and raises average AUROC by up to 12% over the base model, and by up to 45% and 23% over the verbalized-confidence baselines. The code is available at `https://github.com/Rose-STL-Lab/Semantic_Calibration`.

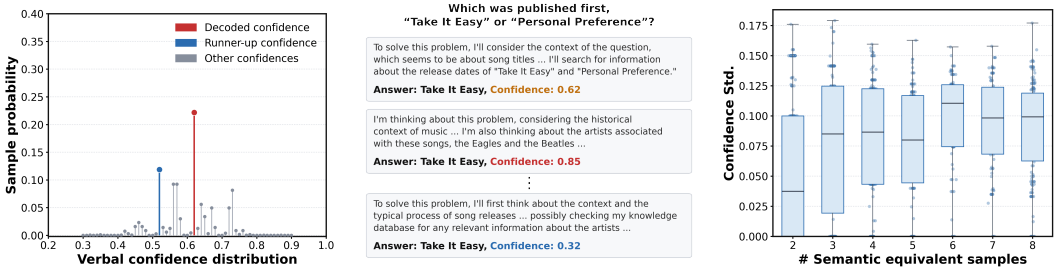
1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become the dominant training paradigm for reasoning language models, achieving strong performance across mathematical reasoning and multi-step inference [20, 12, 34]. The standard RLVR objective uses a binary correctness reward: a rollout receives a positive signal when the generated answer matches the ground truth and zero otherwise. While effective for improving accuracy, this formulation does not account for model confidence in the generated answer. A model that guesses incorrectly with high confidence receives the same signal as one that abstains. In high-stakes deployment settings such as medical question answering and legal reasoning, a reliable system must not only produce correct answers but also express calibrated uncertainty to support sound downstream decisions.

Empirically, RL training with binary correctness rewards is known to worsen calibration and increase hallucination rates [4]. Recent work addresses this by training models to express confidence alongside each answer, as an explicit score or a hedged utterance, and optimizing its agreement with empirical correctness through RL or supervised losses. Methods such as Rewarding Doubt (RD) [2], RLCR [4],

*Equal contribution. †Corresponding author: Rose Yu (`roseyu@ucsd.edu`). ‡Work done during post-doc at Stanford.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Single-sample distribution (b) Inconsistency (c) Variability

Figure 1: Verbalized confidence is not always stable. (a) For a single greedy-decoded answer, the decoded confidence is uncertain, with alternative values appearing with substantial probability. (b) Multiple responses expressing the same answer yield inconsistent confidence scores. (c) Within groups of semantically equivalent rollouts, verbalized confidence still varies. All panels use Llama-3.1-8B-Instruct fine-tuned with RLCR. Setup details are provided in Appendix B.1.

SaySelf [36], ConfTuner [22], and LACIE [29] operate within this paradigm, attaching the training signal to a verbalized confidence that the model generates when prompted.

However, existing work has identified verbalized confidence as an unreliable proxy for model uncertainty [35, 10]. The reported score depends on the answer text and prompt format rather than the distribution of meanings produced under free sampling. Fig. 1 illustrates this under RLCR. Panel (a) shows that even for a single greedy answer the decoded confidence is not stable. Panel (b) shows that responses giving the same answer report inconsistent scores, so verbalized confidence varies across semantically equivalent outputs. Panel (c) shows the same effect at scale: among rollouts grouped by semantic agreement, confidence still varies widely and fails to track the degree of agreement. Together, these observations suggest that existing methods calibrate a reporting interface rather than the model's underlying uncertainty over answer meanings.

In this work, we propose to calibrate language models directly in semantic space, bypassing the verbalized confidence interface. We frame calibration in terms of *semantic-level uncertainty*: the distribution of meanings the model produces when sampled repeatedly from the same question. Semantic agreement is informative of correctness when correct answers form one concentrated semantic cluster and incorrect answers are dispersed. A correctness reward alone does not produce this structure, because it gives every incorrect rollout the same zero reward whether or not the wrong answers agree with each other. The model can thus learn to repeat one wrong answer, which then appears as confident as a correct one. We introduce **Calibration with Semantic Reward (CSR)**, a fine-tuning framework that trains the policy toward this structure: alongside the correctness reward, a semantic calibration reward, added with a scheduled weight, rewards each correct rollout for agreeing with its siblings and each incorrect rollout for disagreeing with them. Confidence is then read off the answer distribution post hoc with semantic entropy [6], without any confidence-reporting prompt, and we show that it becomes both better calibrated (ECE) and more discriminative (AUROC). This confidence can be used directly to abstain on questions the model is likely to get wrong, without changing what the model outputs.

To summarize, our contributions include:

- **Semantic calibration as a training target**: instead of eliciting a confidence score, we train the policy so that the confidence derived from its semantic answer distribution becomes better calibrated, providing a principled alternative to verbalized confidence-reporting interfaces.
- **CSR**, a reinforcement learning framework that adds to the correctness reward a semantic calibration reward, which rewards agreement among correct rollouts and disagreement among incorrect ones using the GRPO rollout group at no extra sampling cost. An exact decomposition (Proposition 1) shows why the correctness term is needed.
- **Empirical validation** across three model families showing that CSR reduces four-dataset average ECE by up to 57% and improves average AUROC by up to 12% over the base model, and by up to 45% and 23% over the verbalized-confidence baselines, while maintaining competitive accuracy and generalizing to out-of-distribution datasets.

2 Related work

Sampling-based uncertainty estimation. Post-hoc methods read uncertainty off the output distribution without modifying the model or adding a confidence interface. Token-level proxies derive confidence from generation likelihoods [25], while sampling-based proxies treat it as dispersion across repeated draws, as in self-consistency [33], black-box similarity measures [23], and semantic entropy [6], which clusters rollouts by semantic equivalence and is therefore invariant to paraphrastic variation. Semantic Density [27] and SIMBA UQ [3] refine sampling-based confidence scores using semantic similarities among sampled responses.

Training for uncertainty calibration. Verbalized reporting is only *one* interface for exposing confidence and, as discussed in the introduction, an unreliable one [35, 39, 10]. A line of work trains this interface directly, through RL or supervision [36, 2, 4, 22, 29, 41, 17]. Closest to us in signal, FUT [5] and the concurrent RLMF [24] derive their target from sampled-response consistency, but still train an *expression* of it, a stated confidence or hedge, on top of a largely preserved answer distribution. CSR supervises no reporting interface at all: it combines each rollout's consistency with its correctness *before* the group-relative advantage is formed (Sec. 4.3), so consistency changes rollout credit and reshapes the distribution from which confidence is derived. A method-by-method comparison is provided in Appendix A.

3 Preliminaries

Reinforcement learning with verifiable rewards (RLVR). Let π_θ be a language model mapping prompts $x \in \mathcal{X}$ to outputs $Y \in \mathcal{Y}$. Given a dataset $\mathcal{D}$ of question-answer pairs (x, Y^*) and a reward function $R : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, RLVR maximizes the expected reward under the policy [20, 34]:

$$\arg\max_\theta \; \mathbb{E}_{(x,Y^*)\sim\mathcal{D},\, Y\sim\pi_\theta(\cdot|x)} \left[R(Y, Y^*)\right]. \tag{1}$$

The standard choice of R is a binary correctness reward, 1 if the generated answer Y is judged equivalent to the gold answer Y^* and 0 otherwise. This reward is effective for improving accuracy but provides no calibration signal, as it is indifferent to the confidence or consistency with which the model produces an answer.

Semantic entropy. Semantic entropy [6] is a sampling-based method for estimating uncertainty in language model outputs. Given a question x, K responses $\{Y^{(1)}, \ldots, Y^{(K)}\}$ are sampled from $\pi_\theta(\cdot \mid x)$ and partitioned into semantic-equivalence classes $\{\mathcal{M}_m\}_{m=1}^M$ by a semantic-equivalence judge $J(Y^{(i)}, Y^{(j)}) \in \{0, 1\}$. Since pairwise verdicts need not be transitive, we follow Farquhar et al. [6]: responses are visited in order, and each joins the first existing class whose first member it is judged equivalent to, or otherwise starts a new class. The probability of each semantic class is estimated empirically as

$$\hat{P}(\mathcal{M}_m \mid x) = \frac{1}{K} \sum_{j=1}^{K} \mathbf{1}\left[Y^{(j)} \in \mathcal{M}_m\right]. \tag{2}$$

Semantic entropy is then the Shannon entropy over the class distribution:

$$\mathrm{SE}(x) = -\sum_{m=1}^{M} \hat{P}(\mathcal{M}_m \mid x) \log \hat{P}(\mathcal{M}_m \mid x). \tag{3}$$

By operating over semantic classes rather than individual tokens or surface forms, SE is invariant to paraphrastic variation. High $\mathrm{SE}(x)$ indicates semantically diverse outputs and high uncertainty, and low $\mathrm{SE}(x)$ indicates consistent semantic agreement across rollouts. We use $\mathrm{SE}(x)$ as our post-hoc uncertainty estimate and convert it into a *derived semantic confidence* $\hat{s}(x) = \exp(-\mathrm{SE}(x))$.

Definitions. We follow standard uncertainty quantification (UQ) terminology [11, 8, 23]. $\mathrm{SE}(x)$ is an *uncertainty*, a property of the question's predictive distribution, and $\hat{s}(x)$ is a question-level UQ score oriented toward correctness, which makes it a *confidence*: higher values indicate that the policy's answers are more likely correct. A fixed score is *calibrated* if, among questions with score c, the policy's accuracy is c, which ECE measures, and *discriminative* if it ranks questions the policy answers correctly above those it does not, which AUROC measures. *Semantic calibration* refers to calibration of $\hat{s}$ in this standard sense. Formal definitions are provided in Appendix B.2.

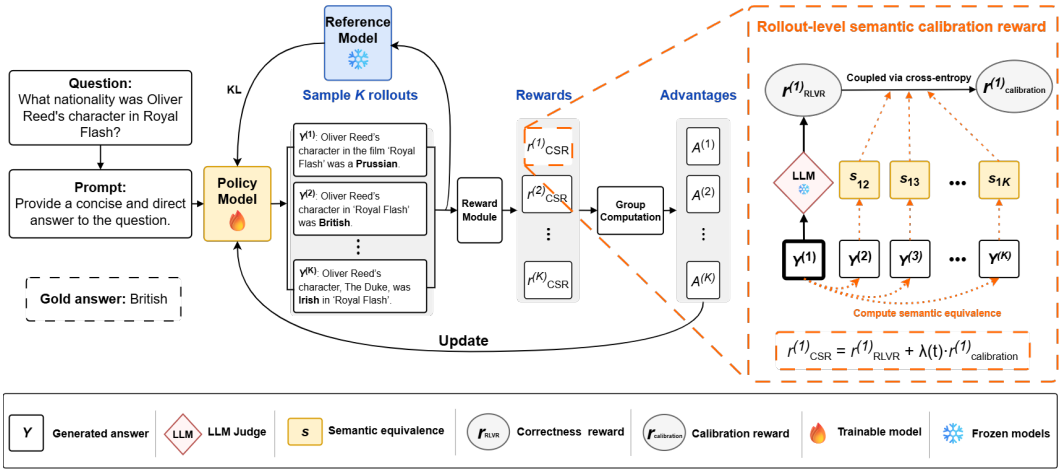


Figure 2: Overview of CSR. For each question x, we draw K rollouts from the policy π_θ, score each rollout with a verifiable correctness reward r_{RLVR} and a semantic calibration reward $r_{\text{calibration}}$ (Eq. (5)), and update the policy via group-relative advantages. CSR concentrates correct rollouts into a tight semantic cluster while keeping incorrect rollouts dispersed, so that semantic agreement among sampled outputs becomes an informative signal of correctness.

4 Calibration with Semantic Reward (CSR)

We present *Calibration with Semantic Reward* (CSR), a framework that calibrates a language model directly in semantic space without introducing any explicit confidence variable (Fig. 2). The central idea is to make *semantic agreement among sampled rollouts informative of correctness*, so that correct answers are supported by consistent rollouts while incorrect answers are not reinforced by spurious agreement. We realize this through two complementary reward signals, a *verifiable correctness reward* that shifts probability mass toward correct semantic modes, and a *semantic calibration reward* (Sec. 4.1) that shapes the agreement structure across rollouts so that it tracks correctness. We combine them through a curriculum learning objective (Sec. 4.2) and optimize with group-relative policy optimization (Sec. 4.3).

Setup and notation. We keep the notation of Sec. 3, and the gold answer Y^* may be a set of acceptable answers. For a group of K rollouts $Y^{(1)}, \ldots, Y^{(K)} \sim \pi_\theta(\cdot \mid x)$ we write $Q_j = J(Y^{(j)}, Y^*)$ for the correctness of rollout j and $s_{ji} = J(Y^{(j)}, Y^{(i)})$ for the agreement of rollouts j and i. The judge J can be instantiated as exact match or thresholded token-level F1 for short factual QA, or as an LLM judge for tasks with paraphrastic variation.

4.1 Reward definitions

Correctness reward. The verifiable correctness reward of rollout j is its judged correctness,

$$r^{(j)}_{\text{RLVR}} \;=\; Q_j \;\in\; \{0, 1\}. \tag{4}$$

Semantic calibration reward. We treat semantic consistency as an *implicit proxy for confidence*: rollouts that concentrate on one meaning signal high semantic confidence, while rollouts that spread across meanings signal low confidence. Calibration in semantic space then amounts to making this implicit confidence track correctness, with consistency high for correct rollouts and low for incorrect ones under the same question.

We turn this principle into a per-rollout *semantic calibration reward* by comparing each rollout's pairwise agreement with the rest of the group against its correctness with respect to the gold answer:

$$r^{(j)}_{\text{calibration}} = -\frac{1}{K-1} \sum_{i \neq j} \text{CE}(s_{ji}, Q_j), \tag{5}$$

where $\text{CE}(\cdot, \cdot)$ is binary cross-entropy and the binary agreement s_{ji} is clipped to $[\epsilon, 1-\epsilon]$ before the cross-entropy is computed. Each summand is maximized exactly when pairwise agreement

matches correctness, so a correct rollout is rewarded for agreeing with its siblings and an incorrect rollout for disagreeing with them. For i.i.d. rollouts, Eq. (5) thus pushes the policy-level agreement probability $p_j = \mathbb{P}_{Y' \sim \pi_\theta(\cdot|x)}(J(Y^{(j)}, Y') = 1 \mid Y^{(j)})$ toward Q_j: correct answers concentrate into one semantic cluster, while incorrect rollouts that agree with one another are penalized relative to dispersed ones, which promotes exploration away from concentrated incorrect modes.

4.2 CSR objective

Limitation of the calibration reward alone. Eq. (5) shapes agreement across rollouts but does not raise the fraction of correct rollouts. Proposition 1 makes this exact.

Proposition 1 (Exact decomposition of the calibration reward)**.** *Let $\{Y^{(j)}\}_{j=1}^{K}$ be drawn i.i.d. from $\pi_\theta(\cdot \mid x)$, let $\alpha = K^{-1}\sum_j Q_j$ be the empirical accuracy of the group, let $\hat{p}_j = (K-1)^{-1}\sum_{i \neq j} s_{ji}$ be the empirical agreement rate of rollout j, let p_j be the policy-level agreement probability defined above, and write $c_\epsilon = \log \frac{1-\epsilon}{\epsilon}$. Then, exactly,*

$$r^{(j)}_{\text{calibration}} = \log \epsilon + c_\epsilon \big[Q_j \hat{p}_j + (1 - Q_j)(1 - \hat{p}_j)\big], \qquad \mathbb{E}\big[\hat{p}_j \mid Y^{(j)}\big] = p_j,$$

and the group-average calibration reward decomposes as

$$\bar{r}_{\text{calibration}} = \log \epsilon + c_\epsilon \Big[\alpha\, \bar{p}_{\mathcal{C}} + (1 - \alpha)\big(1 - \bar{p}_{\mathcal{I}}\big)\Big],$$

where $\bar{p}_{\mathcal{C}}$ and $\bar{p}_{\mathcal{I}}$ average $\hat{p}_j$ over the correct and the incorrect rollouts (a term whose set is empty is omitted). Consequently, when $\alpha \to 0$, the reward is dominated by the second term and is maximized by driving $p_j \to 0$, and when $\alpha \to 1$, it is maximized by driving $p_j \to 1$. Neither regime directly rewards a change in correctness, so when accuracy is low the calibration reward alone can shape agreement structure while correctness stagnates.

Early in training, when α is small, the calibration reward alone therefore favors maximally diverse, mostly incorrect responses, and when α is close to 1 it favors collapse onto a single answer. In neither regime does it reward switching to a correct answer, so confidence can become discriminative while accuracy stagnates or falls, as in Fig. 5(a). A detailed proof is provided in Appendix C.1.

Curriculum learning objective. Building on the analysis above, we define the reward at step t as

$$r^{(j)}_{\text{CSR}}(t) \;=\; r^{(j)}_{\text{RLVR}} \;+\; \lambda(t)\, r^{(j)}_{\text{calibration}}, \tag{6}$$

where $\lambda(t) \geq 0$ controls the strength of the semantic calibration signal at training step t. In the implementation, $r^{(j)}_{\text{calibration}}$ enters in its normalized form $Q_j \hat{p}_j + (1 - Q_j)(1 - \hat{p}_j) \in [0, 1]$, obtained from Proposition 1 by dropping the constant $\log \epsilon$ and the scale c_ϵ, so $\lambda(t)$ weights a reward on the same $[0, 1]$ scale as r_{RLVR}. The correctness term supplies the accuracy incentive that the calibration term lacks, while the calibration term organizes the agreement structure across rollouts. We set $\lambda(t)$ as a curriculum that increases over training, so that the calibration weight starts below the correctness weight, when accuracy is low and the calibration term cannot raise it, and ends above it as correct rollouts become common.

4.3 Optimization with GRPO

We optimize Eq. (6) with group-relative policy optimization (GRPO) [28], which samples K rollouts per input and computes advantages from their relative rewards within the group. This group is exactly what the calibration reward needs to measure pairwise agreement, so CSR adds no sampling overhead, only the cost of judging the rollouts. Because the calibration reward is added to the correctness reward before the group-relative normalization, it changes which rollouts receive credit within a group: among correct rollouts, more agreement earns a larger advantage, and among incorrect ones, less. We discuss the regularization effect of CSR in the GRPO setting in Appendix C.2.

Alignment with post-hoc estimators. For a sampling-based online RL algorithm such as GRPO, the rollouts drawn for each question provide a natural environment for the optimization. The sampled answers that define the reward are the same kind of samples that post-hoc estimators read to measure the uncertainty of the answer distribution. The pairwise score of Eq. (5) and semantic entropy are two such uncertainty estimates, so training on one is aligned with, but not identical to, evaluating with the other, and the transfer between them is empirical. Because the estimator stays fixed, CSR improves calibration and discrimination by changing the policy, which we test with semantic entropy and four other estimators (Secs. 5.2 and 5.3).

Table 1: Calibration results across three model families and four QA datasets. HotpotQA is in distribution and the other three are out of distribution. Confidence is the derived semantic confidence for Base, RLVR, and CSR and the verbalized confidence for RD and RLCR. Average is the four-dataset macro-average, and Tok counts all K rollouts. **Bold** marks the best and underline the second-best value in each column within each model block.

		In-distribution			Out-of-distribution									Average		
		HotpotQA			TriviaQA			MSMARCO			NQ-Open					
Model	Method	Acc↑	ECE↓	AUROC↑	Acc↑	ECE↓	AUROC↑	Acc↑	ECE↓	AUROC↑	Acc↑	ECE↓	AUROC↑	Tok	ECE↓	AUROC↑
Llama	Base	0.1881	0.2949	0.7050	0.6682	0.0772	0.8423	**0.3581**	0.1996	0.7062	0.4494	0.1186	0.7361	839.93	0.1726	0.7474
	RLVR	**0.3079**	0.2567	0.8089	**0.6850**	0.0901	0.8268	0.3514	0.2280	0.7371	**0.4512**	0.1686	0.6957	605.45	0.1859	0.7671
	RD	0.2570	0.1764	0.8316	0.6600	0.0746	0.7996	0.2820	0.3302	0.7298	0.3540	0.2081	0.7464	249.01	0.1973	0.7769
	RLCR	0.2820	0.1757	0.6923	0.6490	0.0943	0.7337	0.2940	0.2340	0.6816	0.3950	0.1497	0.7013	784.71	0.1634	0.7022
	CSR	0.2700	**0.0978**	**0.9133**	0.6668	**0.0670**	**0.8976**	0.3390	**0.1571**	**0.7594**	0.4461	**0.1100**	**0.7670**	663.85	**0.1080**	**0.8343**
Qwen	Base	0.2364	0.3050	0.8413	0.5016	0.1881	0.8281	**0.3217**	0.2945	0.7037	0.2684	0.2487	0.7773	201.67	0.2591	0.7876
	RLVR	**0.2553**	0.3007	0.8365	0.5072	0.1933	0.8241	0.3201	0.2871	0.7094	0.2710	0.2492	0.7594	194.23	0.2576	0.7824
	RD	0.2440	0.1613	0.8235	0.5200	0.0876	0.8013	0.2970	**0.1008**	0.6531	0.2650	0.1380	0.7606	299.52	0.1219	0.7596
	RLCR	0.2100	0.2370	0.7530	**0.5690**	0.0894	0.7293	0.3210	0.2031	0.7026	**0.3240**	0.2671	0.6255	641.07	0.1991	0.7026
	CSR	0.2441	**0.1152**	**0.9260**	0.4914	**0.0855**	**0.9094**	0.3135	0.1560	**0.8001**	0.2657	**0.0866**	**0.8348**	198.54	**0.1108**	**0.8676**
Mistral	Base	0.2457	0.2563	0.7745	**0.6776**	0.0952	0.7860	0.3753	0.3011	0.7087	0.4188	0.1507	0.7035	774.69	0.2008	0.7432
	RLVR	**0.3065**	0.2533	0.7936	0.6751	0.1126	0.7887	0.3683	0.2939	0.7003	0.4201	0.1635	0.7079	634.83	0.2058	0.7476
	RD	0.2460	0.0994	**0.9065**	0.5490	0.1061	0.8118	0.1760	**0.1879**	0.7014	0.1930	0.1395	**0.7897**	166.58	0.1332	0.8024
	RLCR	0.3040	0.2117	0.6715	0.6310	**0.0611**	0.6913	0.3130	0.2669	0.6529	0.3900	0.1942	0.6856	839.80	0.1835	0.6753
	CSR	0.2981	**0.0966**	0.8710	0.6703	0.0751	**0.8530**	**0.3773**	0.2513	**0.7351**	**0.4269**	**0.0974**	0.7606	731.71	**0.1301**	**0.8049**

5 Experiments

We ask whether CSR makes the derived semantic confidence better calibrated and more discriminative than correctness-only and verbalized-confidence training, in and out of distribution (Sec. 5.2), whether the gains hold under other confidence estimators (Sec. 5.3) and in selective prediction (Sec. 5.4), whether a lexical surrogate can remove the cost of the pairwise judge (Sec. 5.5), and which design choices drive the gain (Sec. 5.6). Appendix E shows rollout-level examples for all five methods.

5.1 Experimental setup

Datasets and models. Following prior work [4, 22, 36], we use HotpotQA [40] for training and in-distribution evaluation. To assess generalization, we further evaluate on TriviaQA [16], MS-MARCO [1], and NQ-Open [18, 21], which span trivia, web-search, and open-domain questions. We fine-tune on $10{,}000$ HotpotQA training examples, with $K = 8$ rollouts per question for RLVR and CSR and a sigmoid schedule for $\lambda(t)$ from 0.8 to 1.2 in CSR, and evaluate using $1{,}000$ examples per dataset. We use Llama-3.1-8B-Instruct [9], Qwen2.5-7B-Instruct [37], and Mistral-7B-Instruct-v0.3 [15] (Llama, Qwen, and Mistral) as base models. Training details are provided in Appendix B.1.

Evaluation metrics. We evaluate model performance and calibration quality with accuracy, expected calibration error (ECE) [11], area under the ROC curve (AUROC) [7], and token cost (Tok). ECE and AUROC measure calibration and discrimination in the sense of Sec. 3, and Tok is the average number of prompt and generated tokens per question. Every method, including RD and RLCR, is scored by the same GPT-4.1 nano correctness judge. At evaluation, Base, RLVR, and CSR are scored on $K = 16$ sampled rollouts, with accuracy averaged over the rollouts and a question counted as correct for AUROC when at least half of its rollouts are correct. RD and RLCR are scored on a single greedy answer. Metric definitions are provided in Appendix B.3.

Baselines. We compare against four baselines. **Base** is the unmodified instruction-tuned model, and **RLVR** fine-tunes with a binary correctness reward and no calibration signal. For both, uncertainty is estimated post hoc via semantic entropy, as for CSR. **RD** [2] and **RLCR** [4] are verbalized-confidence methods: RD optimizes only its verbalized confidence with a clipped logarithmic scoring rule, while RLCR combines a binary correctness reward with a Brier-score calibration term. Both require an explicit confidence-reporting interface with additional format constraints. All methods are evaluated closed-book.

5.2 Does CSR improve calibration and discrimination?

Table 1 reports accuracy, ECE, AUROC, and token cost on the in-distribution HotpotQA set and three out-of-distribution (OOD) sets for three model families. The ROC curves for Qwen (Fig. 3) show

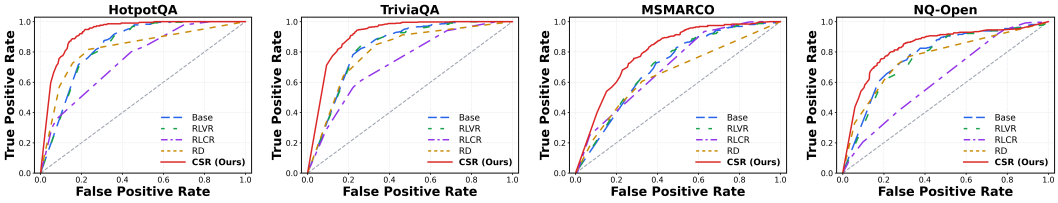


Figure 3: ROC curves on Qwen for HotpotQA (in-distribution) and three OOD datasets. Confidence is the derived semantic confidence for Base, RLVR, and CSR and the verbalized confidence for RD and RLCR. CSR has the highest AUROC in all four panels.

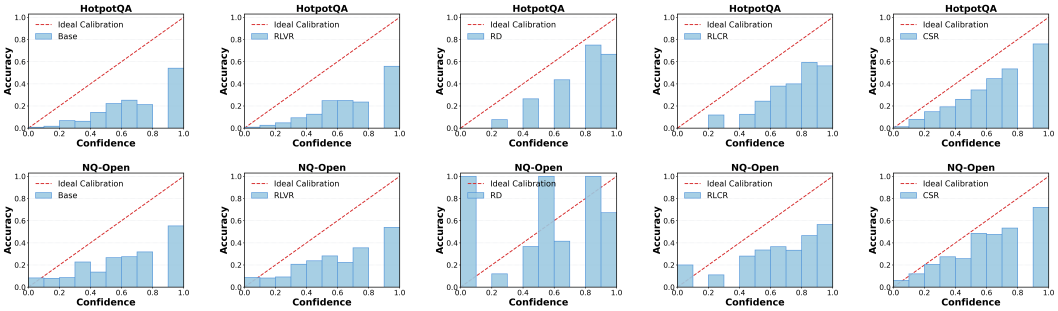


Figure 4: Reliability diagrams on Qwen. Rows are HotpotQA (in-distribution) and NQ-Open (out-of-distribution), and columns are Base, RLVR, RD, RLCR, and CSR. Each bar is the accuracy of the questions whose confidence falls in a bin of width 0.1.

that CSR's advantage is concentrated at low false-positive rates, the region that matters when only high-confidence answers are accepted: on TriviaQA, for example, CSR reaches a true-positive rate of about 0.7 at a false-positive rate of 0.1, while every other method stays below 0.4. Its reliability diagrams on HotpotQA and NQ-Open also lie closest to the diagonal (Fig. 4), whereas those of Base and RLVR fall well below it.

Calibration in and out of distribution. On HotpotQA, CSR has the lowest ECE of all five methods on every backbone, the highest AUROC on Llama and Qwen, and cuts ECE by 62–67% relative to Base. On Llama, ECE falls from 0.2949 to 0.0978. The gains carry over to the three OOD datasets, where CSR has the highest AUROC in eight of the nine model-dataset cells and lower ECE than Base and RLVR in all nine. A verbalized baseline has lower ECE in three OOD cells, but CSR keeps the lowest average ECE on every backbone. RLVR reaches higher HotpotQA accuracy but improves ECE far less. On Llama, where its accuracy lead is largest at 0.038, its ECE is 0.2567, against 0.0978 for CSR. CSR's calibration gain is thus not a by-product of accuracy, and its four-dataset accuracy is within 0.02 of RLVR's on every backbone (Appendix D.2). CSR therefore learns semantic uncertainty structure that transfers beyond the training domain, which we attribute to supervising meaning rather than a reporting format. RD's low accuracy on Mistral comes in part from abstention. Its Mistral policy declines to answer 34.5% of HotpotQA and 27.6% of NQ-Open questions, and each abstention is scored as incorrect (Appendix D.10).

Joint calibration and discrimination. Averaged over datasets, CSR is the only method that both lowers ECE and raises AUROC relative to Base on every backbone. Low ECE alone is not enough, since a near-constant confidence equal to the average accuracy has low ECE but cannot separate correct from incorrect answers, and RLCR drifts toward this regime. RLCR lowers average ECE relative to Base but also lowers AUROC on every backbone, as RD does on Qwen. In two of the three OOD cells where a verbalized baseline has lower ECE than CSR, its AUROC trails CSR's by 0.15 to 0.16. CSR improves both because its confidence is computed from answer agreement, which training ties to correctness, rather than verbalized as a separate output.

Inference cost. CSR's K rollouts are answer-only, so its Tok stays comparable to Base and RLVR and is lower than RLCR's on all three backbones, by up to $3.2\times$ on Qwen. Beyond generation, CSR, like Base and RLVR, needs pairwise equivalence judgments to form semantic clusters, a cost we address in Sec. 5.5. Further token-cost analysis is provided in Appendix D.8.

Table 2: Four-dataset average ECE and AUROC under four sampling-based estimators not used in training, applied to the rollouts of Table 1. **Bold** marks the best value in each column per estimator.

		Llama		Qwen		Mistral	
Estimator	**Method**	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑
SemDens	Base	0.1998	0.7438	0.2529	0.7872	0.2331	0.7415
	RLVR	0.2090	0.7674	0.2523	0.7820	0.2336	0.7474
	CSR	**0.1251**	**0.8378**	**0.1195**	**0.8691**	**0.1517**	**0.8049**
Bayes-beta	Base	0.1950	0.7182	0.3047	0.7559	0.2214	0.6611
	RLVR	0.2249	0.7322	0.3145	0.7528	0.2169	0.6822
	CSR	**0.1585**	**0.7895**	**0.1672**	**0.8383**	**0.1986**	**0.6885**
Arithmetic	Base	0.1121	0.7169	0.2851	0.7562	0.1380	0.6655
	RLVR	0.1516	0.7378	0.2798	0.7583	0.1358	0.6885
	CSR	**0.0756**	**0.7985**	**0.1063**	**0.8428**	**0.1072**	**0.6918**
Clf-pairs+gen	Base	**0.0287**	0.7497	0.1153	0.6987	0.0388	0.6555
	RLVR	0.0618	0.7093	0.1294	0.6756	0.0469	0.6641
	CSR	0.0312	**0.7919**	**0.0605**	**0.8160**	**0.0277**	**0.7127**

5.3 Does CSR remain effective across sampling-based estimators?

CSR is evaluated with semantic entropy in Sec. 5.2, so we test whether its gains depend on that estimator. We apply four other post-hoc estimators to the same Base, RLVR, and CSR rollouts, which cover the main design choices of sampling-based estimation. Semantic Density (SemDens) [27] measures agreement through semantic equivalence and weights answers by length-normalized sequence likelihood. Three SIMBA UQ aggregations [3] instead measure agreement by word-level Jaccard similarity, without the pairwise judge, and aggregate it with a fixed rule (Arithmetic) or a fitted model (Bayes-beta, and Clf-pairs+gen, a supervised random-forest classifier).

Robustness to estimator design. CSR has lower average ECE and higher average AUROC than both Base and RLVR in 23 of the 24 estimator-backbone-metric comparisons (Table 2), and the highest average AUROC under every estimator. The only exception is ECE on Llama under Clf-pairs+gen, where CSR's 0.0312 is above Base's 0.0287 and all three methods are well calibrated. The gain therefore holds whether agreement is judged semantically or lexically and whether the aggregation is fixed or fitted to each policy. Because the estimators are not used in training and only the sampled answers change, the gain comes from the answer distribution CSR learns rather than from one way of reading it. Definitions and per-dataset results are provided in Appendix D.1.

5.4 Does CSR improve selective prediction?

A common use of confidence is to answer only when confident and abstain otherwise, so that the model refuses the questions it is likely to get wrong. Ranking questions by each method's confidence, we report AURAC [6], the fraction of correct questions among those retained, integrated over coverage, and AUGRC [30], the fraction of all questions that are both accepted and answered incorrectly, integrated over coverage. CSR has the best average AURAC and AUGRC on all three backbones (Table 3) and the best value in 22 of the 24 per-dataset cells. Its average AURAC leads the strongest baseline by 0.014 (Llama) to 0.048 (Qwen). RLCR is worse than Base in both metrics on every backbone, consistent with its lower AUROC (Table 1).

Table 3: Selective prediction averaged over the four datasets. **Bold** marks the best value in each column.

	AURAC↑			AUGRC↓		
Method	Llama	Qwen	Mistral	Llama	Qwen	Mistral
Base	0.5689	0.4986	0.5758	0.2357	0.2755	0.2320
RLVR	0.6220	0.5070	0.5992	0.2121	0.2708	0.2210
RD	0.5670	0.5053	0.4890	0.2478	0.2802	0.2981
RLCR	0.5363	0.4848	0.5212	0.2533	0.2798	0.2561
CSR	**0.6359**	**0.5546**	**0.6393**	**0.2104**	**0.2602**	**0.2087**

Gains in and out of distribution. Selective prediction tests whether the confidence can decide when the model should answer, which makes it a form of refusal. Here the model declines by thresholding its derived confidence and its outputs are unchanged, whereas in knowledge-aware refusal [26]

Table 4: Token-level F1 versus the LLM judge for pairwise equivalence on Llama. The suffix names the equivalence test used for clustering, and CSR-F1 also uses F1 in training. The -LLM rows, RD, and RLCR repeat Table 1. **Bold** marks the best and underline the second-best value in each column.

	HotpotQA		TriviaQA		MSMARCO		NQ-Open		Average	
Method	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑
RD	0.1764	0.8316	0.0746	0.7996	0.3302	0.7298	0.2081	0.7464	0.1973	0.7769
RLCR	0.1757	0.6923	0.0943	0.7337	0.2340	0.6816	0.1497	0.7013	0.1634	0.7022
Base-LLM	0.2949	0.7050	0.0772	0.8423	0.1996	0.7062	0.1186	0.7361	0.1726	0.7474
RLVR-LLM	0.2567	0.8089	0.0901	0.8268	0.2280	0.7371	0.1686	0.6957	0.1859	0.7671
CSR-LLM	**0.0978**	**0.9133**	**0.0670**	**0.8976**	0.1571	**0.7594**	**0.1100**	0.7670	**0.1080**	**0.8343**
Base-F1	0.2713	0.6681	0.1212	0.7929	0.1612	0.6094	0.1223	0.6992	0.1690	0.6924
RLVR-F1	0.2727	0.7770	0.0899	0.7819	0.2296	0.6670	0.1798	0.6771	0.1930	0.7258
CSR-F1	0.1240	0.9008	0.1005	0.8498	**0.1042**	0.7223	0.1109	**0.7735**	0.1099	0.8116

the model itself declines in its output. AURAC and AUGRC depend only on how the confidence orders the questions, so they are unchanged by any monotone rescaling of the score and measure discrimination rather than calibration. CSR is best on HotpotQA for all three backbones under both metrics and on 16 of the 18 OOD cells. Metric definitions, per-dataset results, and a further discussion of refusal are provided in Appendices B.3 and D.3.

5.5 Can token-level F1 replace the pairwise LLM judge?

Pairwise equivalence calls dominate CSR's judging cost. At evaluation with $K = 16$, they account for 120 of the 136 judgments per question. We ask whether a lightweight lexical proxy can replace the pairwise equivalence judge during both training and post-hoc evaluation. We compare (i) an *LLM judge* of pairwise semantic equivalence and (ii) *token-level F1 thresholding*, which declares two answers equivalent when their F1 reaches a fixed threshold τ. The threshold is chosen once per model family on training rollouts to best agree with the judge (0.50 for Llama and Mistral, 0.80 for Qwen), and correctness is scored by the LLM judge for every method. Table 4 reports Llama results, and Table 12 reports all three backbones.

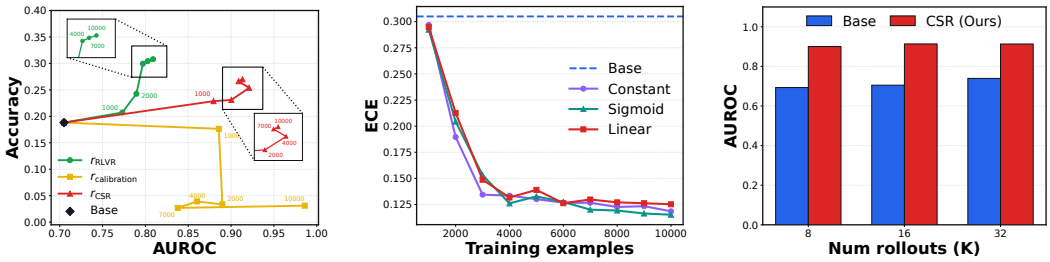
F1 preserves CSR's calibration gains. Replacing the pairwise judge with F1 barely changes CSR's calibration. Its average ECE stays within 0.01 of CSR-LLM on every backbone, and on Qwen it even drops from 0.1108 to 0.1021. CSR-F1 also has higher AUROC than Base-F1 and RLVR-F1 on every dataset and backbone. On Llama its average AUROC of 0.8116 exceeds the 0.7474 of the LLM-judged Base. CSR's calibration gain therefore does not require a pairwise LLM judge.

Relative to CSR-LLM, CSR-F1's average AUROC is lower by 0.023 on Llama to 0.062 on Mistral, and the loss is concentrated where answers vary in wording. On HotpotQA, whose answers are short entities, CSR-F1 is within 0.019 of CSR-LLM on every backbone, and on Llama it is even higher on NQ-Open. On Qwen and Mistral the drop is largest on MSMARCO, where CSR-F1 loses 0.149 and 0.131 AUROC. Its answers are free-form sentences whose paraphrases share few tokens, so F1 likely splits equivalent answers into separate clusters. Results for Qwen and Mistral and the judge-cost accounting are provided in Appendix D.9.

5.6 Ablation studies

All ablations are run on HotpotQA, with Llama-3.1-8B-Instruct in Figs. 5(a) and 5(c) and Qwen2.5-7B-Instruct in Fig. 5(b). Ablation settings are provided in Appendix B.1, and analyses of larger backbones, context-grounded training, and smaller budgets are provided in Appendix D.

Reward component ablation. To test whether both rewards are needed, we train with RLVR alone, the calibration reward alone, and CSR, and track accuracy and AUROC over training (Fig. 5(a)). RLVR raises accuracy from 0.19 to 0.31, but its AUROC reaches only 0.81. RLVR is indifferent to whether incorrect rollouts agree, so a wrong answer produced consistently looks as certain as a right one. The calibration reward alone fails in the opposite way. Its accuracy falls to about 0.03 within 2k examples, the low-accuracy regime of Proposition 1, and its AUROC then swings between 0.84 and



(a) Reward component ablation (b) Schedule comparison (c) Evaluation rollout budget

Figure 5: Ablations on HotpotQA. (a) Training trajectories of RLVR, the calibration reward alone, and CSR in the AUROC-accuracy plane on Llama, starting from Base, with point labels giving the number of training examples. (b) ECE over training for three $\lambda(t)$ schedules on Qwen, with Base as the dashed line. (c) AUROC of Base and CSR against the evaluation budget K on Llama.

0.99 because it is computed on very few correct questions. CSR's AUROC stays above RLVR's at every checkpoint. At 10k examples it reaches 0.91 against 0.81 for RLVR, while its accuracy rises to 0.27, about two-thirds of RLVR's gain. In CSR, the correctness term selects the answer and the calibration term shapes agreement around it.

Schedule comparison. To test whether the gain depends on how $\lambda(t)$ is ramped, we compare the sigmoid schedule of the main experiments with a constant weight and a linear schedule (Fig. 5(b)). All three cut ECE from about 0.30 to 0.13 within the first 4k examples and then improve slowly. The constant weight, which is largest early in training, lowers ECE fastest and reaches 0.19 at 2k examples, while sigmoid and linear reach 0.20 and 0.21. The sigmoid schedule ends lowest at 0.1152, against 0.118 for constant and 0.125 for linear. The final values differ by about 0.01 while all three stay far below Base and RLVR, so the gain comes from the calibration reward itself, and the schedule mainly trades early speed against the final value.

Evaluation rollout budget. To test how discrimination depends on the number of rollouts sampled at test time, we vary the evaluation budget $K \in \{8, 16, 32\}$ (Fig. 5(c)). Base's AUROC rises with K, from 0.69 at $K = 8$ to 0.74 at $K = 32$, while CSR's stays at 0.90–0.91. The AUROC gap between CSR and Base narrows from 0.21 at $K = 8$ to 0.17 at $K = 32$, and CSR with $K = 8$ still outperforms Base with $K = 32$. A likely reason is that training concentrates CSR's correct rollouts in one cluster, which a few samples already identify, whereas Base spreads its mass over more meanings and needs more samples for a stable entropy estimate. CSR's AUROC still rises slightly from $K = 8$ to $K = 16$, as a larger budget gives a more accurate estimate.

6 Conclusion

We propose Calibration with Semantic Reward (CSR), a framework that calibrates the confidence derived from a language model's semantic answer distribution without any explicit confidence interface. CSR is a training-time policy intervention rather than a confidence-elicitation method or a post-hoc calibration layer: coupling a verifiable correctness reward with a semantic calibration reward shapes the answer distribution so that semantic agreement across rollouts becomes an informative proxy for correctness under a fixed post-hoc estimator. Across three model families and four open-ended QA datasets, CSR achieves the lowest four-dataset average ECE and the highest average AUROC on every backbone, while trading at most 0.02 of four-dataset average accuracy against training on correctness alone. The gains also hold under four other sampling-based estimators and in selective prediction.

Limitations and future work. The semantic equivalence judge requires additional LLM evaluations among rollouts, introducing computational overhead. Although token-level F1 thresholding provides an efficient alternative, a performance gap remains compared with the LLM judge. Future work could further address this by designing more fine-grained, rule-based consistency metrics for rollout comparison. Our experiments also cover only short-answer QA, and long-form generation or code would need an equivalence judge for longer outputs. Despite these limitations, CSR represents a promising direction toward reliable uncertainty estimation without explicit confidence interfaces.

Acknowledgments and Disclosure of Funding

This work is supported in part by the U.S. Army Research Office under Army-ECASE award W911NF-07-R-0003-03, the U.S. Department of Energy, Office of Science, ARPA-H-SOL-24-101 program, IARPA HAYSTAC Program, DARPA YFA, NSF Grants #2205093, #2146343, #2134274, #2441832. This work is partially supported by the NSF award CCF-2112665 (TILOS), also supported in part by the CDC-RFA-FT-23-0069 from the CDC's Center for Forecasting and Outbreak Analytics. The authors declare no competing interests.

References

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[2] David Bani-Harouni, Chantal Pellegrini, Paul Stangel, Ege Özsoy, Kamilia Zaripova, Nassir Navab, and Matthias Keicher. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. In *International Conference on Learning Representations*, 2026.

[3] Debarun Bhattacharjya, Balaji Ganesan, Junkyu Lee, Radu Marinescu, Katya Mirylenka, Michael Glass, and Xiao Shou. SIMBA UQ: Similarity-based aggregation for uncertainty quantification in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15880–15894, 2025.

[4] Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training LMs to reason about their uncertainty. In *International Conference on Learning Representations*, 2026.

[5] Bryan Eikema, Evgenia Ilia, José G. C. de Souza, Chrysoula Zerva, and Wilker Aziz. Teaching language models to faithfully express their uncertainty. *arXiv preprint arXiv:2510.12587*, 2025.

[6] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[7] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[8] Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, 2024.

[9] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[10] Tobias Groot and Matias Valdenegro-Toro. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 145–171, 2024.

[11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017.

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[13] Daniel Han, Michael Han, and Unsloth team. Unsloth, 2023. URL `https://github.com/unslothai/unsloth`.

[14] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020.

[15] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

[16] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.

[17] Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. Large language models must be taught to know what they don't know. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[18] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[19] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[20] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tülu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling*, 2025.

[21] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, 2019.

[22] Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. ConfTuner: Training large language models to express their confidence verbally. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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

[24] Gabrielle Kaili-May Liu, Avi Caciularu, Gal Yona, Idan Szpektor, and Arman Cohan. Reinforcement learning with metacognitive feedback elicits faithful uncertainty expression in LLMs. *arXiv preprint arXiv:2606.32032*, 2026.

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

[26] Wenbo Pan, Jie Xu, Qiguang Chen, Junhao Dong, Libo Qin, Xinfeng Li, Haining Yu, and Xiaohua Jia. Can LLMs refuse questions they do not know? Measuring knowledge-aware refusal in factual tasks. In *International Conference on Learning Representations*, 2026.

[27] Xin Qiu and Risto Miikkulainen. Semantic Density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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

[29] Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. LACIE: Listener-aware finetuning for confidence calibration in large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[30] Jeremias Traub, Till J. Bungert, Carsten T. Lüth, Michael Baumgartner, Klaus H. Maier-Hein, Lena Maier-Hein, and Paul F. Jaeger. Overcoming common flaws in the evaluation of selective classification systems. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[31] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[32] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL `https://github.com/huggingface/trl`.

[33] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

[34] Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *International Conference on Learning Representations*, 2026.

[35] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations*, 2024.

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

[37] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[38] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[39] Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for LLMs. *arXiv preprint arXiv:2412.14737*, 2024.

[40] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018.

[41] Caiqi Zhang, Xiaochen Zhu, Chengzu Li, Nigel Collier, and Andreas Vlachos. LoVeC: Reinforcement learning for better verbalized confidence in long-form generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33336–33363, 2026.

A Comparison of training-time calibration methods

Training-time calibration methods differ in what the training signal is computed from and in what the deployed model must emit for a confidence score to exist, as summarized in Table 5.

Table 5: Training-time methods for calibrated confidence, by the source of the training signal and the output the deployed model must emit for a confidence score to exist.

Method	**Training signal computed from**	**Emitted at inference**
SaySelf [36]	Sampled-answer consistency (SFT targets), then stated confidence (RL)	Rationale + confidence
RD [2]	Stated confidence (log scoring rule)	Confidence token
RLCR [4]	Stated confidence (Brier term)	Confidence token
ConfTuner [22]	Confidence-token distribution	Confidence token
LACIE [29]	Listener's inferred confidence	Listener-judged utterance
LoVeC [41]	Stated confidence per statement	Per-statement confidence
Kapoor et al. [17]	Correctness labels of the model's own answers	Confidence head or token
FUT [5]	Sampled answers	Verbal hedge
RLMF [24]	Agreement of stated confidence with sampled-answer consistency	Numeric confidence $\rightarrow$ verbal hedge
CSR (Ours)	**Semantic agreement of sampled rollouts, rewarded to match correctness**	**Answer only (K samples)**

The table shows two differences from CSR. First, every prior method requires the deployed model to produce something beyond the answer, such as a verbalized confidence, a hedged utterance, or the output of a trained confidence head, so its confidence score depends on that additional channel. Second, the methods that share our source signal, SaySelf's supervised stage, FUT, and RLMF, still use that signal as a target to be reported rather than as a reward that reshapes the distribution the signal is computed from. RLMF, the closest of these, rewards a stated confidence for matching the consistency of sampled answers, whereas CSR scores the rollouts themselves: it rewards the agreement structure, which reshapes the answer distribution, so its gains persist under estimators not used in training (Appendix D.1).

B Experimental details

B.1 Training and evaluation settings

Hardware and software. All experiments run on a server with Ubuntu 22.04.4 LTS and NVIDIA A100 80GB PCIe GPUs, using CUDA 12.8, Python 3.11.14, and PyTorch 2.9.0. Each individual fine-tuning or evaluation run uses a single A100 80GB GPU.

Fine-tuning setup (CSR and RLVR). We implement CSR and RLVR with Unsloth [13], TRL [32], and vLLM [19] and use one recipe across the three model families of Table 1. For training, we shuffle HotpotQA [40] with seed 42 and use the first 10,000 examples. Models are loaded in 4-bit precision and fine-tuned with LoRA adapters using rank 32 and alpha 32. We sample $K = 8$ rollouts per example with temperature 0.7 and top-p 0.95, as at evaluation, use one prompt group per microbatch with gradient accumulation of 4 (effective batch size of 4 prompts and 32 rollouts per optimizer step), and train for one epoch of $2{,}500$ optimizer steps. We set the KL coefficient $\beta = 0.01$, the agreement clipping constant of Eq. (5) to $\epsilon = 10^{-6}$, the maximum prompt length to 256 tokens, and the maximum completion length to 768 tokens, and use a learning rate of 5×10^{-6}, an 8-bit AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.99$), weight decay 0.1, a warmup ratio of 0.1, and a cosine learning-rate schedule.

Fine-tuning setup (RD and RLCR). RD and RLCR follow the original settings and released code of Bani-Harouni et al. [2] and Damani et al. [4], and we list only the settings that differ. Both are

trained on the same $10{,}000$ HotpotQA examples and backbones as CSR, with the same rank-32 LoRA adapters. RD is trained for one epoch of $1{,}250$ PPO steps with 8 prompts per step. RLCR uses the same training rollout count ($K = 8$) and batch size (4 prompts and 32 rollouts per optimizer step) as CSR. It keeps its original context-grounded HotpotQA prompt with stochastic paragraph removal. The original release evaluates HotpotQA with these paragraphs and the other datasets without them, whereas we evaluate every dataset closed-book, as for all methods. We do not retrain RLCR closed-book because the prompt and context format are part of the method.

CSR reward settings. For CSR, the calibration reward enters in its normalized form (Sec. 4.2), which does not depend on ϵ. We judge each unordered rollout pair once with the semantic-equivalence judge prompt (Appendix B.4) and share its verdict between both endpoints, giving $\binom{8}{2} = 28$ pairwise judgments per group. Every main-results run uses the sigmoid calibration-weight schedule (iii) defined in the ablation setup below, with $\lambda_{\min} = 0.8$, $\lambda_{\max} = 1.2$, and $\kappa = 10$ on every backbone except Qwen3-14B, which uses $\lambda_{\min} = 0.5$ and $\lambda_{\max} = 0.7$ (Appendix D.6), and the other schedules are used only in the schedule comparison of Fig. 5(b).

Evaluation setup. For evaluation on the four datasets (HotpotQA, TriviaQA, MSMARCO, NQ-Open), we use the validation split of each dataset, which does not intersect with the HotpotQA training data used for fine-tuning. We shuffle each split with seed 42 and use the first 1,000 examples. Base, RLVR, and CSR sample $K = 16$ rollouts per question with temperature 0.7 and top-p 0.95, while RD and RLCR use greedy decoding. Generation is capped at 256 new tokens for Base, RLVR, and CSR, and for RD, whose answer and confidence come in one short output, and at 1536 for RLCR, whose structured output includes a reasoning segment.

Motivation experiment setup. The motivation experiments use the RLCR fine-tuned Llama-3.1-8B-Instruct model described above. For Fig. 1(a), we greedy-decode the response to a single HotpotQA question and plot the probabilities the model assigns to each confidence value at the token position where the confidence is emitted. For Figs. 1(b) and 1(c), we sample 8 answers per question with temperature 0.7 and top-p 0.95. The example in Fig. 1(b) shows three out of eight rollouts that all produce the same correct answer, "Take It Easy", but report different confidence values. For Fig. 1(c), we measure confidence variance within semantically equivalent groups over $1{,}000$ HotpotQA examples. For each question, we use the LLM judge to cluster the 8 sampled answers into different semantic groups. For each group with at least two semantically equivalent samples, we compute the standard deviation of the reported verbalized confidence values within that group. We then aggregate these group-level standard deviations by group size and plot their distributions, measuring whether verbalized confidence remains stable among rollouts that express the same answer.

Ablation setup. We keep the fine-tuning details of the main results and run all ablations on HotpotQA. Fig. 5(a) uses Llama-3.1-8B-Instruct and shows the effect of each reward component using checkpoints of each method's 10,000-example run after $N \in \{1{,}000,\ 2{,}000,\ 4{,}000,\ 7{,}000,\ 10{,}000\}$ examples, each evaluated with $K = 16$ rollouts per question as in Table 1.

Fig. 5(b) uses Qwen2.5-7B-Instruct and compares the sigmoid schedule used in the main experiments against alternative schedules over training step t and total training steps T, with $\lambda_{\min} = 0.8$ and $\lambda_{\max} = 1.2$. We consider (i) a constant weight $\lambda(t) = 1.0$, (ii) linear growth:

$$\lambda(t) = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})\frac{t}{T},$$

and (iii) sigmoid-shaped growth:

$$\lambda(t) = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})\,\frac{\sigma\big(\kappa(t/T - 0.5)\big) - \sigma(-\kappa/2)}{\sigma(\kappa/2) - \sigma(-\kappa/2)},$$

where $\sigma(\cdot)$ denotes the logistic function, $\kappa = 10$ controls the slope, and $T = 2{,}500$ optimizer steps (10k examples at 4 prompts per step). The normalization makes the sigmoid schedule, like the linear one, start exactly at $\lambda_{\min}$ and end exactly at $\lambda_{\max}$.

Fig. 5(c) evaluates the Llama-3.1-8B-Instruct main-results checkpoint on the HotpotQA evaluation set of Table 1 with $K \in \{8, 16, 32\}$ rollouts per question. Each budget is a separate evaluation with K rollouts per question.

B.2 Formal definitions

We use standard terminology [11, 8, 23]. *Uncertainty* is a functional of the predictive distribution $\pi_\theta(\cdot \mid x)$ [23], here the semantic entropy of $\pi_\theta(\cdot \mid x)$, which $\mathrm{SE}(x)$ estimates from K rollouts, and is thus a property of the question alone. *Confidence* estimates the probability that a *particular* answer is correct, so it depends on both the input and the evaluated output. Write $\mathbf{Y} = (Y^{(1)}, \ldots, Y^{(K)}) \sim \pi_\theta(\cdot \mid x)^K$ for a rollout group and $\mathbf{Y}^{(-j)}$ for the siblings of $Y^{(j)}$. A response-level estimate $\psi_\phi(x, Y^{(j)}; \mathbf{Y}^{(-j)}) \in [0, 1]$ (the subscript ϕ denotes the estimator and any fitted parameters) may also read those siblings. It may be *verbalized* by the model or *derived* post hoc from samples, as ours is. Aggregating the response-level estimates, or reading the group directly as semantic entropy does, gives a *question-level UQ score* $c(x) \in [0, 1]$ (Appendix B.3). Because it is oriented so that larger values indicate greater expected correctness, it is a question-level *confidence score*, and its finite-sample value depends on the sampled group $\mathbf{Y}$ as well as on x. We keep SE uncertainty-oriented (larger means less certain) and use $\hat{s}$ as $c(x)$ for Base, RLVR, and CSR.

Let X be a random evaluation question, let $Q = J(Y, Y^*) \in \{0, 1\}$ be the judged correctness of a fresh rollout $Y \sim \pi_\theta(\cdot \mid X)$ against the gold answer Y^*, and let $\eta(x) = \mathbb{E}_{Y \sim \pi_\theta(\cdot|x)}[Q]$ be the policy's correctness probability for the question. A *fixed* post-hoc score is *calibrated* when

$$\mathbb{E}[\eta(X) \,|\, c(X) = v] = v \qquad \text{for every } v \in [0, 1], \tag{7}$$

and ECE estimates the deviation from this condition, with $\eta(x)$ replaced by the accuracy $\mathrm{Acc}(x)$ of the scored answers (the K rollouts, or the single greedy answer for RD and RLCR). *Discrimination*, ranking correct questions above incorrect ones, is measured by AUROC.

B.3 Evaluation metrics

We evaluate model performance using accuracy, expected calibration error, the area under the receiver operating characteristic curve, two selective-prediction summaries (AURAC and AUGRC), and token cost. All metrics are computed from question-level quantities: for Base, RLVR, and CSR we aggregate the K sampled rollouts of each question, and for RD and RLCR we use the single greedy answer.

Accuracy (Acc). For each generated answer, we use GPT-4.1 nano as an LLM judge to determine whether the generated answer is semantically correct with respect to the gold answer(s). The judge prompts are given in Appendix B.4. We define question-level accuracy as the average judged correctness Q_j (Sec. 4) of the K sampled rollouts:

$$\mathrm{Acc}(x) = \frac{1}{K} \sum_{j=1}^{K} Q_j. \tag{8}$$

We then report overall accuracy by averaging this quantity over the evaluation questions. For RD and RLCR, which decode greedily, $K = 1$ and $\mathrm{Acc}(x)$ is the $0/1$ correctness of the single answer.

Confidence score. We write $c(x) \in [0, 1]$ for the confidence score of question x: the semantic confidence $\hat{s}(x)$ of Sec. 3 for Base, RLVR, and CSR, computed from the discrete semantic entropy of Eq. (3) [6] over semantic-equivalence classes formed from one LLM-judge verdict per unordered rollout pair, and the verbalized confidence for RD and RLCR.

Expected calibration error (ECE). With $c(x)$ and $\mathrm{Acc}(x)$ as above, we partition the n evaluation questions into $B = 10$ equal-width bins according to their confidence values, following the binning procedure of Guo et al. [11]. Let $\mathcal{B}_b$ denote the set of questions whose confidence falls into bin b, with empty bins contributing zero. The ECE is computed as

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|\mathcal{B}_b|}{n} \left| \frac{1}{|\mathcal{B}_b|} \sum_{x \in \mathcal{B}_b} \mathrm{Acc}(x) - \frac{1}{|\mathcal{B}_b|} \sum_{x \in \mathcal{B}_b} c(x) \right|. \tag{9}$$

Lower ECE indicates better agreement between confidence and observed accuracy.

AUROC. AUROC evaluates how well confidence distinguishes correct from incorrect predictions. Since our question-level accuracy $\mathrm{Acc}(x)$ is a continuous value in $[0, 1]$ rather than a binary label, we binarize it using a threshold of 0.5:

$$\tilde{a}(x) = \mathbf{1}[\mathrm{Acc}(x) \geq 0.5]\,. \tag{10}$$

We then compute AUROC by treating $c(x)$ as the ranking score and $\tilde{a}(x) \in \{0, 1\}$ as the binary correctness label. Formally, AUROC is the probability that a randomly chosen positive example receives a higher confidence score than a randomly chosen negative example, with ties counted as one half:

$$\text{AUROC} = \Pr\left(c(x^+) > c(x^-)\right) + \tfrac{1}{2}\Pr\left(c(x^+) = c(x^-)\right), \tag{11}$$

where x^+ and x^- denote questions with $\tilde{a}(x^+) = 1$ and $\tilde{a}(x^-) = 0$, respectively. A higher AUROC indicates that the confidence score more effectively ranks correct questions above incorrect ones.

Selective prediction. Two risk-coverage summaries are computed from the same confidence score $c(x)$ (Sec. 5.4, Appendix D.3). Ordering the n evaluation questions by decreasing $c(x)$, the retained set $\mathcal{R}(\rho)$ at coverage $\rho \in (0, 1]$ consists of the $\lceil \rho n \rceil$ most confident questions. Because $c(x)$ has many ties (RD, for example, emits only a few distinct values), a coverage level can fall inside a group of tied questions whose order the score does not determine, so we evaluate both integrals below in exact expectation over uniformly random orderings within each tied group.

AURAC. The area under the rejection-accuracy curve summarizes the fraction of retained questions with $\tilde{a}(x) = 1$ (at least half of the scored answers correct) as increasingly uncertain questions are rejected [6]. It is defined as

$$\text{AURAC} = \int_0^1 \frac{1}{|\mathcal{R}(\rho)|} \sum_{x \in \mathcal{R}(\rho)} \tilde{a}(x)\, d\rho, \tag{12}$$

where higher values are better.

AUGRC. The area under the generalized risk-coverage curve instead integrates the joint probability that a question is both accepted and answered incorrectly [30]. It is defined as

$$\text{AUGRC} = \int_0^1 \frac{1}{n} \sum_{x \in \mathcal{R}(\rho)} \left(1 - \tilde{a}(x)\right) d\rho. \tag{13}$$

Lower values indicate better selective performance. We report AUGRC on $[0, 1]$, without the $\times 1000$ display scaling used by Traub et al. [30].

Token cost (Tok). We also report the token cost per question, defined as the total number of prompt tokens and generated output tokens:

$$\text{Tok}(x) = T_{\text{prompt}}(x) + T_{\text{output}}(x), \tag{14}$$

where $T_{\text{prompt}}(x)$ is the number of prompt tokens for question x, and $T_{\text{output}}(x)$ is the number of generated output tokens, summed over all K rollouts ($K = 1$ for RD and RLCR), with the prompt counted once. We report Tok by averaging this quantity over evaluation questions.

B.4 Prompts

Base and RLVR use the same prompt as CSR.

CSR prompt

Provide a concise and direct answer to the question. Avoid unnecessary explanations or additional text.

RLCR prompt

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

<analysis></analysis> tags. The assistant should reason about its confidence in the solution and its uncertainty in the solution within these tags.
Here are some guidelines for the analysis:
1. Your task is to point out things where the model could be wrong in its thinking, or things where there might be ambiguity in the solution steps, or in the reasoning process itself.
2. You should not suggest ways of fixing the response; your job is only to reason about uncertainties.
3. For some questions, the response might be correct. In these cases, it is also okay to have only a small number of uncertainties and then explicitly say that you are 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>

Rewarding Doubt (RD) prompt

You will get questions. Answer with the correct answer. Additionally provide a confidence between 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, of how sure you are the answer is correct. A value close to 0 means you think there is a high probability that the answer is wrong. The closer the value is to 10, the higher you think is the probability that the answer is correct. The output should have the format Answer: <answer>, Confidence: <confidence> and nothing else.

Judge prompt (semantic equivalence)

Given a question and two potential answers, you need to analyze whether the two answers convey the same meaning. Respond with "yes" or "no".
Your response should be completely based on the similarity between the two answers.
DO NOT use your own knowledge base when do the comparison.

Question:
{q}
Response 1:
{a}
Response 2:
{b}

Your response should use the following format:
Correct: <yes or no>

Judge prompts. Every call to the LLM judge uses GPT-4.1 nano with the system message "You are a helpful assistant that helps people find information." The semantic-equivalence prompt above produces the pairwise agreement s_{ji} and, with the gold answer in place of *Response 2*, the correctness label Q_j on every dataset.

C Analysis of the calibration reward

This appendix proves Proposition 1 and contrasts the calibration reward with an explicit regularizer.

C.1 Proof of Proposition 1

Fix a question x and a rollout group $Y^{(1)}, \dots, Y^{(K)}$ drawn i.i.d. from $\pi_\theta(\cdot \mid x)$, and recall from Sec. 4 and Proposition 1

$$s_{ji} = J\big(Y^{(j)}, Y^{(i)}\big), \qquad Q_j = J\big(Y^{(j)}, Y^{*}\big), \qquad \alpha = \frac{1}{K}\sum_{j=1}^{K} Q_j, \qquad c_\epsilon = \log\frac{1-\epsilon}{\epsilon},$$

$$\hat{p}_j = \frac{1}{K-1}\sum_{i \neq j} s_{ji}, \qquad p_j = \mathbb{P}_{Y' \sim \pi_\theta(\cdot|x)}\big(J(Y^{(j)}, Y') = 1 \mid Y^{(j)}\big)$$

that is, the pairwise agreement, the correctness of rollout j, the empirical accuracy of the group, the cross-entropy scale, the empirical agreement rate of rollout j, and its policy-level counterpart.

Step 1 (one pair). Because $s_{ji}, Q_j \in \{0, 1\}$, the ϵ-softened summand of Eq. (5) takes only two values:

$$-\mathrm{CE}(s_{ji}, Q_j) = \begin{cases} \log(1-\epsilon), & s_{ji} = Q_j, \\ \log\epsilon, & s_{ji} \neq Q_j. \end{cases}$$

Hence the summand is affine in the indicator of agreement with correctness:

$$-\mathrm{CE}(s_{ji}, Q_j) = \log\epsilon + c_\epsilon\, \mathbf{1}[s_{ji} = Q_j].$$

Step 2 (one rollout). Each unordered pair is judged once and its verdict is shared by both rollouts, so $s_{ji} = s_{ij}$. We assume accordingly that J is symmetric, $J(y, y') = J(y', y)$, so that $s_{ji} = J\big(Y^{(j)}, Y^{(i)}\big)$ holds for both orderings of the pair and p_j above is defined with this symmetric judge. For a fixed j, the fraction of siblings whose agreement with $Y^{(j)}$ matches Q_j is

$$\frac{1}{K-1}\sum_{i \neq j} \mathbf{1}[s_{ji} = Q_j] = Q_j\hat{p}_j + (1 - Q_j)(1 - \hat{p}_j).$$

Averaging Step 1 over $i \neq j$ therefore gives

$$r^{(j)}_{\text{calibration}} = \log\epsilon + c_\epsilon\big[Q_j\hat{p}_j + (1 - Q_j)(1 - \hat{p}_j)\big].$$

Conditional on $Y^{(j)}$, the other rollouts are i.i.d. draws from $\pi_\theta(\cdot \mid x)$, so each s_{ji} with $i \neq j$ has conditional mean p_j, and

$$\mathbb{E}\big[\hat{p}_j \mid Y^{(j)}\big] = p_j.$$

Step 3 (the group). Split the rollouts by correctness into $\mathcal{C} = \{j : Q_j = 1\}$ and $\mathcal{I} = \{j : Q_j = 0\}$, so that $|\mathcal{C}| = K\alpha$ and $|\mathcal{I}| = K(1 - \alpha)$, and define the mean agreement rates

$$\bar{p}_{\mathcal{C}} = \frac{1}{|\mathcal{C}|}\sum_{j \in \mathcal{C}} \hat{p}_j,$$

$$\bar{p}_{\mathcal{I}} = \frac{1}{|\mathcal{I}|}\sum_{j \in \mathcal{I}} \hat{p}_j.$$

Averaging Step 2 over j and splitting the sum accordingly gives

$$\bar{r}_{\text{calibration}} = \log\epsilon + c_\epsilon\Big[\alpha\,\bar{p}_{\mathcal{C}} + (1 - \alpha)\big(1 - \bar{p}_{\mathcal{I}}\big)\Big],$$

where a term whose set is empty is omitted.

Consequences. When $\alpha \to 0$, the second term dominates and is maximized by $p_j \to 0$. When $\alpha \to 1$, the first term dominates and is maximized by $p_j \to 1$. Neither regime directly rewards a change from an incorrect to a correct semantic mode. □

Remark (failure of the calibration reward alone). Consider a group in which rollout 1 is correct and agrees with no sibling, and the other $K - 1$ rollouts are incorrect and mutually disagreeing. Then $\hat{p}_j = 0$ for every j, so the normalized reward is

$$Q_j\hat{p}_j + (1 - Q_j)(1 - \hat{p}_j) = \begin{cases} 0, & j = 1 \text{ (the correct rollout)}, \\ 1, & j \neq 1 \text{ (every incorrect rollout)}. \end{cases}$$

Rollout 1 thus has the lowest reward in its group and a negative group-relative advantage: the calibration reward alone pushes probability away from the only correct answer.

Coupling the calibration reward with r_{RLVR} (Eq. (4)) adds a unit reward to correct rollouts only. In this group the correct rollout then receives 1 and each incorrect rollout $\lambda(t)$, so it outranks them whenever $\lambda(t) < 1$. The curriculum therefore starts with $\lambda(t)$ below 1, when accuracy is low and such isolated correct rollouts are common.

C.2 Calibration reward versus explicit regularization

The semantic calibration reward is an *auxiliary reward with a regularizing effect* on the policy's semantic answer distribution, but it is not, in general, equivalent to adding an independent regularization term to the loss.

Reward-space formulation. For one prompt x, let $Y^{(1)}, \ldots, Y^{(K)} \sim \pi_\theta(\cdot \mid x)$ be its rollout group. For brevity write $r_{\mathrm{R}}^{(j)} = r_{\mathrm{RLVR}}^{(j)}$ and let $r_{\mathrm{C}}^{(j)}$ be the normalized calibration reward used in training (Sec. 4.2 and Eq. (17) below), and define their group-centered versions together with the combined group scale by

$$\begin{aligned} \widetilde{r}_u^{(j)} &= r_u^{(j)} - \frac{1}{K}\sum_{\ell=1}^{K} r_u^{(\ell)}, \qquad u \in \{\mathrm{R}, \mathrm{C}\}, \\ \sigma_{\mathrm{CSR}} &= \left[\frac{1}{K}\sum_{\ell=1}^{K}\left(\widetilde{r}_{\mathrm{R}}^{(\ell)} + \lambda(t)\widetilde{r}_{\mathrm{C}}^{(\ell)}\right)^2\right]^{1/2} + \delta. \end{aligned} \tag{15}$$

CSR combines the two rewards *before* group normalization, yielding

$$A_{\mathrm{CSR}}^{(j)} = \frac{\widetilde{r}_{\mathrm{R}}^{(j)} + \lambda(t)\widetilde{r}_{\mathrm{C}}^{(j)}}{\sigma_{\mathrm{CSR}}}. \tag{16}$$

Here $\delta > 0$ is the usual group-normalization stabilizer, distinct from the cross-entropy softening constant ϵ in Eq. (5). Equation (16) shows that the calibration term changes rollout-level credit assignment, rather than adding a batch-level penalty after the RLVR advantage is fixed.

Effect within a rollout group. Under pure RLVR, two rollouts with the same correctness receive the same reward. To expose the additional ordering induced by CSR, recall that, with Q_j and $\hat{p}_j$ as in Proposition 1, this normalized reward is

$$r_{\mathrm{C}}^{(j)} = Q_j\hat{p}_j + (1 - Q_j)(1 - \hat{p}_j). \tag{17}$$

For two rollouts j and j' with equal correctness, the correctness-reward terms cancel exactly. Substituting Eq. (17) into Eq. (16) gives

$$\begin{aligned} A_{\mathrm{CSR}}^{(j)} - A_{\mathrm{CSR}}^{(j')} &= \frac{\lambda(t)}{\sigma_{\mathrm{CSR}}}\left(r_{\mathrm{C}}^{(j)} - r_{\mathrm{C}}^{(j')}\right) \\ &= \frac{\lambda(t)}{\sigma_{\mathrm{CSR}}}\begin{cases} \hat{p}_j - \hat{p}_{j'}, & Q_j = Q_{j'} = 1, \\ \hat{p}_{j'} - \hat{p}_j, & Q_j = Q_{j'} = 0. \end{cases} \end{aligned} \tag{18}$$

Thus, among correct rollouts, greater semantic agreement receives greater advantage. Among incorrect rollouts, greater agreement receives *lower* advantage because $1 - \hat{p}$ decreases with $\hat{p}$. CSR thus concentrates credit on consistent correct answers while suppressing concentrated incorrect modes, which is the sense in which the auxiliary reward has a regularizing effect.

Comparison with an explicit regularizer. A separate regularization construction would first normalize the correctness reward alone and then add an independent calibration term:

$$\begin{aligned} A_{\mathrm{R}}^{(j)} &= \frac{\widetilde{r}_{\mathrm{R}}^{(j)}}{\sigma_{\mathrm{R}}}, \qquad \sigma_{\mathrm{R}} = \left[\frac{1}{K}\sum_{\ell=1}^{K}\left(\widetilde{r}_{\mathrm{R}}^{(\ell)}\right)^2\right]^{1/2} + \delta, \\ \mathcal{L}_{\mathrm{sep}} &= \mathcal{L}_{\mathrm{GRPO}}(A_{\mathrm{R}}) + \gamma(t)\,\Omega_{\mathrm{C}} + \beta\,\mathcal{L}_{\mathrm{KL}}. \end{aligned} \tag{19}$$

Here $\gamma(t) \geq 0$ weights a calibration term Ω_{C}. Because $r_{\mathrm{C}}^{(j)}$ is computed from discrete rollouts and an external judge, Ω_{C} must use a score-function surrogate to transmit a policy gradient, $\Omega_{\mathrm{C}} =$

$-\frac{1}{K}\sum_j \mathrm{sg}\big(\widetilde{r}_{\mathrm{C}}^{(j)}\big)\log \pi_\theta\big(Y^{(j)} \mid x\big)$ with stop-gradient sg, so that $-\nabla_\theta \Omega_{\mathrm{C}} = g_{\mathrm{C}}$, with g_{C} defined in Eq. (20) below. To compare the two constructions, assume that this surrogate uses the same samples and the same centered calibration rewards as CSR, and define the per-group score-function components

$$g_u = \frac{1}{K}\sum_{j=1}^{K} \widetilde{r}_u^{(j)} \nabla_\theta \log \pi_\theta\Big(Y^{(j)} \mid x\Big), \qquad u \in \{\mathrm{R}, \mathrm{C}\}. \tag{20}$$

Ignoring importance-ratio clipping and treating the sampled rewards, judge outputs, and normalization statistics as stop-gradient quantities, the relevant per-group policy-gradient directions, up to a shared positive constant, are

$$\widehat{g}_{\mathrm{CSR}} = \frac{g_{\mathrm{R}} + \lambda(t) g_{\mathrm{C}}}{\sigma_{\mathrm{CSR}}}, \qquad \widehat{g}_{\mathrm{sep}} = \frac{g_{\mathrm{R}}}{\sigma_{\mathrm{R}}} + \gamma(t) g_{\mathrm{C}}. \tag{21}$$

For a single rollout group with nonzero variance, and generically non-collinear g_{R} and g_{C}, these directions are proportional only if

$$\gamma(t) = \frac{\lambda(t)}{\sigma_{\mathrm{R}}}, \qquad \text{in which case} \qquad \widehat{g}_{\mathrm{sep}} = \frac{\sigma_{\mathrm{CSR}}}{\sigma_{\mathrm{R}}} \widehat{g}_{\mathrm{CSR}}. \tag{22}$$

This condition is restrictive: σ_{R} varies across prompt groups, whereas $\gamma(t)$ is ordinarily a single global coefficient, and the factor $\sigma_{\mathrm{CSR}}/\sigma_{\mathrm{R}}$ also changes the relative weighting of groups when their updates are aggregated. Importance-ratio clipping introduces an additional mismatch. Consequently, a fixed additive regularizer does not generally reproduce CSR over a training batch.

Choice in CSR. CSR uses the reward-space construction in Eq. (16), so the semantic structure of grouped rollouts enters before group normalization and directly changes rollout-level credit assignment. The KL term $\beta\,\mathcal{L}_{\mathrm{KL}}$ remains the explicit policy regularizer.

D Supplementary empirical analyses

D.1 Transfer to other confidence estimators

This section defines the four post-hoc estimators of Sec. 5.3 and reports their per-dataset results on the rollouts of Table 1 (Tables 6 and 7). Each estimator is a response-level estimate $\psi_j = \psi_\phi(x, Y^{(j)}; \mathbf{Y}^{(-j)}) \in [0,1]$ in the sense of Appendix B.2, and the question-level confidence of Appendix B.3 is its average over the K rollouts, $c(x) = K^{-1}\sum_{j=1}^{K}\psi_j$. We write Jac_{ji} for the word-level Jaccard similarity between rollouts $Y^{(j)}$ and $Y^{(i)}$, $\bar{\pi}_j = \exp\big(|Y^{(j)}|^{-1}\log \pi_\theta(Y^{(j)} \mid x)\big)$ for the length-normalized sequence probability of $Y^{(j)}$, and $\mathcal{M}(Y)$ for the semantic-equivalence class containing Y (Sec. 3).

Semantic Density (SemDens) [27] scores a rollout by its probability-weighted density among the set $\mathcal{U}$ of distinct answers in the group (the rollout itself included, so $|\mathcal{U}| \le K$),

$$\psi_j = \frac{\sum_{i\in\mathcal{U}} \bar{\pi}_i\, \mathbf{1}\big[Y^{(i)} \in \mathcal{M}(Y^{(j)})\big]}{\sum_{i\in\mathcal{U}} \bar{\pi}_i}, \tag{23}$$

computed as in the official implementation, except that the references are the temperature-sampled rollouts and the kernel is the same-class indicator of our semantic-equivalence judge in place of the original NLI-based similarity. The outer average in $c(x)$ runs over all K rollouts, duplicates included, whereas the inner normalization runs over $\mathcal{U}$.

The other three are aggregations from SIMBA UQ [3]. The **Bayesian aggregation** (Bayes-beta) is the posterior probability that a rollout is correct given its similarities to the others, under Beta likelihoods f_1 and f_0 for correct and incorrect rollouts and a prior p_0,

$$\psi_j = \frac{p_0 \prod_{i\neq j} f_1(\mathrm{Jac}_{ji})}{p_0 \prod_{i\neq j} f_1(\mathrm{Jac}_{ji}) + (1-p_0)\prod_{i\neq j} f_0(\mathrm{Jac}_{ji})}. \tag{24}$$

The unsupervised **arithmetic aggregation** (Arithmetic) averages the similarities to the $K-1$ siblings, $\psi_j = (K-1)^{-1}\sum_{i\neq j}\mathrm{Jac}_{ji}$, so that $c(x)$ is the group's mean pairwise Jaccard similarity. The

Table 6: Three aggregations from SIMBA UQ [3] on the rollouts of Table 1: Bayesian aggregation (Bayes-beta), unsupervised arithmetic aggregation (Arithmetic), and the supervised Clf-pairs+gen classifier. **Bold** marks the best and underline the second-best value in each column within each model block.

Bayesian aggregation (Bayes-beta, supervised)

		In-distribution		Out-of-distribution							
		HotpotQA		TriviaQA		MSMARCO		NQ-Open		Average	
Model	**Method**	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑
Llama	Base	0.2034	0.7575	0.1492	0.8283	0.2391	0.5841	0.1881	0.7028	0.1950	0.7182
	RLVR	0.2084	0.8152	0.2066	0.7946	0.2502	0.6392	0.2343	0.6797	0.2249	0.7322
	CSR	**0.1440**	**0.8889**	**0.1278**	**0.8699**	**0.1882**	**0.6531**	**0.1741**	**0.7460**	**0.1585**	**0.7895**
Qwen	Base	0.2330	0.8233	0.4971	0.8152	0.2546	0.6256	0.2340	0.7595	0.3047	0.7559
	RLVR	0.2550	0.8190	0.4996	0.8163	0.2701	0.6351	0.2333	0.7406	0.3145	0.7528
	CSR	**0.1260**	**0.9191**	**0.1529**	**0.9079**	**0.2138**	**0.6759**	**0.1760**	**0.8501**	**0.1672**	**0.8383**
Mistral	Base	0.2199	0.7015	0.2318	0.7026	**0.1853**	**0.5943**	0.2487	0.6460	0.2214	0.6611
	RLVR	0.2158	0.7380	0.1929	0.7491	0.2215	0.5779	0.2373	**0.6638**	0.2169	0.6822
	CSR	**0.1576**	**0.7974**	**0.1892**	**0.7518**	0.2187	0.5781	**0.2288**	0.6266	**0.1986**	**0.6885**

Arithmetic aggregation (Arithmetic, unsupervised)

		In-distribution		Out-of-distribution							
		HotpotQA		TriviaQA		MSMARCO		NQ-Open		Average	
Model	**Method**	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑
Llama	Base	0.2525	0.7165	0.0810	0.8214	0.0692	0.6025	0.0458	0.7274	0.1121	0.7169
	RLVR	0.2725	0.8011	**0.0629**	0.7991	0.1545	0.6628	0.1166	0.6881	0.1516	0.7378
	CSR	**0.1323**	**0.8789**	0.0903	**0.8759**	**0.0444**	**0.6829**	**0.0355**	**0.7562**	**0.0756**	**0.7985**
Qwen	Base	0.3430	0.8208	0.2043	0.8182	0.2861	0.6330	0.3072	0.7530	0.2851	0.7562
	RLVR	0.3436	0.8201	0.2015	0.8178	0.2863	0.6476	0.2878	0.7476	0.2798	0.7583
	CSR	**0.1339**	**0.9227**	**0.0799**	**0.9111**	**0.1223**	**0.6890**	**0.0891**	**0.8483**	**0.1063**	**0.8428**
Mistral	Base	0.3059	0.7037	0.1058	0.6866	0.0546	0.6024	0.0856	**0.6691**	0.1380	0.6655
	RLVR	0.2873	0.7462	**0.0786**	0.7398	0.0706	0.5990	0.1067	0.6690	0.1358	0.6885
	CSR	**0.1891**	**0.7694**	0.1699	**0.7450**	**0.0342**	**0.6112**	**0.0358**	0.6414	**0.1072**	**0.6918**

Random-forest classifier (Clf-pairs+gen, supervised)

		In-distribution		Out-of-distribution							
		HotpotQA		TriviaQA		MSMARCO		NQ-Open		Average	
Model	**Method**	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑
Llama	Base	**0.0200**	0.8062	0.0302	0.8413	**0.0245**	0.6127	0.0401	0.7387	**0.0287**	0.7497
	RLVR	0.0808	0.7624	0.0694	0.7847	0.0464	0.6270	0.0505	0.6632	0.0618	0.7093
	CSR	0.0465	**0.8622**	**0.0205**	**0.8686**	0.0329	**0.6940**	**0.0248**	**0.7428**	0.0312	**0.7919**
Qwen	Base	0.1113	0.7735	0.1454	0.7657	0.0937	0.5763	0.1108	0.6795	0.1153	0.6987
	RLVR	0.1530	0.7115	0.1499	0.7507	0.1054	0.5763	0.1095	0.6639	0.1294	0.6756
	CSR	**0.0848**	**0.8688**	**0.0795**	**0.8833**	**0.0304**	**0.6941**	**0.0474**	**0.8179**	**0.0605**	**0.8160**
Mistral	Base	0.0390	0.6834	0.0574	0.6860	0.0225	0.5955	0.0364	0.6571	0.0388	0.6555
	RLVR	0.0690	0.6839	0.0726	0.7134	0.0155	0.5887	0.0305	**0.6702**	0.0469	0.6641
	CSR	**0.0357**	**0.7991**	**0.0459**	**0.7791**	**0.0120**	**0.6029**	**0.0173**	0.6696	**0.0277**	**0.7127**

supervised **Clf-pairs+gen** classifier is a random forest h over the similarities and the generative score $\bar{\pi}_j$,

$$\psi_j = h\big((\mathrm{Jac}_{ji})_{i \neq j},\, \bar{\pi}_j\big). \tag{25}$$

The five parameters of Bayes-beta and the random forest are fitted *per policy* with question-grouped five-fold cross-fitting, so every reported confidence is an out-of-fold score.

Table 7: Semantic Density (SemDens) [27] on the rollouts of Table 1. **Bold** marks the best and underline the second-best value in each column within each model block.

		In-distribution		Out-of-distribution							
		HotpotQA		TriviaQA		MSMARCO		NQ-Open		Average	
Model	**Method**	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑
Llama	Base	0.3418	0.6928	0.0656	0.8420	0.2542	0.7061	0.1378	0.7342	0.1998	0.7438
	RLVR	0.2832	0.8073	0.0902	0.8295	0.2707	0.7332	0.1919	0.6997	0.2090	0.7674
	CSR	**0.1213**	**0.9173**	**0.0509**	**0.9039**	**0.2140**	**0.7574**	**0.1143**	**0.7727**	**0.1251**	**0.8378**
Qwen	Base	0.2901	0.8407	0.1776	0.8288	0.3038	0.7030	0.2400	0.7765	0.2529	0.7872
	RLVR	0.2857	0.8360	0.1835	0.8252	0.2994	0.7067	0.2407	0.7601	0.2523	0.7820
	CSR	**0.1170**	**0.9309**	**0.0810**	**0.9096**	**0.1843**	**0.8009**	**0.0957**	**0.8348**	**0.1195**	**0.8691**
Mistral	Base	0.2885	0.7715	0.1066	0.7854	0.3502	0.7075	0.1873	0.7016	0.2331	0.7415
	RLVR	0.2753	0.7932	0.1222	0.7907	0.3425	0.6979	0.1944	0.7078	0.2336	0.7474
	CSR	**0.1239**	**0.8647**	**0.0703**	**0.8561**	**0.3019**	**0.7364**	**0.1108**	**0.7624**	**0.1517**	**0.8049**

Table 8: Relative change (%) against Base, computed from Table 1 and averaged over the three model families. ECE red. is the relative ECE reduction, so positive values indicate improvement. **Bold** marks the best and underline the second-best value in each column.

	In-distribution (HotpotQA)			**Out-of-distribution** (avg. of 3)		
Method	Acc↑	ECE red.↑	AUROC↑	Acc↑	ECE red.↑	AUROC↑
RLVR	**+32.1**	+5.2	+5.5	**+0.3**	−9.0	−0.6
RD	+13.3	+49.5	+11.0	−16.9	+7.0	+0.1
RLCR	+20.8	+26.7	−8.5	−2.5	+2.4	−8.6
CSR	+22.7	**+63.8**	**+17.4**	−1.1	**+31.1**	**+7.7**

Results. The per-dataset values are consistent with the averages of Table 2. CSR is not best in 9 of the 96 per-dataset cells (3 on Llama, 6 on Mistral), none on Qwen. RLVR, by contrast, is often no better than Base, so correctness training alone does not produce the effect.

D.2 Quantifying the accuracy-calibration trade-off

The calibration reward alone does not reward correctness (Proposition 1, Fig. 5(a)), so the combined objective may cost accuracy. From Table 1 we compute, for each method and each metric, the relative change against Base, averaged over the three model families, and split into in-distribution (HotpotQA) and out-of-distribution (computed on the metric averaged over the other three datasets). We use $(\text{method} - \text{Base})/\text{Base}$ for accuracy and AUROC and the relative ECE reduction $(\text{Base} - \text{method})/\text{Base}$, so that a positive value always indicates improvement (Table 8).

Results. CSR improves ECE and AUROC over Base by the largest margin in both settings, reducing ECE by 63.8% in distribution and 31.1% out of distribution. CSR costs some accuracy relative to RLVR, and the cost is modest. In distribution, CSR raises accuracy over Base by 22.7% and RLVR by 32.1%: measured as relative gains over Base averaged over the three backbones, CSR keeps about 70% of RLVR's accuracy gain. Out of distribution, neither method changes accuracy much: CSR lowers it by 1.1% and RLVR raises it by 0.3%.

D.3 Selective prediction

Per-dataset AURAC and AUGRC (Appendix B.3) are reported in Table 9, together with the four-dataset averages of Table 3.

Results. CSR is best in 22 of the 24 per-dataset cells and has the best four-dataset average of both metrics on every backbone. Its lead in average AURAC over the strongest baseline, RLVR, ranges from 0.014 on Llama to 0.048 on Qwen. The verbalized baselines rank questions less reliably: RLCR

Table 9: Per-dataset selective prediction with the four-dataset average. Confidence is verbalized for RD and RLCR and the semantic confidence for the other methods. **Bold** marks the best and underline the second-best value in each column within each model block.

		In-distribution		Out-of-distribution							
		HotpotQA		TriviaQA		MSMARCO		NQ-Open		Average	
Model	**Method**	AURAC↑	AUGRC↓	AURAC↑	AUGRC↓	AURAC↑	AUGRC↓	AURAC↑	AUGRC↓	AURAC↑	AUGRC↓
Llama	Base	0.2716	0.3783	0.8621	0.0853	0.5053	**0.2672**	0.6366	0.2121	0.5689	0.2357
	RLVR	0.5023	0.2811	0.8578	0.0838	**0.5150**	0.2674	0.6128	0.2160	0.6220	0.2121
	RD	0.4668	0.3085	0.8280	0.1029	0.4237	0.3128	0.5496	0.2670	0.5670	0.2478
	RLCR	0.4103	0.3204	0.7866	0.1224	0.4026	0.3157	0.5458	0.2547	0.5363	0.2533
	CSR	**0.5060**	**0.2779**	**0.8840**	**0.0791**	0.5083	0.2745	**0.6453**	**0.2099**	**0.6359**	**0.2104**
Qwen	Base	0.4170	0.3229	0.7137	0.1662	0.4348	0.2972	0.4287	0.3158	0.4986	0.2755
	RLVR	0.4445	0.3089	0.7150	0.1652	0.4435	0.2942	0.4248	0.3149	0.5070	0.2708
	RD	0.4520	0.3187	0.7200	0.1650	0.4049	0.3199	0.4441	0.3171	0.5053	0.2802
	RLCR	0.3545	0.3534	0.7236	0.1595	0.4501	0.2957	0.4109	0.3108	0.4848	0.2798
	CSR	**0.4686**	**0.3020**	**0.7680**	**0.1514**	**0.5039**	**0.2793**	**0.4780**	**0.3079**	**0.5546**	**0.2602**
Mistral	Base	0.3991	0.3278	0.8288	0.0967	0.5009	0.2672	0.5745	0.2365	0.5758	0.2320
	RLVR	0.4893	0.2838	0.8277	0.0973	0.5026	0.2664	0.5771	0.2365	0.5992	0.2210
	RD	0.4978	0.3020	0.7932	0.1485	0.2933	0.3832	0.3717	0.3588	0.4890	0.2981
	RLCR	0.4130	0.3121	0.7555	0.1402	0.4008	0.3110	0.5157	0.2611	0.5212	0.2561
	CSR	**0.5309**	**0.2763**	**0.8710**	**0.0815**	**0.5279**	**0.2594**	**0.6273**	**0.2177**	**0.6393**	**0.2087**

is worse than Base on both metrics on every backbone, and RD has the lowest average AURAC on Mistral, where it often abstains.

Relation to output-based calibration and refusal. Output-based signals, such as verbalized confidence or knowledge-aware refusal [26], are read from what the model emits for a single generation, so methods that train them act on that reporting behavior. CSR reads confidence from the distribution of K sampled answers, so training acts on the answer distribution itself. Both support abstention or deferral when the model is likely to be wrong. The output-based signal needs only one generation but can vary across semantically equivalent outputs (Fig. 1), whereas the semantic confidence needs K samples and does not depend on a reported confidence. The results above show better confidence-thresholded selective prediction, and we do not claim that CSR improves spontaneous refusal or verbalized confidence.

D.4 Context-grounded multi-hop QA

Setup. Table 1 is closed-book, so the model must recall each answer. Here the supporting passages are supplied in training and evaluation, so the answer can be composed from evidence, which is also the setting RLCR is trained in. Every method sees the full paragraph set at evaluation. Base, RLVR, and CSR use the system prompt "Provide a concise and direct answer to the question. Use the provided context and reason step-by-step. End with `Final answer: <answer>`.", RD uses `Context: <passages>, Question: <question>` as its user turn, and RLCR keeps its own format. Training uses HotpotQA only. MuSiQue [31] and the compositional questions of 2WikiMultiHopQA [14] serve as out-of-distribution test sets, presented in the same way with their own supporting and distractor paragraphs. For each, we shuffle the validation split with seed 42 and use the first $1{,}000$ questions, for 2WikiMultiHopQA after keeping only the compositional questions. CSR and RLVR are retrained on HotpotQA with the paragraphs in the prompt, a maximum prompt length of 3072 tokens, and a completion budget of 1024 tokens. RD is retrained on the same data with no separate prompt cap, only a maximum sequence length of 4096 tokens. RLCR is the checkpoint of Table 1, which is already trained with paragraphs, so the two tables differ only in whether it is evaluated with them. Evaluation uses a maximum sequence length of 4096 tokens on HotpotQA, 6144 on MuSiQue, and 5120 on 2WikiMultiHopQA, whose passages are longer, and the generation budgets of Appendix B.1.

Results. Averaged over the three datasets, CSR has the lowest ECE, 0.0852, and the highest AUROC, 0.7892, and RLVR is second on both (Table 10). In distribution, CSR has the lowest ECE and the highest AUROC and raises accuracy over Base by 0.097, but RLVR is slightly more accurate and CSR leads it by at most 0.014 on the other two metrics. With the passages in context, Base already

Table 10: Context-grounded QA with Qwen2.5-7B-Instruct, trained on HotpotQA and evaluated on HotpotQA (in-distribution) and on MuSiQue and 2WikiMultiHopQA (out-of-distribution). Average is the macro-average over the three datasets. **Bold** marks the best and underline the second-best value in each column.

	In-distribution			Out-of-distribution							
	HotpotQA			MuSiQue			2WikiMultiHopQA			Average	
Method	Acc↑	ECE↓	AUROC↑	Acc↑	ECE↓	AUROC↑	Acc↑	ECE↓	AUROC↑	ECE↓	AUROC↑
Base	0.6725	0.0751	0.7588	0.3308	0.1573	0.7038	0.4873	0.1908	0.6768	0.1411	0.7131
RLVR	**0.7774**	0.0711	0.7667	**0.4482**	0.0608	0.8220	0.5863	0.1492	0.7504	0.0937	0.7797
RD	0.6860	0.0781	0.6989	0.3390	0.1529	0.7084	0.4380	**0.0976**	**0.7666**	0.1095	0.7246
RLCR	0.6970	0.1242	0.6650	0.3450	0.0735	0.7208	0.5370	0.1065	0.7438	0.1014	0.7099
CSR	0.7692	**0.0587**	**0.7807**	0.4198	**0.0564**	**0.8268**	**0.5923**	0.1404	0.7601	**0.0852**	**0.7892**

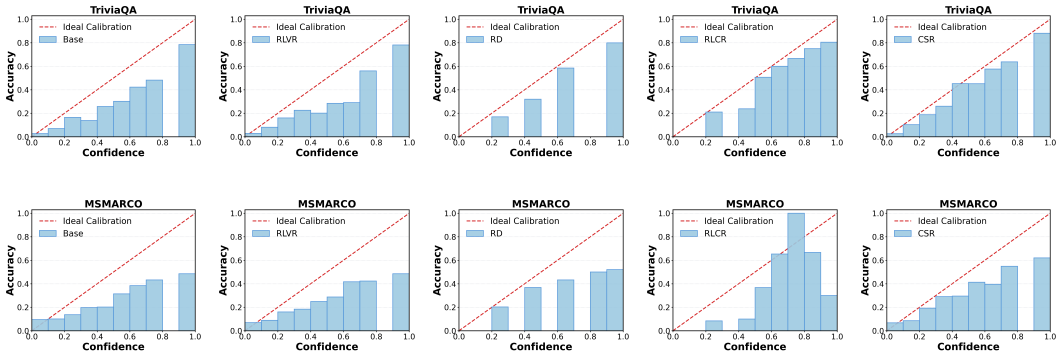


Figure 6: Reliability diagrams on Qwen for TriviaQA (top) and MSMARCO (bottom), both out of distribution. Columns are Base, RLVR, RD, RLCR, and CSR. Each bar is the accuracy of the questions whose confidence falls in a bin of width 0.1.

reaches an ECE of 0.0751 on HotpotQA, against 0.3050 closed-book, so little calibration error is left for the calibration term to remove. Base is less calibrated on 2WikiMultiHopQA, with an ECE of 0.1908, and there CSR lowers ECE from 0.1492 for RLVR to 0.1404 and raises AUROC from 0.7504 to 0.7601. RD has the best ECE and AUROC on this dataset, while CSR has the highest accuracy. On MuSiQue, CSR has the lowest ECE and the highest AUROC.

D.5 Reliability diagrams

Fig. 6 shows reliability diagrams on TriviaQA and MSMARCO, the two OOD datasets not shown in Fig. 4. On TriviaQA, CSR's bars stay close to the diagonal across the confidence range, consistent with the lowest ECE of the five methods, 0.0855, while Base and RLVR fall well below the diagonal from the middle bins upward. On MSMARCO, every method is overconfident in its highest-confidence bins, but CSR's accuracy still rises steadily with confidence, which matches its highest AUROC on this dataset, 0.8001.

D.6 Transfer to larger backbones

Setup. The backbones in Table 1 are all in the 7–8B range, so we repeat the protocol on Qwen3-14B and Qwen3-32B [38], roughly two and four times that scale. Thinking mode is disabled in training and evaluation for every method except RLCR, whose output format places its reasoning in `<think>` tags and so requires it. All methods follow the training and evaluation settings of Appendix B.1, except that on Qwen3-14B CSR's calibration weight ranges from $\lambda_{\min} = 0.5$ to $\lambda_{\max} = 0.7$ instead of from 0.8 to 1.2, because with the original range the calibration reward lowered ECE further but also lowered accuracy. Qwen3-32B uses the original range.

Table 11: Transfer to Qwen3-14B and Qwen3-32B across HotpotQA and three out-of-distribution datasets. **Bold** marks the best and underline the second-best value in each column within each model block.

		In-distribution			Out-of-distribution									Average		
		HotpotQA			TriviaQA			MSMARCO			NQ-Open					
Model	Method	Acc↑	ECE↓	AUROC↑	Acc↑	ECE↓	AUROC↑	Acc↑	ECE↓	AUROC↑	Acc↑	ECE↓	AUROC↑	Acc↑	ECE↓	AUROC↑
Qwen3 14B	Base	0.2404	0.3528	0.8019	0.6043	0.1820	0.8017	0.3808	0.3000	0.6859	0.3666	0.2585	0.7467	0.3980	0.2733	0.7590
	RLVR	0.2897	0.3220	0.8362	0.6035	0.1918	0.7832	0.3783	0.3107	0.6933	0.3632	0.2817	0.7374	0.4087	0.2766	0.7625
	RD	**0.2950**	0.2080	0.7506	0.5910	0.1300	0.8081	0.3420	0.3134	0.7304	0.3180	0.3469	0.6886	0.3865	0.2496	0.7444
	RLCR	0.2640	0.4384	0.7349	**0.6690**	0.1750	0.8260	**0.4090**	0.3726	0.7284	**0.4230**	0.3899	0.6951	**0.4413**	0.3440	0.7461
	CSR	0.2717	**0.1430**	**0.8999**	0.6006	**0.0869**	**0.8718**	0.3790	**0.2296**	**0.7355**	0.3691	**0.1359**	**0.8004**	0.4051	**0.1488**	**0.8269**
Qwen3 32B	Base	0.2386	0.2697	0.8201	0.6451	0.1061	0.8571	0.3885	0.2386	0.7525	0.3894	0.1688	0.7791	0.4154	0.1958	0.8022
	RLVR	**0.3115**	0.2255	0.8488	0.6546	0.1025	0.8513	0.3822	0.2470	0.7430	0.3946	0.1721	0.7824	0.4357	0.1868	0.8064
	RD	0.3000	0.1451	0.7712	0.6560	0.2488	0.7612	0.3970	**0.1029**	0.7010	0.3800	**0.0880**	0.6680	0.4333	0.1462	0.7254
	RLCR	0.2750	0.5315	0.7349	**0.7080**	0.1957	0.7612	**0.4150**	0.4520	0.6991	**0.4400**	0.4485	0.6692	**0.4595**	0.4069	0.7161
	CSR	0.2980	**0.1300**	**0.9023**	0.6514	**0.0642**	**0.8748**	0.3793	0.2071	**0.7638**	0.3938	0.1187	**0.7843**	0.4306	**0.1300**	**0.8313**

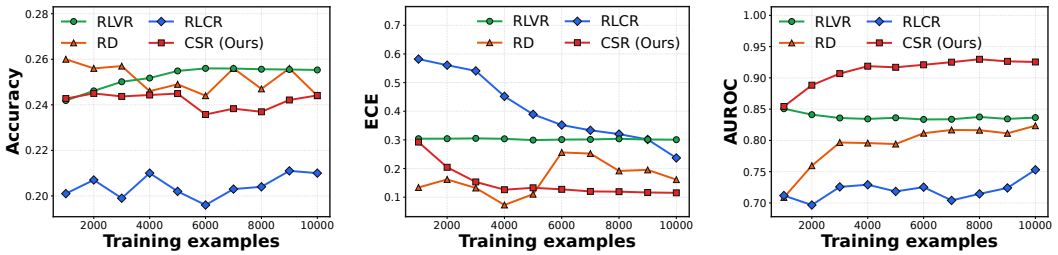


Figure 7: Sample efficiency on HotpotQA with Qwen2.5-7B-Instruct at matched training budgets of 1k to 10k examples. Left to right: accuracy, ECE, and AUROC.

Results. The main trends of Table 1 carry over to both larger backbones (Table 11). On Qwen3-14B, CSR attains the lowest ECE and the highest AUROC on every dataset, and it raises average AUROC from 0.7590 for Base to 0.8269. On Qwen3-32B, CSR has the highest AUROC on every dataset, and it raises average AUROC from 0.8022 for Base to 0.8313. On HotpotQA, it lowers ECE from 0.2697 for Base and 0.2255 for RLVR to 0.1300. CSR also has the lowest average ECE, although RD is lower on MSMARCO and NQ-Open. On both backbones, RLCR has the highest average accuracy, measured on a single greedy answer, but more than twice CSR's average ECE, and RLVR again stays close to Base in average ECE and AUROC.

D.7 Sample efficiency under matched training budgets

Setup. To separate CSR's effect from the size of its training budget, we evaluate every method on HotpotQA with Qwen2.5-7B-Instruct at matched budgets of 1k to 10k examples. The intermediate budgets are checkpoints of each method's 10k run, so the $\lambda(t)$ and learning-rate schedules are those of the full run, and all other settings follow Appendix B.1. The CSR curve comes from the same run as the sigmoid arm in Fig. 5(b), and the 10k points of the baselines are the Qwen HotpotQA cells of Table 1.

Results. CSR attains the highest AUROC at every budget and the lowest ECE from 6k on (Fig. 7). Its gain arrives early: its ECE falls from about 0.29 at 1k to about 0.13 at 4k, after which its curves flatten. RLVR's accuracy instead rises slightly with more data, from about 0.24 to 0.26, while its ECE stays near 0.30. CSR's accuracy stays near 0.24, so the extra data mainly improves calibration for CSR and, modestly, accuracy for RLVR. RD has the lowest ECE up to 5k but ends above CSR, while its AUROC rises and its accuracy falls. RLCR lowers ECE, but its AUROC stays below CSR's and RLVR's.

Practical guidance on training data. Within the 10k run, most of CSR's gain is reached by about 4k examples: from 4k to 10k, its ECE falls by only about 0.01 and its AUROC rises by less than 0.01. These are checkpoints of a single run, so the curve suggests that a few thousand questions (roughly 4k to 6k) suffice on a task of similar difficulty. Which questions are used also matters. The correctness term gives a group-relative signal only when a rollout group mixes correct and incorrect answers,

Table 12: LLM-judge and F1 equivalence on all three backbones. The Llama rows repeat Table 4. **Bold** marks the best and underline the second-best value in each column within each model block.

		In-distribution		Out-of-distribution							
		HotpotQA		TriviaQA		MSMARCO		NQ-Open		Average	
Model	Method	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑	ECE↓	AUROC↑
Llama	RD	0.1764	0.8316	0.0746	0.7996	0.3302	0.7298	0.2081	0.7464	0.1973	0.7769
	RLCR	0.1757	0.6923	0.0943	0.7337	0.2340	0.6816	0.1497	0.7013	0.1634	0.7022
	Base-LLM	0.2949	0.7050	0.0772	0.8423	0.1996	0.7062	0.1186	0.7361	0.1726	0.7474
	RLVR-LLM	0.2567	0.8089	0.0901	0.8268	0.2280	0.7371	0.1686	0.6957	0.1859	0.7671
	CSR-LLM	**0.0978**	**0.9133**	**0.0670**	**0.8976**	0.1571	**0.7594**	**0.1100**	0.7670	**0.1080**	**0.8343**
	Base-F1	0.2713	0.6681	0.1212	0.7929	0.1612	0.6094	0.1223	0.6992	0.1690	0.6924
	RLVR-F1	0.2727	0.7770	0.0899	0.7819	0.2296	0.6670	0.1798	0.6771	0.1930	0.7258
	CSR-F1	0.1240	0.9008	0.1005	0.8498	**0.1042**	0.7223	0.1109	**0.7735**	0.1099	0.8116
Qwen	RD	0.1613	0.8235	0.0876	0.8013	**0.1008**	0.6531	0.1380	0.7606	0.1219	0.7596
	RLCR	0.2370	0.7530	0.0894	0.7293	0.2031	0.7026	0.2671	0.6255	0.1991	0.7026
	Base-LLM	0.3050	0.8413	0.1881	0.8281	0.2945	0.7037	0.2487	0.7773	0.2591	0.7876
	RLVR-LLM	0.3007	0.8365	0.1933	0.8241	0.2871	0.7094	0.2492	0.7594	0.2576	0.7824
	CSR-LLM	0.1152	**0.9260**	0.0855	**0.9094**	0.1560	**0.8001**	0.0866	**0.8348**	0.1108	**0.8676**
	Base-F1	0.2650	0.8206	0.1500	0.8166	0.1862	0.6079	0.1870	0.7489	0.1970	0.7485
	RLVR-F1	0.2684	0.8250	0.1485	0.8157	0.1868	0.6126	0.1756	0.7364	0.1948	0.7474
	CSR-F1	**0.1132**	0.9079	**0.0531**	0.9019	0.1761	0.6509	**0.0659**	0.8031	**0.1021**	0.8159
Mistral	RD	0.0994	**0.9065**	0.1061	0.8118	0.1879	0.7014	0.1395	**0.7897**	0.1332	0.8024
	RLCR	0.2117	0.6715	**0.0611**	0.6913	0.2669	0.6529	0.1942	0.6856	0.1835	0.6753
	Base-LLM	0.2563	0.7745	0.0952	0.7860	0.3011	0.7087	0.1507	0.7035	0.2008	0.7432
	RLVR-LLM	0.2533	0.7936	0.1126	0.7887	0.2939	0.7003	0.1635	0.7079	0.2058	0.7476
	CSR-LLM	**0.0966**	0.8710	0.0751	**0.8530**	0.2513	**0.7351**	**0.0974**	0.7606	**0.1301**	**0.8049**
	Base-F1	0.3901	0.6533	0.1692	0.6571	0.1946	0.6000	0.1722	0.6677	0.2315	0.6445
	RLVR-F1	0.3477	0.6991	0.1211	0.7179	0.1829	0.5914	0.1900	0.6558	0.2104	0.6660
	CSR-F1	0.1098	0.8523	0.1522	0.8127	**0.1407**	0.6041	0.1235	0.7041	0.1316	0.7433

and only then can the calibration reward separate agreement among correct answers from agreement among incorrect ones. We therefore recommend corpora on which the backbone is right on some rollouts and wrong on others, with reliably verifiable answers. All our experiments use short-answer QA. We expect similar gains in other domains where rollout groups mix correct and incorrect answers that can be reliably verified, and smaller gains where they rarely do, while long-form generation and code would additionally need an equivalence judge that compares longer outputs.

D.8 Token cost across methods

Table 1 reports the average total token cost per question alongside the calibration metrics. RD has the lowest cost on Llama and Mistral because it decodes a single short answer, whereas Base, RLVR, and CSR decode $K = 16$. RLCR costs more than CSR on every backbone, because its prompt is long, its output format has many required fields, and it generates a reasoning trace before answering. The gap is largest on Qwen, where RLCR uses 641.07 tokens per question and CSR uses 198.54, so RLCR costs about $3.2\times$ as much. On Llama and Mistral the gap is smaller, with RLCR costing about 15% to 18% more than CSR.

Each CSR rollout is an answer-only completion comparable in length to a Base or RLVR generation, so CSR's total per-question cost scales with K. Tok counts policy tokens only. The $\binom{K}{2}$ equivalence judgments that Base, RLVR, and CSR use at inference are accounted for in Appendix D.9.

D.9 Token-level F1 as a surrogate for the semantic-equivalence judge

Judging cost. With $K = 8$ rollouts, each training question requires 8 correctness judgments against the gold answer and $\binom{K}{2} = 28$ pairwise judgments, giving at most 36 judge calls per question and at most $360{,}000$ calls across a full 10k-example epoch. Evaluation uses $K = 16$, so $\binom{16}{2} + 16 = 136$ judgments per question and $136{,}000$ calls per dataset at $n = 1{,}000$. In a timed training run on Llama-

Table 13: Format-valid rate (Valid) and abstention rate (Abstain) on the runs of Table 1. Valid is the fraction of answers with a parsed, in-range confidence for RD and RLCR, and the answer-extraction rate for Base, RLVR, and CSR, which emit no confidence. Abstain is the fraction of responses that are abstentions, over 16,000 rollouts per cell for Base, RLVR, and CSR and 1,000 greedy answers for RD and RLCR. **Bold** marks the best value in each column within each model block.

		HotpotQA		TriviaQA		MSMARCO		NQ-Open	
Model	**Method**	Valid↑	Abstain↓	Valid↑	Abstain↓	Valid↑	Abstain↓	Valid↑	Abstain↓
Llama	Base	**1.000**	0.126	**1.000**	0.011	**1.000**	0.004	**1.000**	0.017
	RLVR	**1.000**	0.053	**1.000**	0.004	**1.000**	0.002	**1.000**	0.010
	RD	0.999	0.069	**1.000**	0.008	0.998	0.004	0.994	0.005
	RLCR	0.969	0.043	0.941	0.007	0.982	0.008	0.952	0.038
	CSR	**1.000**	**0.001**	**1.000**	**0.001**	**1.000**	**0.000**	**1.000**	**0.001**
Qwen	Base	**1.000**	0.003	**1.000**	0.001	**1.000**	**0.000**	**1.000**	0.001
	RLVR	**1.000**	0.003	**1.000**	0.001	**1.000**	**0.000**	**1.000**	0.001
	RD	**1.000**	0.093	**1.000**	0.010	0.999	0.013	0.996	0.037
	RLCR	0.995	0.128	0.996	0.008	**1.000**	0.024	0.999	0.029
	CSR	**1.000**	**0.001**	**1.000**	**0.000**	**1.000**	**0.000**	**1.000**	**0.000**
Mistral	Base	**1.000**	0.002	**1.000**	**0.000**	**1.000**	0.001	**1.000**	0.004
	RLVR	**1.000**	0.001	**1.000**	**0.000**	**1.000**	**0.000**	**1.000**	0.003
	RD	**1.000**	0.345	**1.000**	0.099	0.997	0.121	0.998	0.276
	RLCR	0.999	0.042	0.999	**0.000**	**1.000**	0.005	**1.000**	0.008
	CSR	**1.000**	**0.000**	**1.000**	**0.000**	**1.000**	**0.000**	**1.000**	**0.000**

3.1-8B-Instruct, judging accounted for about one third (33.2%) of the wall-clock. At deployment, the K correctness judgments against gold answers are unnecessary, but semantic entropy with the LLM judge still uses $\binom{K}{2}$ equivalence judgments to cluster the sampled answers. For short-answer tasks, the F1 surrogate below removes those remaining LLM calls by replacing the semantic-equivalence test used to form the clusters.

F1 equivalence. We approximate semantic equivalence by lexical overlap. For two sampled answers $Y^{(i)}$ and $Y^{(j)}$, we define an approximate equivalence label by thresholding their token-level F1:

$$\tilde{s}_{ji} = \mathbf{1}\Big[\mathrm{F1}\Big(Y^{(j)}, Y^{(i)}\Big) \geq \tau\Big]. \tag{26}$$

We replace the original semantic-equivalence signal in the semantic calibration reward with $\tilde{s}_{ji}$ (in place of s_{ji} in Eq. (5)) and use the same rule to form semantic clusters at evaluation. This keeps the reward structure while removing pairwise LLM-judge calls from both training and deployment.

Scope. Token-level F1 is a low-cost surrogate *for short, entity-style answers*, where semantically equivalent answers usually share most of their tokens, and it is not intended to replace the judge for long or compositional answers. Its characteristic failure is under-crediting paraphrase and alias equivalence. This matches the AUROC gap of Sec. 5.5.

Training and evaluation details. All other settings follow the main experiments, with the threshold of Sec. 5.5 fixed across datasets. Base-F1 and RLVR-F1 use F1 only at evaluation because their training objectives contain no semantic-equivalence term.

Results. The F1-equivalence variant preserves the main qualitative trend of CSR (Table 12). Against Base-F1 and RLVR-F1, CSR-F1 has higher AUROC in all 12 model-dataset comparisons and lower ECE in 10 of 12, the exceptions being TriviaQA for Llama and Mistral against RLVR-F1. Relative to CSR-LLM, AUROC is lower in 11 of 12 cells, all but Llama on NQ-Open.

D.10 Parsing and abstention rates

The verbalized baselines, RD and RLCR, require the model to emit a numeric confidence in a fixed format alongside its answer (a `Confidence:` field on a 0–10 scale, rescaled to $[0, 1]$, for RD and `<confidence>` tags for RLCR). Format failures occur when a generation exhausts its token budget before emitting the required fields or emits no recoverable confidence (Table 13).

RD and RLCR emit a parsable confidence on at least 94% of questions in every cell. Unparsable confidences are scored $c(x) = 0$ and affect up to 5.9% of RLCR's questions (Llama on TriviaQA, Valid 0.941) and at most 0.6% of RD's, so they are rare but can shift RLCR's metrics in the affected cells of Table 1. Abstention accounts for part of RD's lower accuracy on Mistral. We count a response as an abstention when its extracted answer, lowercased and stripped of punctuation, is an abstention phrase such as "unknown" or "I don't know", or begins with a refusal such as "I'm sorry". A bare "none" is not counted, since it can be the correct answer. Abstentions are scored as incorrect. Mistral RD abstains on 34.5% of HotpotQA and 27.6% of NQ-Open questions, whereas CSR abstains on at most 0.1% of responses in any cell and Base on at most 1.7%, except on HotpotQA with Llama, where 12.6% of its rollouts are refusals. RLCR abstains on 12.8% of HotpotQA questions with Qwen and on at most 4.3% elsewhere.

Mistral RD's abstention is consistent with the form of its reward. The logarithmic scoring rule scores the stated confidence against the outcome, so an abstention stated with a confidence of 0 earns the same score as the best-scored incorrect answer. Llama and Mistral both learn to state 0, on 32.6% and 47.2% of HotpotQA questions, but Llama abstains on only 6.9%, so it still writes an answer in most of these cases. CSR gives abstention no such advantage, because an abstention earns nothing from its correctness term.

E Qualitative rollout examples

We provide qualitative rollout-level samples from CSR, Base, RLVR, RD, and RLCR on Llama-3.1-8B-Instruct, generated with the prompts in Appendix B.4. Within each box, every numbered line $[k]$ shows one sampled rollout, annotated with ✓ or ✗ according to the LLM judge against the gold answer(s). For Base, RLVR, and CSR, whose semantic confidence comes from $K = 16$ rollouts as in Table 1, we display all sixteen rollouts. For RD and RLCR we display the single greedy output with its verbalized confidence. Correct answers should receive high confidence (Appendix E.1) and wrong answers low confidence (Appendix E.2). Bold text only highlights the answers.

E.1 Correct and consistent rollout groups

When CSR answers a question correctly, its rollouts concentrate in one semantic cluster and the semantic confidence is high, reaching 1.000 when every rollout falls in that cluster. The two examples below show that one cluster is not the same as one string: the rollouts differ in surface form and the semantic-equivalence judge merges them by meaning, so the confidence reflects agreement in meaning rather than a repeated output.

E.1.1 HotpotQA (in-distribution)

Question

Charles Haughey held what position when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989?
Ground Truth Answer: Taoiseach of Ireland

The question asks for Haughey's office. CSR answers it in fourteen distinct strings, from the bare "Taoiseach (Prime Minister)" to a full restatement of the question ("Charles Haughey held the position of Taoiseach (Prime Minister) when he dissolved . . . "). The semantic-equivalence judge places all sixteen rollouts in one cluster, so its accuracy and confidence are both 1.000. Base and RLVR reach the same values with seven distinct strings each, and six of RLVR's sixteen rollouts are the same sentence. The verbalized baselines are also correct, with RD and RLCR both answering "Taoiseach".

CSR (Ours) generated answer

[0] **Taoiseach (Prime Minister)** ✓
[1] He was the **Taoiseach (Prime Minister** of Ireland) ✓
[2] He was the **Taoiseach (Prime Minister)**. ✓
[3] **Prime Minister (Taoiseach)** ✓

[4] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[5] The **Taoiseach** (**Prime Minister**). ✓
[6] Charles Haughey was the **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[7] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[8] He was the **Taoiseach** of Ireland ✓
[9] He was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[10] Charles Haughey held the position of **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[11] Charles Haughey was the **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[12] **Taoiseach** (**Prime Minister** of Ireland) ✓
[13] **Taoiseach** (**Prime Minister**). ✓
[14] He held the position of **Taoiseach** (**Prime Minister**). ✓
[15] **Taoiseach** (**Prime Minister** of Ireland) ✓

Accuracy: 1.000 **Confidence:** 1.000

Base generated answer

[0] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[1] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[2] Charles Haughey was the **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[3] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[4] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[5] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[6] Charles Haughey was the **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[7] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[8] Charles Haughey was the **Taoiseach** (**Prime Minister** of Ireland) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[9] Charles Haughey was the **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[10] Charles Haughey was the **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[11] Charles Haughey was the **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓

[12] Charles Haughey was the **Taoiseach** (**Prime Minister** of Ireland) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[13] Charles Haughey was the **Taoiseach**. ✓
[14] Charles Haughey held the position of **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[15] Charles Haughey was the **Taoiseach** of Ireland. ✓

Accuracy: 1.000 **Confidence:** 1.000

RLVR generated answer

[0] **Taoiseach**. ✓
[1] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[2] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[3] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[4] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[5] The **Prime Minister** of Ireland. ✓
[6] Charles Haughey was the **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[7] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[8] Charles Haughey was the **Taoiseach** (**Prime Minister** of Ireland). ✓
[9] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[10] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland. ✓
[11] Charles Haughey was the **Taoiseach** (**Prime Minister**) when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓
[12] Charles Haughey was the **Taoiseach**. ✓
[13] Charles Haughey was the **Taoiseach**. ✓
[14] Charles Haughey was the **Taoiseach**. ✓
[15] Charles Haughey was the **Taoiseach** (**Prime Minister**) of Ireland when he dissolved the list of members who were elected to the lower house of the Oireachtas of Ireland on 25 May 1989. ✓

Accuracy: 1.000 **Confidence:** 1.000

RD generated answer

[0] Answer: **Taoiseach**, Confidence: 8 ✓

Accuracy: 1.000 **Confidence:** 0.800

RLCR generated answer

[0] <think> To solve this problem, I'll first recall that Charles Haughey was the Taoiseach (Prime Minister) of Ireland from 1981 to 1982 and again from 1987 to 1992. I'll then consider the date given in the question, which is 25 May 1989. I'll think about the fact that the question is asking about the dissolution of the lower house of the Oireachtas, which is the Dáil Éireann. I'll recall that the Taoiseach has the power to dissolve the Dáil and call a general election. Given that Charles Haughey was the Taoiseach at the time, I'll reason that he was likely the one who dissolved the list of members who were elected to the lower house of the Oireachtas on 25 May 1989. </think>

<answer>**Taoiseach**</answer> <analysis> There are a few uncertainties in this solution. Firstly, I'm assuming that Charles Haughey was still the Taoiseach in May 1989, but I'm not certain about his exact tenure as Taoiseach. Secondly, I'm assuming that the dissolution of the list of members was a direct action by the Taoiseach, but it's possible that there were other factors at play. Additionally, there could be alternative approaches to solving this problem, such as looking at the specific legislation or events surrounding the dissolution of the Dáil in 1989. </analysis> <confidence> **0.62** </confidence> ✓

Accuracy: 1.000 **Confidence:** 0.620

E.1.2 NQ-Open (out-of-distribution)

Question

how many seasons of the smurfs are there
Ground Truth Answer: 9

The same behavior holds out of distribution, where the rollouts range from the bare number to a full sentence and still form a single cluster. Base answers correctly on fifteen of its sixteen rollouts, and the one that invents a second series and assigns nine seasons to each forms a second cluster and lowers its confidence to 0.792, so the dispersion tracks the rollout it gets wrong.

CSR (Ours) generated answer

[0] There are **9 seasons** of the Smurfs animated television series, ✓
[1] There are **9 seasons** of the Smurfs. ✓
[2] There are **9 seasons** of the Smurfs (Smurfs), ✓
[3] There are **9 seasons** of the Smurfs. ✓
[4] There are **9 seasons** of the animated television series 'The Smurfs', ✓
[5] **9** ✓
[6] **9** ✓
[7] **9 seasons** ✓
[8] There are **9 seasons** of the Smurfs. ✓
[9] **9 seasons** ✓
[10] There are **9 seasons** of the Smurfs animated television series. ✓
[11] There are **9 seasons** of the Smurfs TV series. ✓
[12] There are **9 seasons** of the original Smurfs cartoon series that aired from 1981 to 1989. ✓
[13] There are **9 seasons** of The Smurfs ✓
[14] **9** ✓
[15] There are **9 seasons** of the Smurfs. ✓

Accuracy: 1.000 **Confidence:** 1.000

Base generated answer

[0] There are **9 seasons** of the classic animated television series ''The Smurfs'' (also known as ''Smurf'' or ''Smurfs''), which aired from 1981 to 1990. ✓
[1] There are **9 seasons** of the Smurfs. ✓
[2] There are **9 seasons** of the classic animated series ''The Smurfs'' (also known as ''Smurf'' or ''Smurfs'') ✓
[3] There are **9 seasons** of the Smurfs. ✓
[4] There are **9 seasons** of the animated TV series ''The Smurfs''. ✓

[5] There are **9 seasons** of the original Smurfs animated series, which aired from 1981 to 1990. ✓
[6] There are **9 seasons** of the Smurfs. ✓
[7] There are **9 seasons** of the animated TV series ‘‘The Smurfs’’. ✓
[8] There are **9 seasons** of the Smurfs TV show. ✓
[9] There are **9 seasons** of the original Smurfs animated series (1981-1990). ✓
[10] There are **9 seasons** of the Smurfs animated television series that aired from 1981 to 1990. ✓
[11] There are **9 seasons** of the animated television series ‘‘The Smurfs’’ (also known as ‘‘The Smurfs’’ or ‘‘Smurfs’’). ✓
[12] There are **9 seasons** of the original Smurfs cartoon series that aired from 1981 to 1990. ✓
[13] There are **9 seasons** of the Smurfs animated television series, which aired from 1981 to 1990. ✓
[14] There are **9 seasons** of the original animated series ‘‘The Smurfs’’ (1981-1990), and a total of **9 seasons** of the new CGI-animated series ‘‘The Smurfs’’ (2011-2019). ✗
[15] There are **9 seasons** of the original Smurfs cartoon series. ✓

Accuracy: 0.938 **Confidence:** 0.792

RLVR generated answer

[0] There are **9 seasons** of the Smurfs. ✓
[1] There are **9 seasons** of the Smurfs. ✓
[2] There are **9 seasons** of the Smurfs. ✓
[3] There are **9 seasons** of the Smurfs. ✓
[4] There are **9 seasons** of the Smurfs. ✓
[5] There are **9 seasons** of the original Smurfs animated TV series. ✓
[6] There are **9 seasons** of the Smurfs. ✓
[7] There are **9 seasons** of the Smurfs. ✓
[8] There are **9 seasons** of the Smurfs. ✓
[9] There are **9 seasons** of the original Smurfs cartoon series. ✓
[10] There are **9 seasons** of the Smurfs. ✓
[11] There are **9 seasons** of the Smurfs. ✓
[12] There are **9 seasons** of the Smurfs (1981-1990) ✓
[13] There are **9 seasons** of the Smurfs. ✓
[14] There are **9 seasons** of the Smurfs. ✓
[15] There are **9 seasons** of the Smurfs. ✓

Accuracy: 1.000 **Confidence:** 1.000

RD generated answer

[0] Answer: **9**, Confidence: 6 ✓

Accuracy: 1.000 **Confidence:** 0.600

RLCR generated answer

[0] <think> I’m thinking about the popular animated TV show ‘‘The Smurfs’’ and trying to recall the information about its seasons. I’m searching through my knowledge database to find the correct information. I’m considering the show’s initial run from 1981 to 1990 and any potential revivals or spin-offs. I’m also thinking about the show’s production history and the typical length of seasons for

animated TV series. </think> <answer>**There are 9 seasons of the original "The Smurfs" TV series**</answer> <analysis> Uncertainties:
1. I'm assuming the question is referring to the original ''The Smurfs'' TV series, but it's possible the user is asking about a specific spin-off or revival, which could have a different number of seasons.
2. I'm relying on my knowledge of the show's initial run, but there might be some ambiguity about the exact dates of the show's production and broadcast.
3. I'm considering the typical length of seasons for animated TV series, but there could be exceptions or variations in the case of ''The Smurfs''.
4. I'm not accounting for any potential differences in season length or episode count between the original series and any spin-offs or revivals. </analysis> <confidence> **0.62** </confidence> ✓

Accuracy: 1.000 **Confidence:** 0.620

E.2 Incorrect and dispersed rollout groups

The complementary case is a question CSR cannot answer. The rollouts then spread over many semantic clusters and the semantic confidence falls, which signals the error. The failure this guards against is a confident hallucination: a policy that commits to one wrong answer reports high confidence, which the calibration reward penalizes by rewarding incorrect rollouts for disagreeing. In both examples below no method answers correctly, and the methods differ only in how clearly their confidence says so.

E.2.1 HotpotQA (in-distribution)

Question

What memorial park located at John J. Pershing Drive in North Omaha has, among other amenities, a baseball park?
Ground Truth Answer: Dodge Memorial Park

CSR spreads its sixteen rollouts over eleven semantic clusters and reports 0.105, whereas RLVR collapses the same guesses onto four clusters and reports 0.313.

CSR (Ours) generated answer

[0] **Fontenberry Park** ✗
[1] **Fontenberry Memorial Park** ✗
[2] **Lauritzen Gardens** ✗
[3] **The memorial park at John J. Pershing Drive in North Omaha is Fontenelle Park.** ✗
[4] **Fontenello Park** ✗
[5] **Bubbling Springs Memorial Park** ✗
[6] **Bemis Park.** ✗
[7] **Fontenot Memorial Park** ✗
[8] **Bemis Park** ✗
[9] **Fontenberry Park** ✗
[10] **Fontenelle Park** ✗
[11] **Lake Manawa Memorial Park** ✗
[12] **The memorial park you are referring to is Elmwood Park.** ✗
[13] **Fontenberry Park** ✗
[14] **Standing Bear Park** ✗
[15] **Miller Park** ✗

Accuracy: 0.000 **Confidence:** 0.105

Base generated answer

[0] **Fontenberry Park** ✗
[1] **Fontenberry Park** ✗
[2] **Fontenello Park** ✗
[3] **Lake Manawa Park** ✗
[4] **Fontenello Park** ✗
[5] **Benson Memorial Park.** ✗
[6] **Fontenberry Park.** ✗
[7] **Fontenot Memorial Park** ✗
[8] **Bemis Park** ✗
[9] **Fontenberry Park** ✗
[10] **Fontenello Memorial Park** ✗
[11] **Lake Manawa Memorial Park** ✗
[12] **Fontenot Park** ✗
[13] **Fontenberry Park** ✗
[14] **Fontenberry Memorial Park.** ✗
[15] **Fontenello Park.** ✗

Accuracy: 0.000 **Confidence:** 0.206

RLVR generated answer

[0] **Fontenberry Park** ✗
[1] **Fontenberry Park** ✗
[2] **Fontenello Memorial Park** ✗
[3] **Fontenberry Park.** ✗
[4] **Fontenello Park** ✗
[5] **Fontenot Park** ✗
[6] **Fontenberry Park.** ✗
[7] **Fontenello Memorial Park** ✗
[8] **Fontenello Memorial Park** ✗
[9] **Fontenberry Park** ✗
[10] **Fontenello Memorial Park.** ✗
[11] **Lake Manawa Memorial Park** ✗
[12] **Fontenot Memorial Park.** ✗
[13] **Fontenberry Park** ✗
[14] **Fontenberry Memorial Park.** ✗
[15] **Fontenello Park.** ✗

Accuracy: 0.000 **Confidence:** 0.313

RD generated answer

[0] Answer: **Fontenello Park**, Confidence: 0 ✗

Accuracy: 0.000 **Confidence:** 0.000

RLCR generated answer

[0] <think> I'm thinking about the memorial park located at John J. Pershing Drive in North Omaha. I'm considering the amenities it might have, including a baseball park. I'm searching through my knowledge about parks in Omaha and their features. I'm looking for a match between the location and the amenities. I'm thinking about the possibility of the park being a well-known memorial park in Omaha. I'm recalling the names of parks in Omaha and their locations. I'm trying to find a connection between the park and the baseball field. </think>

<answer>**Fontenello Park**</answer> <analysis> There are several uncertainties in my reasoning process. Firstly, I'm relying on my knowledge about Omaha's parks, which might not be exhaustive. Secondly, I'm assuming that the park I'm thinking of is a well-known memorial park, but there could be other parks with similar features. Thirdly, I'm not considering the possibility of the park being a smaller or lesser-known park that still has a baseball field. Lastly, I'm not accounting for the possibility of the park being renamed or redeveloped since my knowledge cutoff. </analysis> <confidence> **0.52** </confidence> ✗

Accuracy: 0.000 **Confidence:** 0.520

E.2.2 NQ-Open (out-of-distribution)

Question

who said one man's vulgarity is another's lyric
Ground Truth Answer: Justice Harlan

Out of distribution the gap widens. RLVR returns the same wrong name in all sixteen rollouts, forms one cluster, and so reports 1.000. CSR spreads over eleven clusters and reports 0.113, so its confidence exposes the error.

CSR (Ours) generated answer

[0] **George Carlin** ✗
[1] **This quote is attributed to Robert A. Heinlein, an American science fiction author.** ✗
[2] **Lillian Hellman** ✗
[3] **E.B. White** ✗
[4] **Alfred Alvarez** ✗
[5] **Alexander Woollcott** ✗
[6] **George Carlin** ✗
[7] **George Carlin** ✗
[8] **George Carlin** ✗
[9] **H.L. Mencken** ✗
[10] **Oscar Wilde** ✗
[11] **George Carlin** ✗
[12] **Aldous Huxley** ✗
[13] **Justice Oliver Wendell Holmes Jr.** ✗
[14] **H.L. Mencken** ✗
[15] **Frederic Raphael** ✗

Accuracy: 0.000 **Confidence:** 0.113

Base generated answer

[0] **George Carlin** ✗
[1] **This quote is often attributed to Oscar Wilde.** ✗
[2] **George Carlin.** ✗
[3] **George Carlin.** ✗
[4] **Oscar Wilde** ✗
[5] **Lawrence J. Peter.** ✗
[6] **George Carlin.** ✗
[7] **George Carlin** ✗
[8] **George Carlin** ✗
[9] **H.L. Mencken** ✗
[10] **Oscar Wilde** ✗

[11] **George Carlin.** ✗
[12] **George Carlin** ✗
[13] **George Carlin** ✗
[14] **H.L. Mencken.** ✗
[15] **George Carlin** ✗

Accuracy: 0.000 **Confidence:** 0.353

RLVR generated answer

[0] **George Carlin** ✗
[1] **George Carlin** ✗
[2] **George Carlin.** ✗
[3] **George Carlin.** ✗
[4] **George Carlin** ✗
[5] **George Carlin** ✗
[6] **George Carlin.** ✗
[7] **George Carlin** ✗
[8] **George Carlin** ✗
[9] **George Carlin** ✗
[10] **George Carlin** ✗
[11] **George Carlin.** ✗
[12] **George Carlin** ✗
[13] **George Carlin** ✗
[14] **George Carlin** ✗
[15] **George Carlin** ✗

Accuracy: 0.000 **Confidence:** 1.000

RD generated answer

[0] Answer: **George Carlin**, Confidence: 7 ✗

Accuracy: 0.000 **Confidence:** 0.700

RLCR generated answer

[0] <think> I'm thinking about the origin of this quote, which seems to be a phrase that highlights the subjective nature of what is considered "vulgar" or "lyric." It appears to be a quote from an American author, and I'm trying to recall the exact words and context in which it was said. I'm considering the possible sources of this quote, such as literature, music, or even a famous speech. I'm also thinking about the possible authors who might have coined this phrase, such as Mark Twain or H.L. Mencken. I'm weighing the likelihood of this quote being a common saying or a unique phrase from a specific work. </think> <answer>**Oscar Wilde**</answer> <analysis> One uncertainty is that I'm relying on my training data to identify the quote, and there's a possibility that the quote might be misattributed or not accurately represented in my training data. Another uncertainty is that there might be multiple authors who have used similar phrases, making it difficult to pinpoint the exact origin of this quote. Additionally, the quote's meaning and context could be open to interpretation, leading to different perspectives on what is considered "vulgar" or "lyric." </analysis> <confidence> **0.62** </confidence> ✗

Accuracy: 0.000 **Confidence:** 0.620

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the claims, which are supported by the results in Section 5.2 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses the limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We give the proof of Proposition 1 in Appendix C.1.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide complete experiment setup, training and evaluation configurations in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use publicly available datasets and provide code at `https://github.com/Rose-STL-Lab/Semantic_Calibration`. The repository includes the implementation, evaluation scripts, and configuration files needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.1 summarizes the setup, and Appendix B.1 gives the full training and evaluation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars because repeating every RL training run across three backbones and four trained methods is computationally expensive (Appendix D.9). Base, RLVR, and CSR are evaluated on 16 sampled rollouts per question, while RD and RLCR use a single greedy answer.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Every training and evaluation run uses one NVIDIA A100 80GB GPU (Appendix B.1). Depending on the backbone, a full training run on HotpotQA takes about 9 to 32 GPU-hours for CSR, 7 to 21 for RLVR, 8 to 11 for RD, and 18 to 37 for RLCR.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Better-calibrated confidence can help users decide when to trust a model's answer, for example in medical or legal question answering. A negative impact is over-reliance: a calibrated score can still be wrong, especially out of distribution (e.g., MSMARCO in Table 1), so it should not be treated as a guarantee in high-stakes settings.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work uses publicly available datasets and pre-trained models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the source of every asset. Datasets: HotpotQA (CC BY-SA 4.0), TriviaQA (Apache 2.0), MS MARCO (non-commercial research terms), Natural Questions/NQ-Open (CC BY-SA 3.0), MuSiQue (CC BY 4.0), and 2WikiMultiHopQA (Apache 2.0). Models: Llama-3.1-8B-Instruct (Llama 3.1 Community License), and Qwen2.5-7B-Instruct, Qwen3-14B, Qwen3-32B, and Mistral-7B-Instruct-v0.3 (Apache 2.0). GPT-4.1 nano is used

under the OpenAI API terms. Baseline code: RLCR (MIT) and Rewarding Doubt (public repository, no license specified). Libraries: TRL, Unsloth, and vLLM (Apache 2.0).

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release our code under the MIT license at `https://github.com/Rose-STL-Lab/Semantic_Calibration`, with a README documenting installation, training, evaluation, and the configurations used in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: An LLM (GPT-4.1 nano) is a core component of the method: it judges the correctness of each rollout and the semantic equivalence of rollout pairs, both in the training reward and in the confidence estimate (Section 5.1, Appendices B.3 and B.4).

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.