Systematic Evaluation of Uncertainty Estimation Methods in Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) produce outputs with varying levels of uncertainty, and, just as often, varying levels of correctness; making their practical reliability far from guaranteed. To quantify this uncertainty, we systematically evaluate four approaches for confidence estimation in LLM outputs: **VCE**, **MSP**, **Sample Consistency**, and **CoCoA** (Vashurin et al., 2025).

For the evaluation of the approaches, we conduct experiments on four question-answering tasks using a state-of-the-art open-source LLM. Our results show that each uncertainty metric captures a different facet of model confidence and that the hybrid CoCoA approach yields the best reliability overall, improving both calibration and discrimination of correct answers. We discuss the trade-offs of each method and provide recommendations for selecting uncertainty measures in LLM applications.

1 Introduction

Large language models (LLMs) and their outputs can be difficult to trust: models may produce incorrect answers with unwarranted confidence or hedge on easy questions. This unreliability is especially problematic in high-stakes applications where undetected errors can have serious consequences. One way to mitigate this risk is to equip LLMs with *uncertainty estimation* mechanisms that quantify the model's confidence in its own outputs. Accurate confidence estimates enable downstream systems to calibrate predictions, abstain from low-confidence outputs, or request human intervention when needed (Ji et al., 2023).

Since uncertainty can arise from different sources that affect model behavior, prior work distinguishes between *aleatoric* uncertainty, arising from inherent data noise, and *epistemic* uncertainty, reflecting limited model knowledge (Der Kiureghian and Ditlevsen, 2009). Estimating these uncertainties in LLMs is challenging because their predictions are text tokens. Recent studies propose proxy metrics, including model probability distributions (Hendrycks and Gimpel, 2017; Jiang et al., 2021b; Chen et al., 2023), elicited self-reports (Kadavath et al., 2022; Xiong et al., 2023), and multi-sample consistency signals (Wang et al., 2022; Lyu et al., 2024; Wang et al., 2024). Each has advantages and limitations, and it remains unclear which methods are most effective across tasks.

We compare **Verbalized Confidence Elicitation** (VCE) asks the model to state its own confidence; **Maximum Sequence Probability** (MSP) derives confidence from the probability assigned to the output; **Sample Consistency** measures uncertainty through the stability of answers across multiple sampled outputs, and **Confidence-Consistency Aggregation** (CoCoA) combines model-internal confidence with output consistency.

We make four main contributions: (1) we provide a unified implementation and benchmarking of four uncertainty estimation methods for LLMs, including a hybrid that fuses confidence and consistency (Vashurin et al., 2025); (2) we introduce compact calibration via confidence-like scores; (3) we develop a multi-dataset evaluation of calibration and discrimination; and (4) we provide practical guidance for deploying uncertainty measures.

2 Related Work

Calibration and confidence for NLP/LLMs. Modern neural networks, including transformers, are often miscalibrated, yielding overconfident predictions even when incorrect (Guo et al., 2017; Kumar et al., 2019). This miscalibration undermines reliability in downstream applications, since confidence scores no longer reflect true likelihoods of correctness (Hendrycks and Gimpel, 2017; Lakshminarayanan et al., 2017). Post-hoc methods can partially mitigate these issues (Desai and Dur-

rett, 2020). For generative models, token or sequence likelihoods can be informative, since higher probabilities often signal model confidence while lower probabilities reflect ambiguity, though these measures remain sensitive to length and decoding choices (Ott et al., 2018). LLMs can self-assess to some degree, for example by assigning confidence scores to their own answers or expressing subjective probabilities in natural language (Kadavath et al., 2022), though raw self-reports tend to be overconfident (Xiong et al., 2023).

Consistency and multi-sample reasoning. Consistency-based approaches, including self-consistency via multiple sampled chains, have shown promising results for detecting uncertainty and hallucinations in black-box LLMs (Wang et al., 2023; Manakul et al., 2023). Recent hybrid methods, such as the CoCoA approach, combine confidence from likelihoods with agreement signals, improving calibration and discrimination (Vashurin et al., 2025).

Selective prediction and abstention. Rather than forcing a prediction for every case, models can be trained to respond only when confident and abstain otherwise. This lowers coverage but improves the reliability of the outputs, making real-world deployment more trustworthy (Geifman and El-Yaniv, 2017, 2019).

3 Uncertainty Estimation Methods

We consider four approaches for estimating the uncertainty of an LLM's output on a given query. Each produces a confidence or uncertainty score that can be compared against the correctness of the model's answer.

3.1 Verbalized Confidence Elicitation (VCE)

A simple approach is to ask the model for its own confidence: Verbalized Confidence Elicitation (VCE) prompts it to provide both an answer and a self-reported confidence score (Xiong et al., 2023). For example, the model might be prompted with: *"Answer the question and provide your confidence (0–100)."* A possible response could be: *"Answer: 42. Confidence: 80."* This approach is model-agnostic (requiring no access to model internals) and easy to implement.

Prior work suggests that LLMs can sometimes provide reasonably calibrated self-evaluations when prompted appropriately (Kadavath et al., 2022; Geng et al., 2023). However, in practice, we find that verbalized confidence tends to be systematically overestimated, particularly for instruction-tuned models that are trained to sound assured (Wang et al., 2024; Kalai et al., 2025). Models often express high confidence even when they are incorrect, limiting the utility of raw VCE as a reliability signal. We also experiment with using *multiple* samples in the VCE setting, inspired by Xiong et al. (2023). Specifically, we obtain M independent answers (via repeated sampling), each with a self-reported confidence C_i. We then compute an *agreement-weighted confidence* that gives higher weight to answers that agree with the majority prediction:

$$\hat{C}_{\text{agree}} = \frac{\sum_{i=1}^{M} \mathbb{1}\{\hat{Y}_i = \tilde{Y}\} \cdot C_i}{\sum_{i=1}^{M} C_i}, \quad (1)$$

where $\tilde{Y}$ is the answer given by the majority of samples and $\mathbb{1}\{\cdot\}$ is an indicator for agreement. This aggregated confidence $\hat{C}_{\text{agree}}$ is used as the final VCE score for that query. Intuitively, it discounts confident outlier answers that disagree with the majority, which helps reduce overconfidence in the combined estimate.

3.2 Maximum Sequence Probability (MSP)

Probability-based metrics derive uncertainty from the model's internal likelihood assigned to its output. A simple and widely used measure is MAXIMUM SEQUENCE PROBABILITY (MSP), analogous to the Maximum Softmax Probability for classifiers (Hendrycks and Gimpel, 2017). Given an input x and an output sequence $y = [y_1, y_2, \ldots, y_T]$ produced by the model, we define the *unnormalized uncertainty* as the negative log-likelihood of the sequence:

$$U_{\text{MSP}}(x, y) = -\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, x). \quad (2)$$

A lower U_{MSP} (higher probability) indicates higher model confidence in that particular output. MSP can be obtained with negligible overhead during generation, since modern decoding APIs often return token probabilities.

Direct values of U_{MSP} are not comparable across different inputs or tasks because they depend on sequence length and other factors (see Appendix Figure A.4). Extending prior work, we apply a normalization to convert U_{MSP} into a bounded *confidence-like score* $C_{\text{MSP}}(x, y) \in [0, 1]$ (Hendrycks and

Gimpel, 2017; Jiang et al., 2021b). We first clip extreme values by truncating U_{MSP} at the 98th percentile over the evaluation set. This mitiagates outliers. Denote this clipped uncertainty as $\tilde{U}$. We then apply min-max normalization

$$C_{\text{MSP}}(x, y) \;=\; 1 \;-\; \frac{\tilde{U}(x, y) - \min(\tilde{U})}{q_{0.98} - \min(\tilde{U})}\,, \quad (3)$$

where $q_{0.98}$ is the 98th percentile of U_{MSP} values (after clipping) and $\min(\tilde{U})$ is the minimum. Thus, C_{MSP} maps the model's likelihood for an output onto a $[0, 1]$ scale, where 1 corresponds to high confidence. Although this normalization makes scores more comparable, MSP-based confidence still suffers from known issues: models tend to assign overly high probability to incorrect answers (Guo et al., 2017; Jiang et al., 2021a), and longer outputs incur lower probabilities by construction (Ott et al., 2018). These limitations motivate augmenting probability-based measures with additional signals.

3.3 Sample Consistency

Another perspective on uncertainty is to check if an LLM produces the same answer when asked the same question multiple times. **Sample consistency** assesses the stability of model outputs under stochastic generation: if the model almost always gives the same answer, it is likely confident; if it frequently changes its answer, it is uncertain. This method captures a form of epistemic uncertainty by probing the model's decision boundary with multiple samples (Lyu et al., 2024).

In our experiments, we generate k answers for each question using either (a) repeated decoding with different random seeds, or (b) requesting multiple outputs in one call (e.g., top-k sampling). We then measure pairwise semantic similarity among the k answers. High agreement indicates higher confidence, whereas diverse or contradictory answers signal uncertainty. We use two complementary similarity measures:

- **Embedding Similarity:** We compute the cosine similarity between each pair of answers using Sentence-BERT embeddings (Reimers and Gurevych, 2019). This captures semantic closeness (e.g., paraphrases).
- **NLI-Based Consistency:** We also evaluate whether one answer entails the other using a RoBERTa-large model fine-tuned on natural language inference (NLI) (Liu et al., 2019; Williams et al., 2018). For each pair, we take the predicted probability of *entailment* as a directional consistency score. This detects contradictions that cosine similarity might miss.

For each question, we aggregate the pairwise similarities across all $M = \binom{k}{2}$ pairs:

$$U_{cons}(x) = \frac{1}{M} \sum_{i<j} sim(x_i, x_j) \quad (4)$$

The average similarity provides a *consistency score* for that question, while the variance or minimum similarity can indicate divergence among outputs. In practice, we found the mean similarity to be a simple and effective proxy for consistency, which we use as the Sample Consistency metric.

Sample consistency directly captures whether the model's knowledge and reasoning are robust for a given query (Wang et al., 2022). It can reveal uncertainty even when probability-based metrics are overconfident, and it does not require ground-truth answers or model introspection. However, it comes at a high computational cost (multiple model runs) and can be fooled if the model consistently repeats the *same wrong answer*. In such cases, consistency alone would be high even though the model is confidently wrong.

3.4 Confidence–Consistency Aggregation (CoCoA)

Finally, we evaluate **CoCoA** (Vashurin et al., 2025), a recently proposed method that combines the model's internal confidence with output consistency. CoCoA is formulated in a minimum Bayes risk framework: it penalizes an answer y^* if (i) the model assigns low probability to y^* (information-based uncertainty u), and (ii) y^* is semantically dissimilar to other likely answers (consistency-based uncertainty U_{cons}). In our implementation we instantiate the first term with sequence likelihood, $u(y^\star \mid x) = -\log P(y^\star \mid x)$, using the same token log-probabilities as MSP (Sec. 3.2). *While u could also be instantiated with VCE, we prefer MSP because it is prompt-agnostic, numerically stable across tasks, and less overconfident in our setting (Sec. 5), which leads to more reliable fusion with the consistency term.* For the consistency term, we generate M alternative answers $\{y^{(i)}\}_{i=1}^{M}$ via stochastic decoding and compute their semantic similarity to y^* using the same NLI-based model as for Sample Consistency (see Chapter 3.3). Let

$s(y^*, y') \in [0, 1]$ be the similarity (entailment probability) between y^* and another output y'. The average dissimilarity is:

$$U_{\text{cons}}(y^*|x) \;=\; \frac{1}{M}\sum_{i=1}^{M}\big(1 - s(y^*,\, y^{(i)})\big)\,. \quad (5)$$

CoCoA combines the two components multiplicatively:

$$\hat{U}_{\text{CoCoA}}(y^*|x) \;=\; u(y^*|x) \;\cdot\; U_{\text{cons}}(y^*|x)\,, \quad (6)$$

such that high uncertainty is only assigned when the model is both internally unsure and externally inconsistent with itself. The raw $\hat{U}_{\text{CoCoA}}$ values are then mapped to a *confidence-like score* C_{CoCoA} in $[0, 1]$ via the same clipping and normalization procedure used for MSP (applying it to $\hat{U}_{\text{CoCoA}}$ across the dataset, see Chapter 3.2). Thus, CoCoA produces a final confidence-like score $C_{CoCoA} = 1 - \tilde{U}_{\text{CoCoA,norm}}$ for each answer.

CoCoA effectively fuses two signals: the model's own confidence in an answer and the semantic relationship between multiple outputs generated using repeated sampling Prior evaluations have shown that CoCoA can improve correlation with correctness over using either confidence or consistency alone (Vashurin et al., 2025). The main drawback is the computational overhead of generating multiple samples for each query. In settings where latency is critical, CoCoA might be too expensive, but it represents a strong upper bound on what post-hoc uncertainty quantification can achieve by leveraging all available signals. *For high-risk deployments, a more conservative approach to fusion is also possible, such as an OR-style rule* $U_{\text{OR}}(y^\star \mid x) = \max\{u(y^\star \mid x), U_{\text{cons}}(y^\star \mid x)\}$. *This tends to reduce coverage, but also lowers the likelihood of accepting uncertain answers.*

Resources and tooling. We use Sentence-BERT (Reimers and Gurevych, 2019) and RoBERTa (Liu et al., 2019) with MultiNLI (Williams et al., 2018), and build on HuggingFace Transformers (Wolf et al., 2020). For the evaluation, we use MSP and CoCoA (see Section 3.2 and 3.4, respectively) using the `lm-polygraph` package (IINemo, 2025). Tasks include BoolQ (Clark et al., 2019), SQuAD 2.0 (Rajpurkar et al., 2018), TriviaQA (Joshi et al., 2017), and GSM8K (Cobbe et al., 2021). Base models follow the Llama 3 family (Meta AI, 2024).

4 Experimental Setup

4.1 Tasks and Model

Our goal is to stress-test uncertainty estimators across qualitatively different cases: (i) *binary* decisions with short answers, where overconfidence is common; (ii) *extractive* reading comprehension tasks with long context passages; (iii) *open-domain* factoid QA with knowledge gaps and entity confusion; and (iv) *multi-step reasoning* with arithmetic and intermediate steps. To that end, we evaluate on four standard QA benchmarks:

BoolQ (Clark et al., 2019) is a yes/no dataset derived from natural queries. Answers are single tokens (YES/NO), providing a straightforward testbed for calibration, since probabilities directly correspond to the correctness of yes/no decisions. Ambiguity in question phrasing induces aleatoric uncertainty, while topical drift can elicit epistemic uncertainty.

SQuAD 2.0 (Rajpurkar et al., 2018) augments extractive QA with unanswerable questions. Models must either return a short span from the supplied passage or abstain, making it ideal for studying selective prediction and calibration.

TriviaQA (Joshi et al., 2017) is an open-domain factoid QA. Evidence passages are not guaranteed to contain a succinct rationale, and entity-heavy questions surface knowledge gaps and plausible-but-wrong confusions. This setting is known to benefit ranking-style confidence (e.g., likelihood-based scores) while remaining challenging to calibrate.

GSM8K (Cobbe et al., 2021) contains grade-school math word problems that require multi-step reasoning and arithmetic. Even small reasoning slips lead to confidently wrong final numbers, which stresses both discrimination (AUROC) and calibration. We ask the model for a short final answer; intermediate steps may be produced but are not required.

Model. We use **LLaMA 3.2 (3B-Instruct)** (Meta AI, 2025, 2024) for three reasons. First, it is *open* and exposes token log-probabilities, enabling MSP without internal modifications and making our pipeline reproducible. Second, the *3B* size offers a strong accuracy and compute trade-off: it is capable enough to make the uncertainty problem non-trivial while being small enough that we can

afford repeated sampling (SEP) at scale. Third, the instruction-tuned variant produces structured self-reports for VCE and handles long contexts (e.g., SQuAD passages) with its large context window. While larger LLaMA variants may yield higher raw accuracy, our focus is on the *relative behavior* of uncertainty estimators; we expect the qualitative trends we observe (e.g., VCE overconfidence, gains from consistency, CoCoA's calibration) to transfer across model sizes.

4.2 Decoding & multi-sample regimes

We adopt a temperature of $T{=}0.7$ for all uncertainty estimation methods, as it provides a robust trade-off across tasks (see Figs. A.1 and A.2). For *multi-sample* conditions ($M{=}5$ unless noted), we evaluate two regimes *for VCE and Sample Consistency*: **TOP-K** (multi-output from one decoding call) and **SEP** (five separate decodings with independent seeds). For *CoCoA* we **always use SEP** to obtain $M{=}10$ independent samples; TOP-K is not used.
The resulting M answers are aggregated per method (agreement-weighted confidence for VCE, pairwise similarity for consistency, and the combined confidence-consistency score for CoCoA).

Effective sample sizes (N) and filtering. Unless noted otherwise, we target $N{=}1000$ examples per dataset for *single-sample VCE* and $N{=}500$ for all other methods (including multi-sample VCE, Sample Consistency, MSP, and CoCoA). The realized N can be slightly smaller due to automatic filtering of unparseable model outputs (e.g., missing or non-numeric VCE self-reports, malformed JSON), empty/overlong generations, or dataset-specific preprocessing. For SQuAD 2.0, we remove unanswerable questions to align with our evaluation prompts, yielding $N{=}638$ for the single-sample condition and $N{=}310$ in the multi-sample setting.

4.3 Evaluation Metrics

To quantify the effectiveness of each uncertainty estimation method, we consider two primary evaluation criteria: **Calibration** and **Discrimination**. Calibration measures how well the confidence scores align with actual correctness frequencies, while discrimination measures how well the scores separate correct vs incorrect outputs.

Expected Calibration Error (ECE). We compute the expected calibration error (ECE) (Naeini et al., 2015) of each method's confidence scores. ECE partitions the predictions into bins by confidence and compares the average confidence to the empirical accuracy in each bin. If B_b is the set of indices of predictions falling into bin b, the ECE is:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|B_b|}{n} \left|\mathrm{acc}(B_b) - \mathrm{conf}(B_b)\right|, \qquad (7)$$

where $\mathrm{acc}(B_b)$ is the fraction of correct answers in bin b and $\mathrm{conf}(B_b)$ is the average predicted confidence in that bin, and n is the total number of predictions. Lower ECE (closer to 0) indicates better calibration (the confidence scores reflect true probabilities of correctness). We use $B = 10$ bins of equal width over $[0, 1]$.

AUROC. To evaluate discrimination, we use the Area Under the Receiver Operating Characteristic curve (AUROC) (Fawcett, 2006). This treats confidence-based ranking as a binary classification of correct vs. incorrect answers. The ROC curve plots true positive rate vs false positive rate as we vary a confidence threshold τ. The AUROC is the probability that a randomly chosen correct answer is assigned a higher confidence than a randomly chosen incorrect answer. An AUROC of 1.0 indicates perfect rank ordering by confidence, whereas 0.5 is no better than chance. We calculate AUROC for each method by comparing its confidence scores against the ground-truth correctness of the model's answers.

Additionally, we report the **accuracy** (fraction of correct answers) of the model on each dataset to contextualize the difficulty and the headroom for uncertainty metrics (which cannot exceed the model's own accuracy). Note that methods like CoCoA and MSP do not change which answer is given (they only score the given answer), so accuracy remains the same across those methods.

5 Results and Analysis

Table 1 summarizes performance across datasets. In addition to **Accuracy**, **ECE**, and **AUROC**, we report **Average Confidence** (in %) to expose over/under-confidence directly. *Sample sizes* (N, rightmost column) follow the filtering rules in Section 4.2: minor deviations occur for VCE due to the automatic removal of unparseable generations. For SQuAD 2.0, unanswerable questions were removed, yielding $N{=}638$ in the single-sample case and $N{=}310$ in the multi-sample setting.

Dataset	Method	Accuracy ↑	Avg. Confidence (%)	ECE ↓	AUROC ↑	N
BoolQ	VCE: Single Sample	0.780	90.371	0.183	0.646	973
	VCE: Multi-Sample (TOP-K)	0.768	88.925	**0.146**	0.646	500
	VCE: Multi-Sample (SEP)	0.768	91.107	0.168	0.636	500
	MSP	**0.790**	58.714	0.203	0.673	500
	CoCoA	**0.790**	65.219	0.160	**0.687**	500
SQuAD	VCE: Single Sample	0.868	97.633	0.124	0.573	638
	VCE: Multi-Sample (TOP-K)	0.829	77.338	0.118	0.722	310
	VCE: Multi-Sample (SEP)	0.848	77.372	0.113	0.746	310
	Sample Cons. (TOP-K)	0.865	61.616	0.186	0.739	310
	Sample Consistency (SEP)	0.832	61.506	0.149	0.758	310
	MSP	**0.890**	78.023	0.122	0.836	310
	CoCoA	**0.890**	83.229	**0.062**	**0.844**	310
TriviaQA	VCE: Single Sample	0.501	68.123	0.309	0.700	994
	VCE: Multi-Sample (TOP-K)	0.511	70.564	0.216	0.645	489
	VCE: Multi-Sample (SEP)	0.496	71.456	0.233	0.690	500
	Sample Cons. (TOP-K)	0.508	37.491	0.162	0.793	500
	Sample Consistency (SEP)	0.510	38.122	**0.134**	0.793	500
	MSP	**0.610**	65.186	0.147	**0.841**	500
	CoCoA	0.608	68.662	0.167	**0.841**	500
GSM8K	VCE: Single Sample	0.651	98.800	0.335	0.548	988
	VCE: Multi-Sample (TOP-K)	0.749	46.800	0.129	0.775	462
	VCE: Multi-Sample (SEP)	**0.781**	47.000	0.145	0.736	468
	MSP	0.690	64.557	0.104	**0.838**	500
	CoCoA	0.680	65.219	**0.081**	0.786	500

Table 1: Accuracy, calibration (ECE), and discrimination (AUROC) on each dataset, plus average confidence. Multi-sample *VCE* and Sample Consistency uses $M{=}5$ with either TOP-K (single decoding with multiple returns) or SEP (five separate decodings). CoCoA use $M{=}10$ via SEP only. Bold = best (lowest ECE or highest AUROC) per dataset.

High-level takeaways. (1) **VCE overconfidence is pronounced.** Single-sample VCE reports extremely high mean confidence (e.g., 97.6% on SQuAD, 98.8% on GSM8K) despite much lower accuracy, yielding large ECE. (2) **Multi-sample aggregation moderates VCE.** Aggregating self-reports reduces average confidence by 20–50pp and improves calibration/AUROC (e.g., BoolQ ECE $0.183 \rightarrow 0.146$; SQuAD AUROC $0.573 \rightarrow 0.722$–0.746). (3) **Hybridization pays off.** CoCoA is best-calibrated (SQuAD 0.062, GSM8K 0.081) with competitive AUROC (SQuAD 0.844; tie on TriviaQA).

Dataset-level analysis (task difficulty matters). *BoolQ* (binary yes/no) is comparatively easy; all methods achieve similar accuracy (≈ 0.79), and CoCoA provides the best ranking (AUROC 0.687) while VCE(TOP-K) attains the lowest ECE (0.146). *SQuAD 2.0* (extractive QA with unanswerables) benefits from combining confidence and consistency: CoCoA achieves both the lowest ECE and highest AUROC at top accuracy (0.890). *TriviaQA* (open-domain factoid) stresses retrieval/knowledge; MSP ranks best (0.841), while *consistency (SEP)* calibrates best (0.134). *GSM8K* (multi-step math reasoning) is the hardest: self-consistency boosts accuracy (VCE-SEP **0.781**); MSP remains the best ranker (0.838); CoCoA is clearly the most calibrated (0.081).

TOP-K vs. SEP. Across datasets, TOP-K and SEP are *largely comparable*. For VCE multi-sample, the mean absolute difference between TOP-K and SEP is ≈ 0.03 in AUROC and ≈ 0.015 in ECE (e.g., BoolQ AUROC 0.646 vs. 0.636; SQuAD 0.722 vs. 0.746; TriviaQA 0.645 vs. 0.690; GSM8K 0.775 vs. 0.736). Accuracy differences average ≈ 0.02 (e.g., GSM8K 0.749 vs. 0.781). Sample Consistency with separated decoding (SEP) achieves slightly higher AUROC values than Top-K on SQuAD (0.758 vs. 0.739) and the same in TriviaQA (0.794). Top-K consistently achieves lower ECE values(0.186 vs. 0.149 on SQuAD; 0.162 vs. 0.134 on TriviaQA), suggesting that it provides better calibrated confidence scores. This near parity yields a practical recommendation: **when compute/latency matters, prefer TOP-K** (single decoding, multiple returns), using SEP only when small diversity gains are needed. (For CoCoA

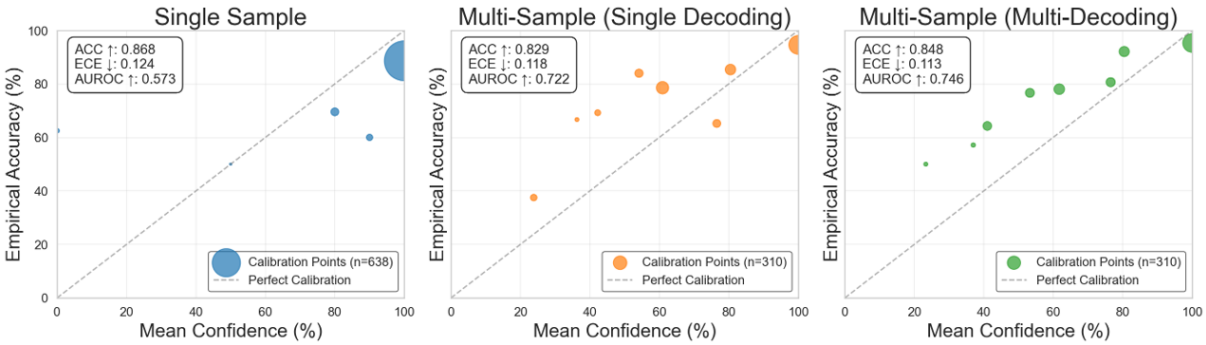


Figure 1: **VCE calibration on SQuAD 2.0 across decoding methods.** Reliability plots (empirical accuracy vs. mean confidence) for single-sample, multi-sample with single decoding (TOP-K), and multi-sample with multi-decoding (SEP). Bubble area is proportional to bin count; the dashed line denotes perfect calibration $y{=}x$. Multi-decoding tightens high-confidence bins, reduces ECE, and improves AUROC while keeping accuracy comparable.

we use SEP exclusively.)

As suggested by Figure 1, the dominant failure mode for single-sample VCE is systematic overconfidence, which multi-sample aggregation largely mitigates. Calibration plots for all datasets are provided in Appendix Fig. A.3.

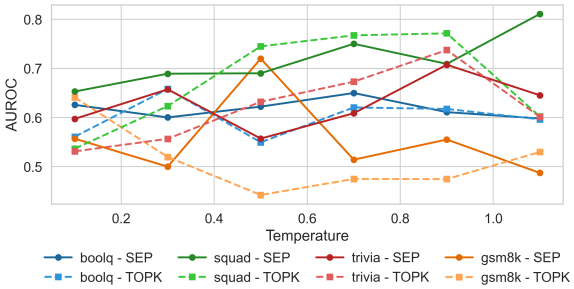


Figure 2: **Temperature effects on ranking quality.** AUROC vs. decoding temperature for multi-sample VCE under SEP (solid) and TOP-K (dashed). Optima differ by dataset, with most gains occurring at mid-range T (0.6–0.8).

Temperature sweeps: no universal best. We conducted comprehensive temperature sweeps evaluating AUROC, ECE, and overconfidence across all tasks. Figure 2 shows one representative result: AUROC vs. temperature under multi-sample VCE. This figure illustrates that temperature effects are *dataset-specific*. AUROC typically increases from very low T (0.1–0.3) toward mid-range values ($\approx$ 0.7), though the precise optimum varies across datasets. At higher temperatures, calibration error and overconfidence often plateau or worsen. *There is no single temperature that is uniformly optimal across tasks and evaluation metrics*; T must be adjusted for each dataset and objective. The full single-sample and multi-sample sweeps, including accuracy, calibration error, and bias, are provided in Appendix Figures A.1 and A.2

MSP. MSP achieves strong discrimination with the highest AUROC values on TriviaQA (0.841) and GSM8K (0.838), making it a dependable ranker. However, calibration is slightly weaker in some cases. Since ECE is typically higher than CoCoA's, the probability scores are less aligned with true correctness (see Table 1). See Appendix Fig. A.5 for MSP reliability diagrams across all datasets. MSP is therefore well-suited for reliable ranking, but less effective when well-calibrated confidence scores are required.

Sample Consistency. Sample Consistency, especially with separated decoding, offers low calibration error in several cases, such as TriviaQA ($ECE = 0.134$), highlighting its advantage for confidence calibration. At the same time, its discrimination (AUROC) is generally lower across datasets compared to MSP, with the strongest values around 0.79. Overall, Sample Consistency offers a favorable calibration–discrimination trade-off.

Dataset	Overall ACC	Filtered ACC	Improvement (pp)	Remaining Samples
BoolQ	79.00%	90.85%	+11.85	164 / 500 (32.8%)
SQuAD	89.00%	95.98%	+6.98	224 / 310 (72.3%)
TriviaQA	60.80%	87.12%	+26.32	264 / 500 (52.8%)
GSM8K	68.00%	89.18%	+21.18	194 / 500 (38.8%)

Table 2: Selective prediction with CoCoA ($C_{CoCoA} > 0.8$). Accuracy improves substantially while keeping a large fraction of examples.

Selective prediction with CoCoA. A selective-prediction rule that accepts only high-confidence outputs can materially improve reliability. Table 2

reports accuracy when **filtering by CoCoA confidence** $C_{CoCoA} > 0.8$ (absolute percentage-point gains and retained coverage shown). In short, *precision rises sharply while retaining useful coverage*: gains are largest on harder tasks (TriviaQA, GSM8K), consistent with Figure 3 illustrates CoCoA on TriviaQA, where high-confidence bins closely match ground-truth accuracy. See Appendix Fig. A.6 for all CoCoA calibration plots on all datasets.

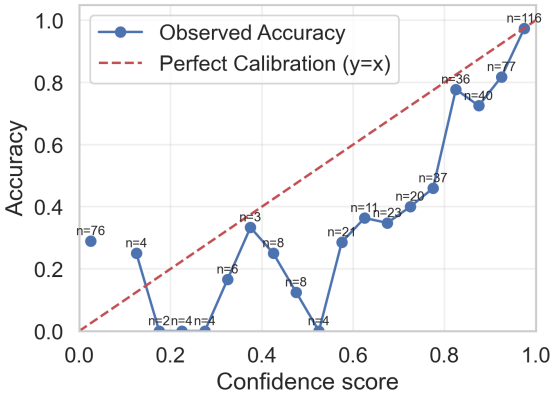


Figure 3: **Calibration of CoCoA on TriviaQA.** Bin-wise accuracy increases monotonically with confidence and aligns closely with the $y{=}x$ diagonal at high confidence, indicating well-calibrated scores in the decision-relevant range (bin sizes annotated).

6 Conclusion

Task structure strongly shapes which uncertainty estimator is most useful. On the simplest setting-binary decisions in BoolQ-methods are closer together: MSP and CoCoA tie on accuracy while CoCoA provides the best ranking signal, and multi-sample VCE mostly reduces modest overconfidence. In contrast, on the most complex setting-multi, step reasoning in GSM8K, single-sample verbalized confidence is highly overconfident and weak at discrimination, while multi-sample aggregation markedly improves accuracy. The hybrid CoCoA yields the most calibrated scores (ECE 0.081) even when MSP remains the strongest ranker. TriviaQA underscores the value of probability-based ranking under knowledge uncertainty. Importantly, for VCE we observe that TOP-K and SEP produce very similar reliability (average gaps ~ 0.03 AUROC and ~ 0.015 ECE), enabling practitioners to reuse alternatives from a single decoding to approximate separated decoding's benefits and thereby save compute and latency. We reserve separated decoding for cases where small diversity gains matter. Temperature sweeps further show no universal best setting-optimal T is task-specific-so moderate temperatures can help discrimination but do not by themselves fix miscalibration.

Overall, our results suggest a practical recipe. MSP works best when ranking is most important, especially on knowledge-heavy tasks. Self-consistency helps to stabilize performance on difficult reasoning problems. CoCoA is preferable when downstream applications require calibrated and reliable confidence. When using verbalized confidence, TOP-K offers a good trade-off by reducing the cost of multi-sample decoding.

Limitations

Our study focuses on question-answering tasks with relatively short outputs. The performance of uncertainty metrics may vary on other tasks such as long-form generation, dialogue, or code generation. Future work should evaluate whether the trends observed here hold for tasks with longer, more complex outputs. Additionally, our experiments used a single model (LLaMA 3.2 3B-Instruct). Larger models or different architectures might have different calibration characteristics. However, we expect the qualitative insights to generalize. Another limitation is the computational cost of multi-sample methods. Using TOP-K for VCE mitigates this, but Sample Consistency and CoCoA still require separated decodings. Finally, our evaluation assumes access to ground-truth correctness to measure calibration and AUROC.

Acknowledgment on AI Assistance. DeepL and ChatGPT were used for translation, summarization, and grammatical or stylistic rephrasing of text. GitHub Copilot was used to improve code efficiency and assist in generating auxiliary code components.

References

Xinyun Chen, Yang Liu, and Xipeng Qiu. 2023. Calibrating large language models via temperature scaling. *arXiv preprint arXiv:2306.07900*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proc. of NAACL-HLT*, pages 2924–2936.

Karl Cobbe, Vivek Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Łukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *EMNLP*.

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

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *NeurIPS*.

Yonatan Geifman and Ran El-Yaniv. 2019. Deep neural networks with a reject option. In *NeurIPS*.

Jinxin Geng, Fangyu Cai, Yiren Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2023. A survey of confidence estimation and calibration in large language models. *arXiv preprint arXiv:2311.08298*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proc. of ICML*, pages 1321–1330.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proc. of ICLR*.

IINemo. 2025. LM-Polygraph: Benchmarking uncertainty quantification methods for llms. https://github.com/IINemo/lm-polygraph. Retrieved September 7, 2025.

Zhengyan Ji, Nayeon Lee, Rita Frieske, Tongshuang Yu, Daniel Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Hein Jiang, Been Kim, Muyuan Guan, and Maya R. Gupta. 2021a. To trust or not to trust a classifier. *Advances in Neural Information Processing Systems (NeurIPS)*, 34.

Zhengbao Jiang, He He, Yejin Shao, Kai-Wei Chang, and Tania Wang. 2021b. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the ACL*, 9:1167–1181.

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

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Neal DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Adam T. Kalai, Ofir Nachum, Santosh S. Vempala, and Eric Zhang. 2025. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Qing Lyu, Yu Meng, Xiang Ren, and Nanyun Peng. 2024. Calibrating large language models with sample consistency. *arXiv preprint arXiv:2402.13904*.

Potsawee Manakul, Faisal Ladhak, Jeffrey Flanigan, and Yonatan Belinkov. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv:2303.08896*.

Meta AI. 2024. Llama 3: Open and efficient foundation language models. *arXiv:2407.21783*.

Meta AI. 2025. LLaMA 3.2 model card. Accessed June 10, 2025.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*.

Myle Ott, Michael Auli, David Grangier, and Marc Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *Proc. of ICML*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proc. of EMNLP*.

Ruslan Vashurin, Mikhail Goloburda, Anastasia Ilina, Andrei Rubashevskii, Preslav Nakov, Artem Shelmanov, and Maxim Panov. 2025. Uncertainty quantification for LLMs through minimum Bayes risk: Bridging confidence and consistency. *arXiv preprint arXiv:2502.04964*.

Pu Wang, Yongchao Wang, Ming Diao, Kaize He, Gongshen Dong, and Wei Xu. 2024. Multi-perspective consistency enhances confidence estimation in large language models. *arXiv preprint arXiv:2402.11279*.

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

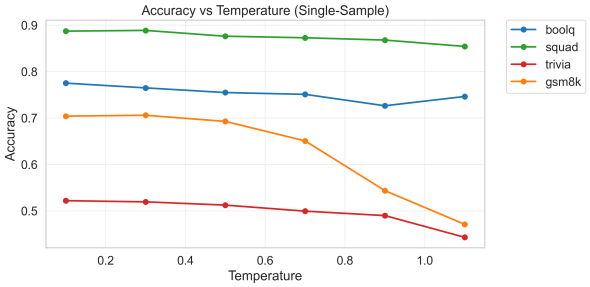
Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *arXiv:2203.11171*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. of NAACL-HLT*.

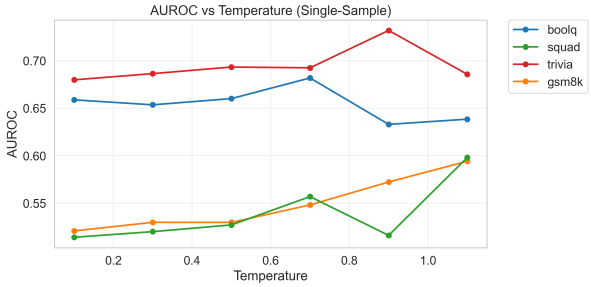
Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clément Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. Transformers: State-of-the-art natural language processing. In *Proc. of EMNLP (Demos)*, pages 38–45.

Machel Xiong, Zhuang Hu, Xingyu Lu, Yuxiang Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. *arXiv preprint arXiv:2306.13063*.

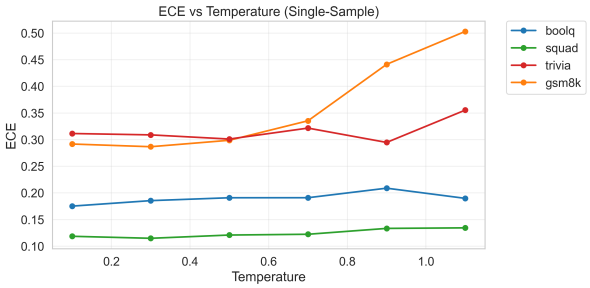
A Additional Results and Figures



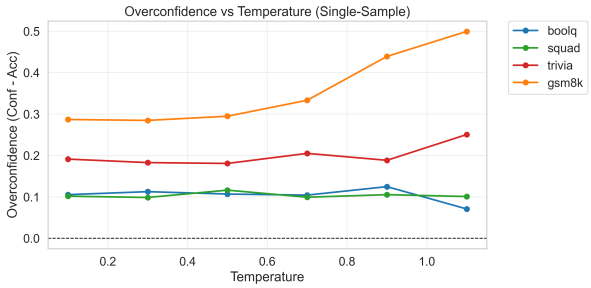
(a) Accuracy vs. temperature (single-sample VCE).



(b) AUROC vs. temperature (single-sample VCE).

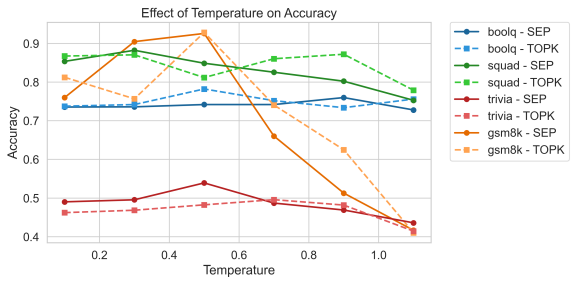


(c) ECE vs. temperature (single-sample VCE).

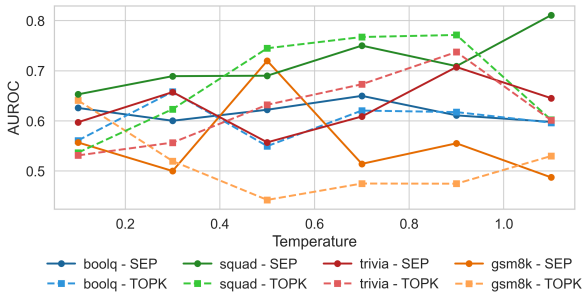


(d) Overconfidence vs. temperature (single-sample VCE).

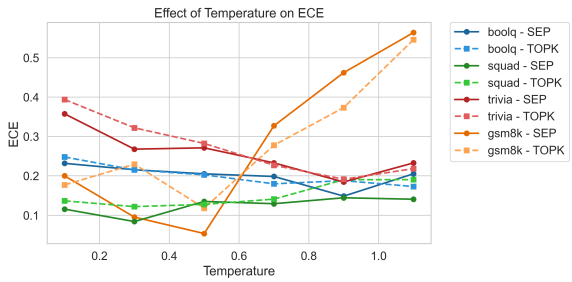
Figure A.1: Single-sample VCE temperature sweep. Optima are task-specific; no single T dominates across datasets.



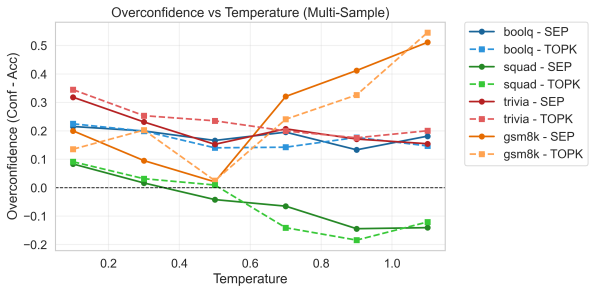
(a) Accuracy vs. temperature (multi-sample VCE, SEP vs. TOP-K).



(b) AUROC vs. temperature (multi-sample VCE, SEP vs. TOP-K).

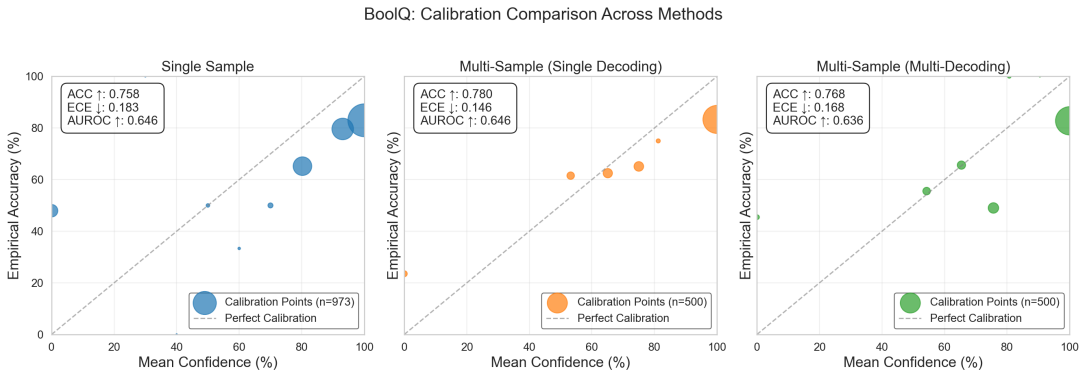


(c) ECE vs. temperature (multi-sample VCE, SEP vs. TOP-K).

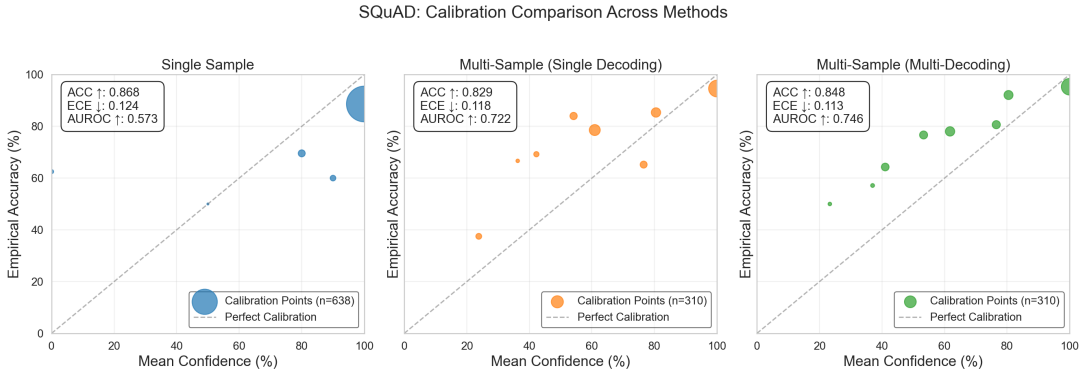


(d) Overconfidence vs. temperature (multi-sample VCE, SEP vs. TOP-K).

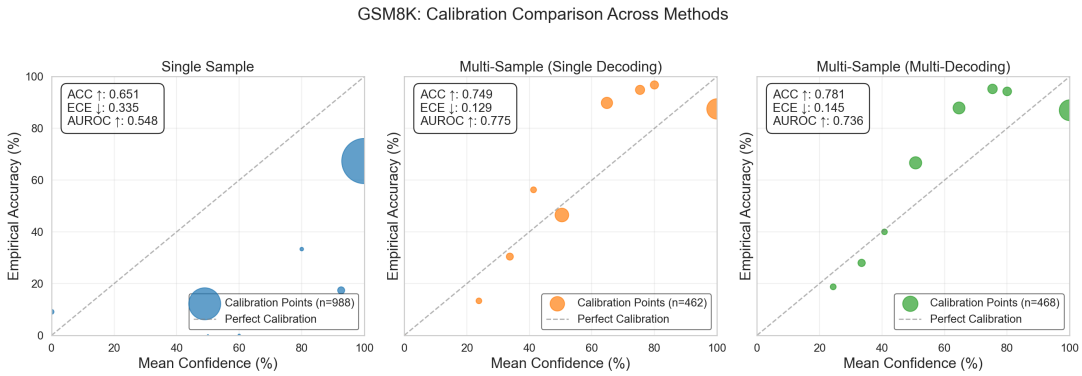
Figure A.2: Multi-sample VCE temperature sweep. Optima are task-specific; no single T dominates across datasets; SEP and TOP-K are broadly comparable.



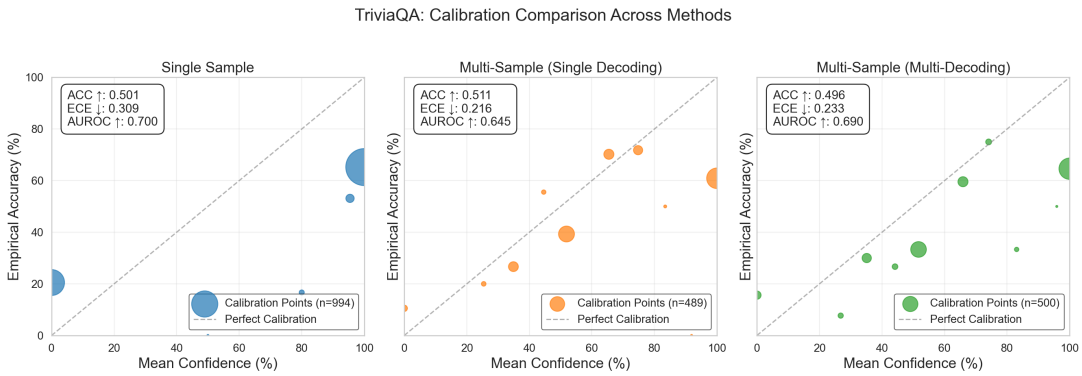
(a) **BoolQ.** Multi-sample corrects overconfident high bins; AUROC is stable.



(b) **SQuAD 2.0.** AUROC improves strongly; ECE reduces to $\mathbf{0.113}$ (SEP).



(c) **GSM8K.** Largest relative ECE drop; ACC and AUROC improve substantially.



(d) **TriviaQA (open-domain).** Large ECE drop; ACC comparable and AUROC slightly lower than single-sample.

Figure A.3: **VCE: calibration by decoding method.** Points are confidence bins (bubble size = bin count); dashed line is perfect calibration $y{=}x$.

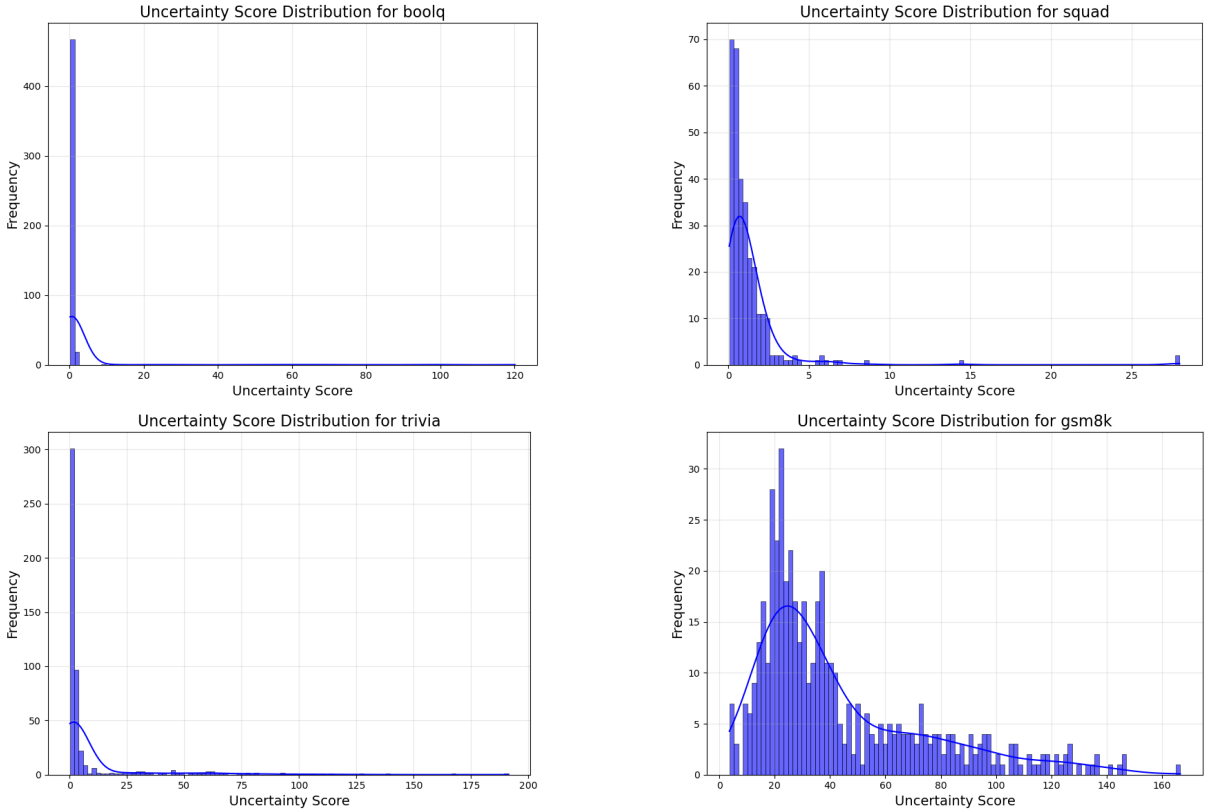
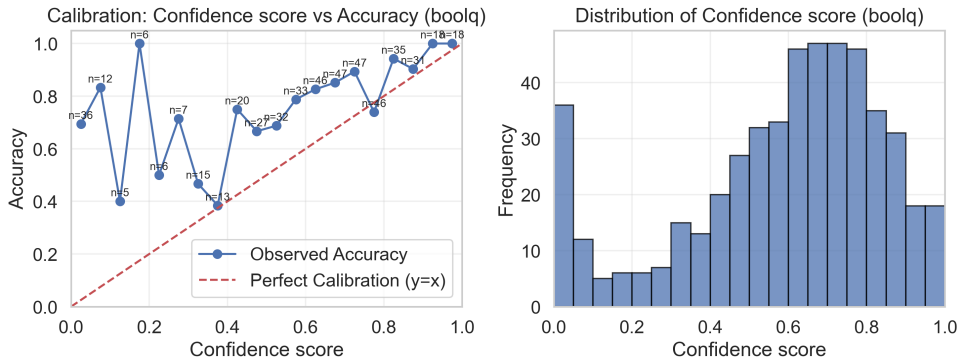
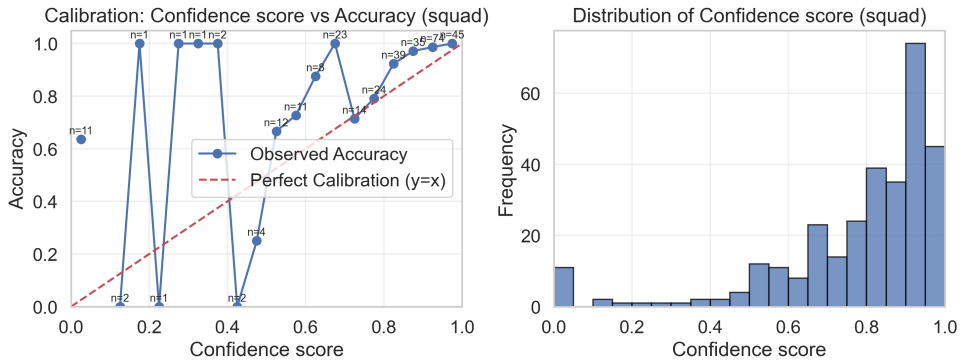


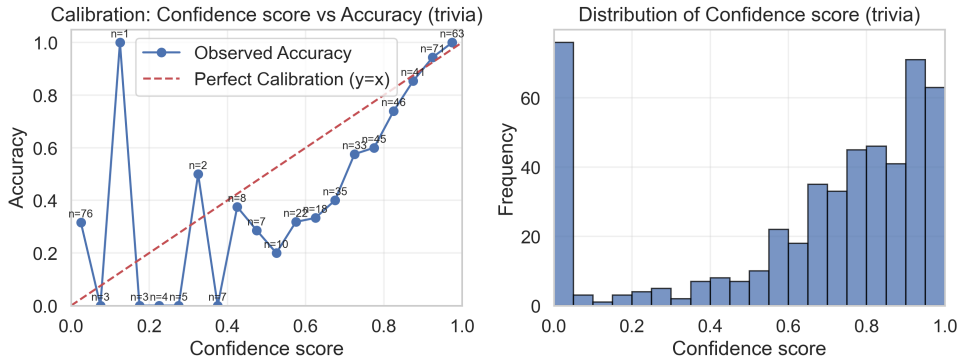
Figure A.4: Distribution of raw MSP uncertainty scores



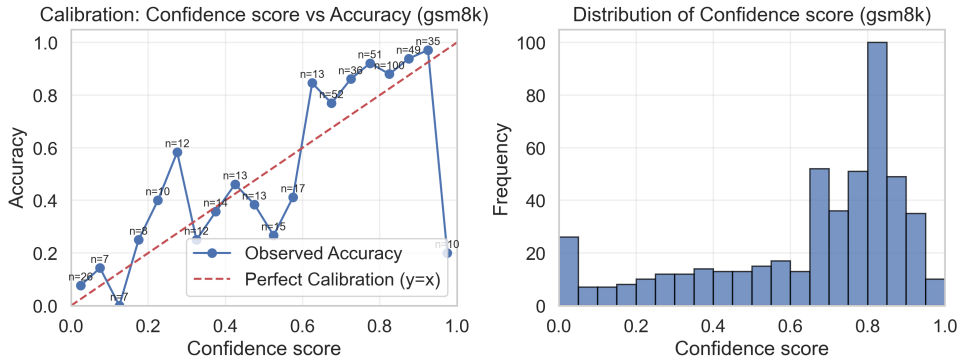
(a) BoolQ



(b) SQuAD 2.0

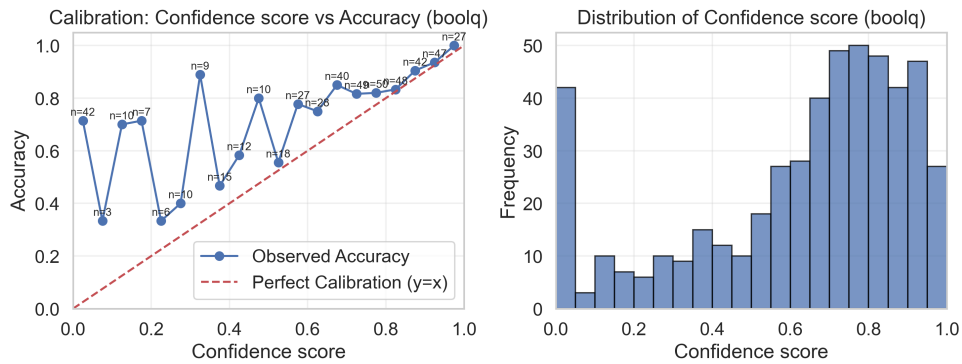


(c) TriviaQA

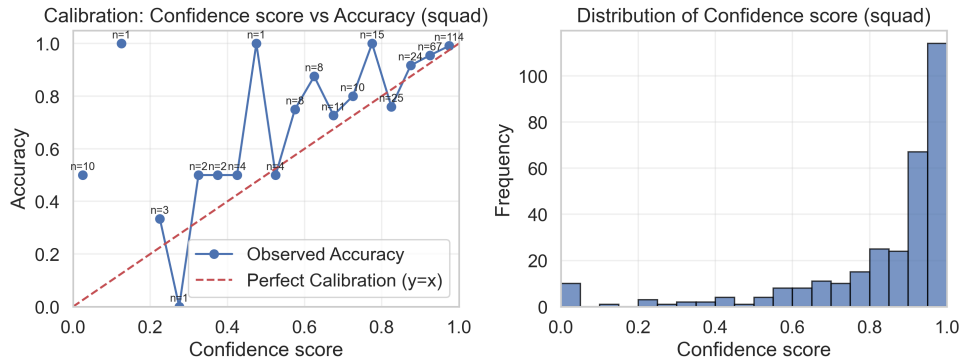


(d) GSM8K

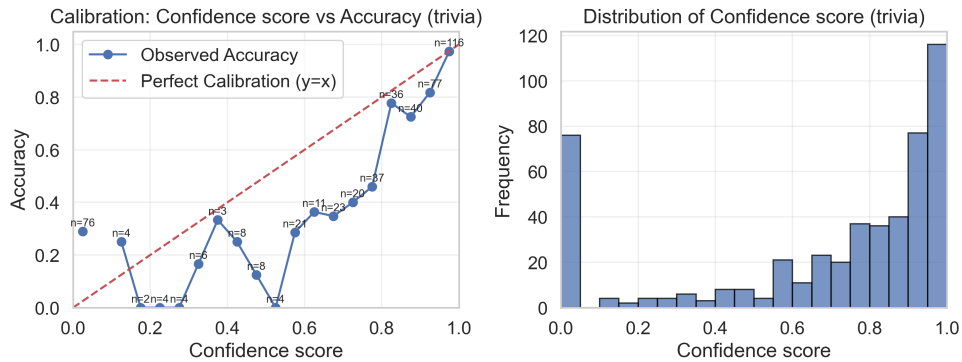
Figure A.5: **MSP calibration with C_{MSP}.** Reliability (left) and confidence distribution (right). Labels above points denote bin counts. MSP aligns high-confidence bins with $y{=}x$ on SQuAD, TriviaQA and BoolQ.



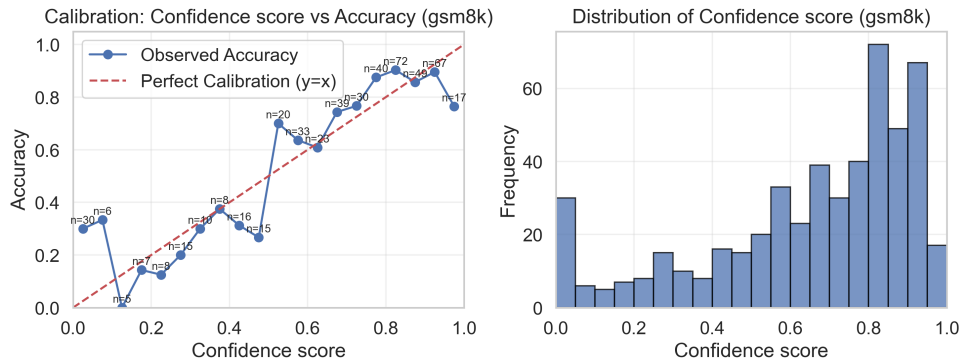
(a) BoolQ



(b) SQuAD 2.0



(c) TriviaQA



(d) GSM8K

Figure A.6: **CoCoA calibration with C_{CoCoA}.** Reliability (left) and confidence distribution (right). Labels above points denote bin counts. CoCoA aligns high-confidence bins with $y{=}x$ on SQuAD/TriviaQA and matches BoolQ's best VCE calibration, and tracks the diagonal well on GSM8K.