Coherent Confidence in Large Language Models

Krish Matta

May 2026

School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213

COMMITTEE

Matt Fredrikson
Andy Zou

Submitted in partial fulfillment of the requirements
for the Senior Honors Thesis

Keywords: large language models, confidence estimation, calibration, uncertainty quantification, epistemology

Abstract

Confidence estimation in large language models is evaluated almost exclusively through calibration: whether stated confidence matches empirical accuracy. We argue that calibration alone is insufficient, showing that it admits trivially uninformative estimators and provides no guarantee about individual predictions. We formalize a distinction between the confidence function, defined over semantic equivalence classes, and the confidence estimator, which operates on the text actually produced by the model. Building on this abstraction, we propose a framework of properties organized into three axes: structural coherence, faithfulness, and usefulness. We evaluate three confidence estimation methods against this framework and find that no existing method satisfies all three axes. The most commonly used methods produce distributions too degenerate to test any property meaningfully, while the only method with useful confidence estimates reveals that the model's underlying beliefs are not internally coherent.

To my family and mentors.

Contents

1 Introduction

As large language models are deployed in high-stakes settings such as medical triage, legal analysis, and autonomous decision-making, practitioners need to know when to trust a model's output. The natural mechanism for this is confidence estimation: assigning a probability that a model's response is correct. A reliable confidence estimate enables a simple and powerful decision rule: trust the model when confidence is high, defer to a human when it is low.

Confidence estimation in language models is a well-established problem. Tian et al. (2023) elicit confidence by directly asking the model to state its probability of correctness. Kadavath et al. (2022) extract confidence from output logits via a true/false verification prompt. Gekhman et al. (2024) estimate confidence by sampling multiple responses and measuring agreement. These methods are evaluated primarily, and often exclusively, through calibration (Geng et al. 2024), and recent high-profile benchmarks report calibration error as a central metric (Phan et al. 2026). We argue that this is not enough.

Calibration requires that among all predictions assigned confidence p, a fraction p are correct. It is important, but it is a property of a confidence function averaged over a distribution of inputs. It provides no guarantee about any individual prediction. We show that a constant predictor, one that assigns the same confidence to every question regardless of difficulty, can achieve perfect calibration while carrying zero instance-level information. More fundamentally, calibration is distribution-relative: a function well-calibrated on a benchmark can be arbitrarily miscalibrated on sub-populations of that same benchmark.

We argue that confidence estimation should be evaluated not only for calibration but also for internal coherence. A confidence function that assigns beliefs violating the probability axioms is not merely imprecise; it is uninterpretable. If a model reports 90% confidence in an answer and 85% confidence in a logically easier sub-question, no consistent reading of those numbers exists: they cannot both be right, and a practitioner relying on either one has no way to know which to trust. Such violations are not hypothetical. Our experiments show that they are present across existing confidence estimation methods.

We propose a framework for evaluating confidence along three axes. Structural properties require that confidence obey the probability axioms: normalization, the product rule for conjunctions, and monotonicity under entailment. Faithfulness properties require that the confidence estimator be insensitive to semantically irrelevant variation in how a prompt or answer is worded. Usefulness properties, namely calibration and discrimination, ask whether confidence tracks ground truth. Together, these provide a more complete picture than calibration alone.

We evaluate three confidence estimation methods, verbal confidence, logit-based confidence, and SliCK, against this framework using a mixture-of-experts reasoning model (Qwen-30B-A3B-Thinking). Verbal and logit-based confidence, the two most widely used methods, produce distributions so concentrated near 1.0 that they are effectively degenerate: the model reports near-certain confidence on almost every generation regardless of correctness. These degenerate distributions vacuously satisfy structural properties (there is no variance in which violations could manifest) while failing catastrophically on usefulness, with RMSCE values of 0.778 and 0.700 and AUROC barely above chance. This is particularly concerning for verbal confidence, the method a non-expert practitioner would reach for first, which is the worst-performing method across nearly every metric we study.

SliCK, which estimates confidence by sampling multiple rollouts and measuring agreement, is the only method that achieves meaningful calibration and discrimination. However, it reveals that the model's beliefs are not internally coherent: SliCK violates entailment monotonicity on 31% of questions and shows substantial conjunction consistency deviations, indicating that the model does not reliably assign higher confidence to easier questions or respect the product rule when composing sub-questions. These failures were invisible under verbal and logit-based estimation, not because they were absent, but because those estimators lacked the expressiveness to surface them.

No existing method performs well on all three axes of evaluation. We take this as evidence that confidence estimation in language models requires both better estimators and a richer evaluation methodology than calibration alone provides.

Our contributions are as follows. First, we formalize the distinction between the confidence function c, defined over semantic equivalence classes, and the confidence estimator $\hat{c}$, which operates on the text actually produced by the model. This separation is what allows us to ask whether an estimator faithfully recovers a well-formed underlying confidence function, a question that prior work does not pose. Second, we propose a framework of properties organized into three axes: structural coherence, faithfulness, and usefulness. Third, we evaluate three confidence estimation methods against this framework and find that no existing method performs well on all three axes. Verbal and logit-based confidence produce distributions too degenerate to test any property meaningfully, while SliCK, the only method with useful confidence estimates, reveals substantial violations of the structural properties that a coherent confidence function should satisfy.

2 Related Work

2.1 Confidence Estimation Methods

A range of methods have been proposed for estimating confidence in language model outputs. Verbal confidence (Tian et al. 2023) elicits a numerical probability by prompting the model to assess its own correctness. Logit-based confidence (Kadavath et al. 2022) extracts confidence from output token probabilities via a true/false verification prompt. Sampling-based methods estimate confidence by generating multiple responses and measuring agreement. SliCK (Gekhman et al. 2024) groups sampled generations into semantic equivalence classes and computes the fraction of rollouts falling into each class. Semantic entropy (Kuhn, Gal, and Farquhar 2023; Farquhar et al. 2024) similarly clusters generations by meaning but computes entropy over the resulting distribution, producing an unbounded uncertainty measure rather than a $[0, 1]$ confidence score. Because our framework requires confidence values interpretable as probabilities, semantic entropy does not map well into the properties we define, and we do not evaluate it here. Kossen et al. (Kossen et al. 2024) train linear probes on hidden states to approximate semantic entropy at reduced cost, addressing the inference-time expense of sampling-based methods but inheriting the same representational limitation.

2.2 Evaluation and Calibration

The primary evaluation criterion for confidence estimation is calibration: whether stated confidence matches empirical accuracy (Geng et al. 2024). Recent benchmarks report calibration error as a central metric (Phan et al. 2026), and the finding that RLHF degrades calibration relative to base models (OpenAI et al. 2024) has further entrenched calibration as the canonical measure of confidence quality. We argue that calibration alone is insufficient and propose structural and faithfulness properties as complementary evaluation axes. Prior work treats confidence estimation and the underlying confidence function interchangeably: a method produces a number, and that number is evaluated for calibration. No existing framework separates properties of the confidence function itself from properties of the estimator used to approximate it.

The closest precedent to our structural analysis is Zhu and Griffiths (2025), who find that LLMs systematically violate probabilistic identities when asked to forecast external events. Our work differs in that we study confidence in the model's own correctness, which is the operationally relevant quantity for deployment, and that we compare specific estimation methods rather than raw model outputs, revealing that the choice of estimator determines whether violations are even detectable.

Mazeika et al. (2025) study whether LLM preferences satisfy the axioms of expected utility theory, finding that larger models exhibit increasingly coherent value systems. We apply the same logic to confidence: if confidence is to function as a probability, it should satisfy the corresponding axioms.

2.3 Improving Confidence Through Training

Several recent methods aim to improve confidence through training rather than post-hoc estimation. EliCal (Ni et al. 2026) proposes a two-stage framework that first elicits internal confidence via self-consistency supervision, then calibrates with a small set of correctness annotations. Rewarding Doubt (Bani-Harouni et al. 2026) uses reinforcement learning with a logarithmic scoring rule to train calibrated verbal confidence directly. Both methods target calibration as their primary objective. Our framework provides a broader set of evaluation criteria, including structural coherence and faithfulness, against which such training-based methods could be assessed.

3 Background

3.1 Confidence Functions

Let $\mathcal{M}$ be a language model, x denote a prompt, and y denote a generation produced by $\mathcal{M}$ in response to x. Furthermore, let $\simeq$ denote a semantic equivalence relation on strings, and write $[x]$ and $[y]$ for the corresponding equivalence classes. A confidence function c maps prompt-generation equivalence class pairs to the unit interval:

$$c : ([x], [y]) \mapsto [0, 1]$$

where $c([x], [y])$ represents the degree of belief that $[y]$ is a correct response to $[x]$.

We argue that $c([x], [y])$ should be interpretable as the probability of the event that $[y]$ is a correct response to $[x]$, and should therefore satisfy the usual probability axioms. Two classical arguments support this. The Dutch book argument Ramsey (1926) interprets $c([x], [y])$ as the price at which the agent would buy or sell a contract that pays \$1 if $[y]$ is a correct response to $[x]$ and \$0 otherwise. If c violates the probability axioms, there exists a finite combination of such bets that yields a guaranteed loss regardless of outcomes. Cox's theorem (Cox 1946) finds that under weak desiderata (real-valued credences, consistency with Boolean logic) any measure of belief is isomorphic to a probability measure.

We additionally define the prompt-level confidence as

$$\bar{c}([x]) = \max_{[y]} c([x], [y])$$

This admits a natural decision-theoretic justification: under 0-1 loss for correctness, a rational agent outputs $\arg\max_{[y]} c([x], [y])$, and its self-assessed probability of answering correctly is the confidence in that class.

3.2 Confidence Estimators

A confidence estimator $\hat{c}$ is a computable function that maps prompt-generation text pairs to the unit interval:

$$\hat{c} : (x, y) \mapsto [0, 1]$$

where $\hat{c}(x, y)$ attempts to estimate $c([x], [y])$. Unlike c, which is defined over equivalence classes, $\hat{c}$ operates on the text actually available to or produced by the model.

3.3 Confidence Estimation Methods

We compare three methods for computing $\hat{c}(x, y)$ in large language models.

Verbal Confidence. Following Tian et al. (2023), we first prompt the model to respond to x, then once it's provided a generation y, ask it in a follow up to provide the probability that its response is correct. The model's numerical response is parsed as $\hat{c}(x, y)$.

Logit-based Confidence. Following Kadavath et al. (2022), we prompt the model to verify whether its proposed answer y is true or false to the prompt x. Confidence is then computed as

$$\hat{c}(x, y) = \frac{P(\text{True})}{P(\text{True}) + P(\text{False})}$$

where P is computed from the output logits.

SliCK. Following Gekhman et al. (2024), we independently sample $k = 16$ generations $y_1, \ldots, y_k$ to the same prompt x. We group generations into semantic equivalence classes under $\hat{\simeq}$ using LLM-as-a-judge. Rollouts in which the model declines to answer or exhausts its token budget are excluded; let k' denote the number of remaining rollouts. The confidence for a generation y is the fraction of sampled responses equivalent to that generation, i.e.

$$\hat{c}(x, y) = \frac{|\{j : y_j \hat{\simeq} y\}|}{k'}.$$

This differs slightly from the original method, which does not exclude refusals.

3.4 Experimental Setup

We describe the shared infrastructure used throughout Sections 5 to 7. The dataset, generation procedure, and property-specific evaluation vary across experiments and are described in their respective sections.

Model. Unless otherwise stated, all experiments use Qwen-30B-A3B-Thinking (Yang et al. 2025), a 30B-parameter mixture-of-experts reasoning model with 3B active parameters. The same model serves as both the generation model and the evaluation model.

Across all methods, generations in which the model declines to answer or exhausts its token limit are excluded from evaluation.

3.5 Confidence Distributions

Figure 1 shows the distribution of confidence scores each method assigns across generations on three QA benchmarks. Verbal and logit-based confidence concentrate nearly all mass at extreme values, while SliCK produces scores across the full range.

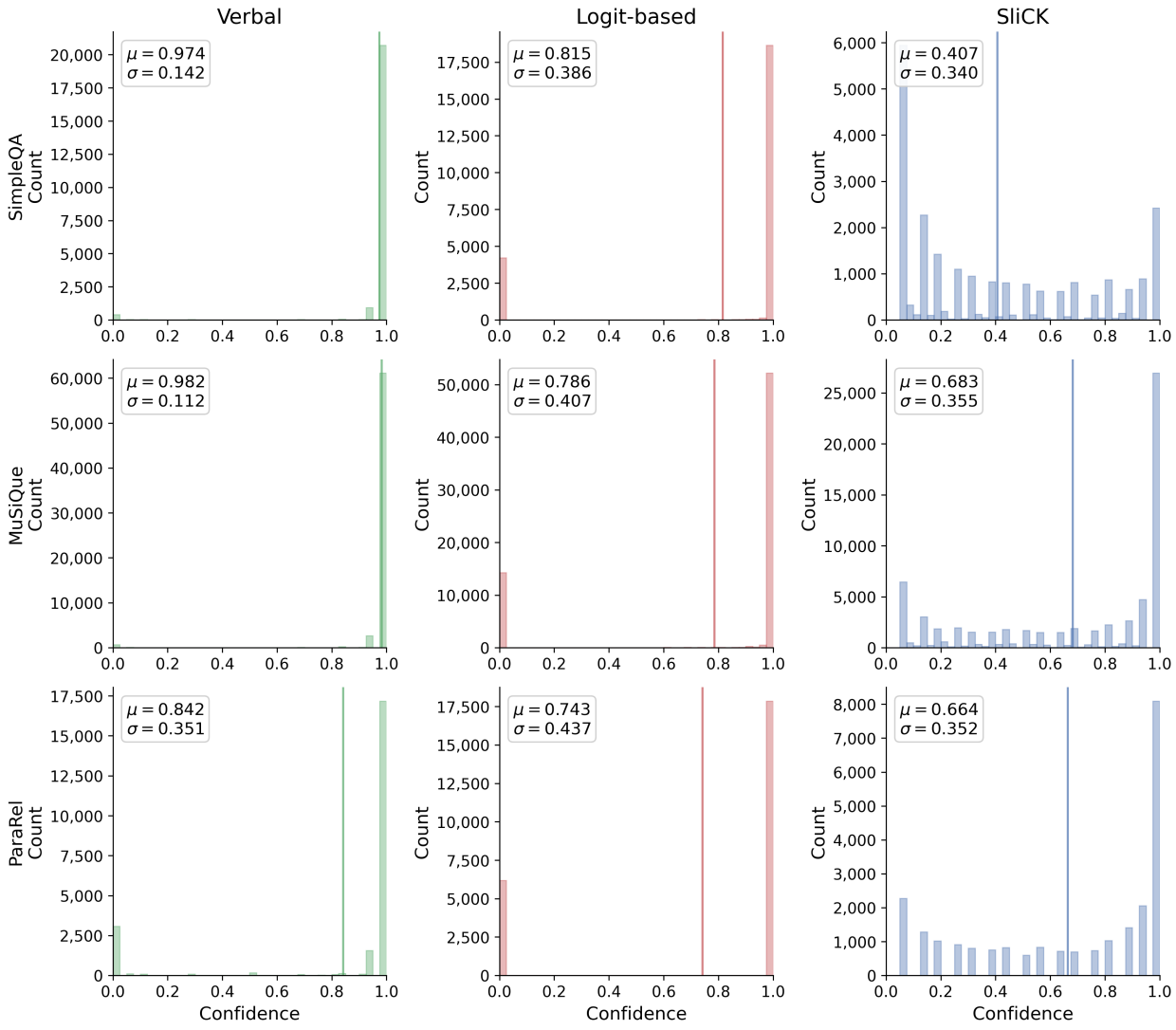

Figure 1: Distribution of confidence scores for each estimation method (columns) across three datasets (rows). Vertical lines indicate the mean.

4 Properties of Confidence Functions and Estimators

In this section, we define a set of properties for evaluating confidence in language models, covering both the confidence function c and the confidence estimator $\hat{c}$. These properties are organized into three categories. **Structural properties** are constraints on the confidence function c itself, ensuring that it obeys probability axioms. **Faithfulness properties** are constraints on the confidence estimator $\hat{c}$, ensuring that its behavior is consistent with the existence of a well-formed underlying c. **Usefulness properties** ask whether c tracks ground truth. Together, they provide a framework for evaluating confidence estimation methods that goes beyond asking "is the model well-calibrated?" to also ask "are the model's beliefs internally coherent?" and "does the estimator faithfully represent those beliefs?"

This approach is inspired by the *utility engineering* framework of Mazeika et al. (2025), who study whether LLM preferences satisfy the structural requirements of rational utility functions. We apply the same logic to confidence.

4.1 Calibration is Insufficient

Calibration is the primary evaluation criterion for confidence estimation in the LLM literature. A confidence function c is *calibrated* on a distribution of prompt-generation pairs $\mathcal{D}$ if for all $p \in [0, 1]$:

$$P_{(x,y)\sim\mathcal{D}}([y] \text{ is correct for } [x] \mid c([x], [y]) = p) = p.$$

Calibration is typically evaluated using the root-mean-square calibration error (RMSCE) (Phan et al. 2026), which bins confidence scores into $B = 20$ equal width bins over $[0, 1]$ and computes:

$$\text{RMSCE} = \sqrt{\sum_{b=1}^{B} \frac{n_b}{N} \left(\text{acc}_b - \mu_b\right)^2}$$

where n_b is the number of pairs in bin b, acc_b is the fraction of pairs in the bin for which $[y]$ is truly correct for $[x]$, and μ_b is the mean confidence in bin b.

Two of the confidence methods we study: verbal confidence (Tian et al. 2023) and logit-based confidence (Kadavath et al. 2022), are all evaluated primarily through calibration. Surveys of LLM uncertainty quantification (Geng et al. 2024) organize the field around calibration as the primary desideratum, and recent benchmarks such as Humanity's Last Exam (Phan et al. 2026) report calibration error as a central evaluation metric. The GPT-4 technical report's finding that RLHF degrades calibration relative to the base model (OpenAI et al. 2024) has further entrenched calibration as the canonical evaluation metric for confidence.

Unfortunately, calibration as a sole criterion of confidence admits confidence functions that are internally incoherent. We illustrate this with two examples.

Constant predictor. Let α denote the model's marginal accuracy on a distribution of prompts $\mathcal{D}$. The confidence function $c([x], [y]) = \alpha$ for all (x, y) is perfectly calibrated on $\mathcal{D}$, achieving a RMSCE of exactly zero. Let $b^\star$ denote the bin containing α. Since c is constant, every prediction falls into $b^\star$, so $n_{b^\star} = N$ and $n_b = 0$ for all $b \neq b^\star$. The mean confidence is $\mu^\star = \alpha$, and moreover, $\text{acc}_b = \alpha$. Hence,

$$\text{RMSCE} = \sqrt{\sum_{b=1}^{B} \frac{n_b}{N} \left(\text{acc}_b - \mu_b\right)^2} = \sqrt{\frac{N}{N}(\alpha - \alpha)^2} = 0.$$

Yet, c carries no instance-level information and cannot distinguish a question the model answers reliably from one it answers by chance.

Calibration is distribution-relative. More fundamentally, calibration is not a property of a confidence function c alone, but of the confidence function c *paired* with the dataset $\mathcal{D}$. A function well-calibrated on $\mathcal{D}$ can be arbitrarily miscalibrated on sub-distributions of $\mathcal{D}$ itself.

To see this, let c have RMSCE $= 0$ on $\mathcal{D}$. Then, let $\mathcal{D}' \subset \mathcal{D}$ be the sub-distribution of examples the model answers incorrectly. On $\mathcal{D}'$, every bin has accuracy zero. The corresponding bins contribute a strictly positive term $(n'_b/N') \cdot (\bar{c}'_b)^2$ to the squared RMSCE. Hence, $\text{RMSCE}(\mathcal{D}') > 0 = \text{RMSCE}(\mathcal{D})$.

This is not a hypothetical concern. Any partition of $\mathcal{D}$ (for example, by topic) yields sub-distributions on which c may be substantially miscalibrated. RMSCE reported on a benchmark thus characterizes behavior on that benchmark in aggregate, and provides no guarantee about the sub-populations one actually cares about.

4.2 Structural Properties

Structural properties are hard constraints that follow from the position that c behave as a probability measure. A confidence function that violates them is internally incoherent, assigning beliefs that contradict each other.

4.2.1 Normalization

Definition. For a prompt class $[x]$:

$$\sum_{[y]} c([x], [y]) = 1.$$

Justification. Normalization follows directly from the axioms of probability. The answer classes $[y]$ form a partition of the space of possible responses, hence their probabilities must sum to one.

4.2.2 Conjunction Consistency

Definition. Suppose that correctly responding to $[x]$ decomposes into answering sub-prompt $[x_1]$ with correct answer $[y_1^\star]$, followed by answering a sub-prompt $[x_2]$ whose correct answer coincides with the answer to $[x]$. Then,

$$\bar{c}([x]) = c([x_1], [y_1^\star]) \cdot \bar{c}([x_2] \mid [x_1], [y_1^\star]).$$

Justification. The event "answers $[x]$ correctly" is the conjunction of two events: $[y_1^*]$ is correct for $[x_1]$, and the best answer to $[x_2]$ given $[y_1^*]$ is correct. The product rule of probability gives $P(A \cap B) = P(A) \cdot P(B \mid A)$.

4.2.3 Entailment Monotonicity

Definition. Let $[x]$ and $[x']$ be two prompt classes such that answering $[x]$ correctly entails answering $[x']$ correctly. Then,

$$\bar{c}([x]) \leq \bar{c}([x']).$$

Justification. Each $\bar{c}([x])$ is a credence in the event "the model's best answer to $[x]$ is correct." If answering $[x]$ correctly entails answering $[x']$ correctly, the former event is a subset of the latter. A coherent assignment of credences to events must respect set inclusion. Intuitively, a strictly harder question should never receive higher confidence than an easier one.

4.3 Faithfulness Properties

Faithfulness properties characterize how well the estimator $\hat{c}$ approximates the underlying confidence function c. Since c is defined over equivalence classes, it is by construction invariant to how a prompt or answer is expressed. But $\hat{c}$ operates on strings, meaning that there is no guarantee it recovers the same value for equivalent inputs. Violations indicate that $\hat{c}$ is sensitive to features of the text that are invisible at the equivalence class level, and therefore cannot faithfully represent c.

4.3.1 Prompt Semantic Invariance

Definition. For prompts $x \simeq x'$,

$$\hat{c}(x, y) = \hat{c}(x', y).$$

Justification. If $\hat{c}$ faithfully estimates $c([x], [y])$, it must be invariant to the choice of representative from the prompt equivalence class.

4.3.2 Generation Semantic Invariance

Definition. For prompts $y \simeq y'$,

$$\hat{c}(x, y) = \hat{c}(x, y').$$

Justification. If $\hat{c}$ faithfully estimates $c([x], [y])$, it must be invariant to the choice of representative from the generation equivalence class.

4.4 Usefulness

Usefulness properties measures whether confidence tracks ground truth.

4.4.1 Calibration

Definition. A confidence function c is *calibrated* on a distribution of prompt-generation pairs $\mathcal{D}$ if for all $p \in [0, 1]$:

$$P_{(x,y)\sim\mathcal{D}}([y] \text{ is correct for } [x] \mid c([x], [y]) = p) = p.$$

Justification. If $c([x], [y])$ is to be interpretable as a probability that $[y]$ is correct for $[x]$, then among all pairs assigned confidence p, exactly a fraction p should be correct.

4.4.2 Discrimination

Definition. A confidence function c has perfect discrimination if for all pairs $([x], [y])$ and $([x'], [y'])$ drawn from $\mathcal{D}$, whenever $[y]$ is correct for $[x]$ and $[y']$ is incorrect for $[x']$:

$$c([x], [y]) > c([x'], [y']).$$

Justification. Calibration asks whether confidence values are accurate as probabilities; discrimination asks whether confidence ranking separates correct from incorrect responses. Notably, a constant predictor achieves perfect calibration but chance-level discrimination.

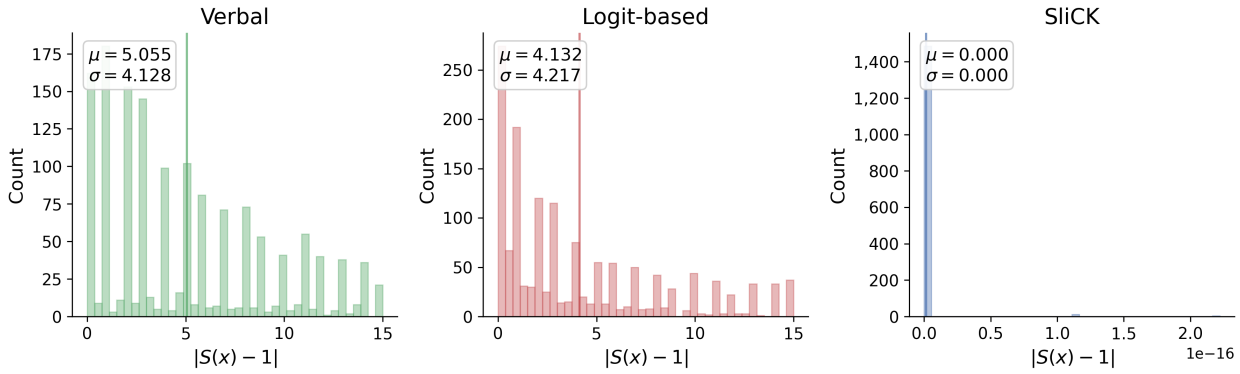

Figure 2: Distribution of normalization deviation $|S(x) - 1|$ for three confidence estimation methods on SimpleQA. Verbal and logit-based confidence severely violate normalization, with mean deviations of 5.055 and 4.132 respectively, indicating that confidence sums far exceed 1. SliCK perfectly satisfies normalization (mean deviation 0).

5 Structural Coherence

The structural properties defined in Section 4.2 are constraints on the confidence function c. Since c is not directly observable, we evaluate whether each estimator $\hat{c}$ in Section 3.3 satisfies them. Violations may indicate that the model's underlying beliefs are incoherent, or that the estimator fails to faithfully represent a coherent c. Moreover, different confidence estimators may be approximating different underlying confidence functions, so failures should be interpreted per-method rather than as statements about the model's beliefs in general.

5.1 Normalization

5.1.1 Experimental Setup

We sample $N = 1{,}500$ questions from SimpleQA (Wei et al. 2024) and generate $k = 16$ rollouts per question at temperature $T = 0.5$, faithful to the original SliCK method (Gekhman et al. 2024). Rollouts are grouped into equivalence classes $[y_1], \ldots, [y_m]$ under $\hat{\simeq}$. For each class $[y_j]$, we compute the mean confidence across its members, then sum across classes:

$$S(x) = \sum_{j=1}^{m} \frac{1}{|[y_j]|} \sum_{y \in [y_j]} \hat{c}(x, y).$$

A well-normalized confidence function satisfies $S(x) = 1$ for every question. We report the distribution of $|S(x) - 1|$ across all questions. Note that the sum is only over equivalence classes observed in k rollouts, so the reported violations are lower bounds on the true normalization deviation.

5.1.2 Results

Figure 2 shows the distribution of normalization deviation for each method. Verbal and logit-based confidence violate normalization severely, with mean deviations of 5.055 and 4.132, meaning that the model assigns high confidence to many distinct answer groups, producing sums that far exceed 1. SliCK satisfies normalization exactly with a mean deviation 0. This is expected: by construction, SliCK assigns each rollout y_i a confidence of $|[y_i]|/k'$, and these fractions sum to 1.

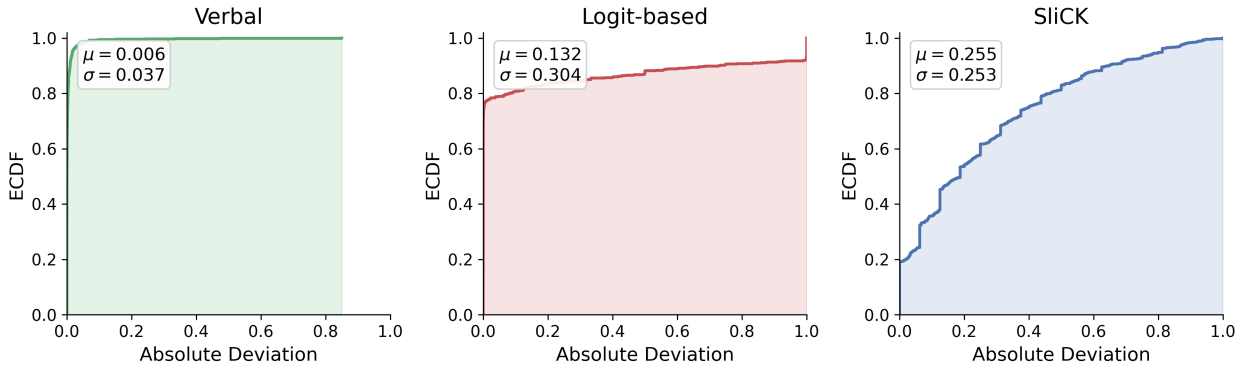

Figure 3: Empirical CDF of conjunction consistency deviation $\Delta(x) = |\bar{c}(x) - c(x_1, y_1^\star) \cdot \bar{c}(x_2 \mid x_1, y_1^\star)|$ for three confidence estimation methods on MuSiQue.

5.2 Conjunction Consistency

5.2.1 Experimental Setup

We use MuSiQue (Trivedi et al. 2021), a multi-hop QA dataset where each question is constructed by composing single-hop sub-questions. We filter to 2-hop answerable questions with natural language sub-questions. Each question decomposes into a full question x, a first hop x_1 with gold answer $y_1^\star$, and a second hop x_2.

We evaluate conjunction consistency (Section 4.2.2). For each question we prompt $\mathcal{M}$ three times: with x, with x_1, and with x_2 preceded by the first-hop gold answer $y_1^\star$ as prior context. We generate $k = 16$ rollouts per prompt at temperature $T = 0.5$. We operationalize $c(x_1, y_1^\star)$ as the mean confidence across rollouts of x_1 that are judged correct, and $\bar{c}$ as the max confidence across rollouts. Questions where x_1 has zero correct rollouts are dropped, since $c(x_1, y_1^\star)$ is undefined.

We measure the deviation:

$$\Delta(x) = |\bar{c}(x) - c(x_1, y_1^\star) \cdot \bar{c}(x_2 \mid x_1, y_1^\star)|\,,$$

and report the distribution of $\Delta(x)$ across all questions.

5.2.2 Results

Figure 3 shows the empirical CDF of conjunction consistency deviation for each method. Verbal confidence achieves the lowest mean deviation (0.006), with nearly all questions at zero deviation. This is vacuous: as shown in Figure 1, verbal confidence assigns $c \approx 1.0$ to nearly all generations, so $1.0 \approx 1.0 \times 1.0$ holds trivially. Logit-based confidence shows a mean deviation of 0.132, with roughly 80% of questions at near-zero deviation and a long tail extending to 1.0. The concentration at zero reflects the same saturation near 1.0, with violations occurring only on the minority of questions where logit-based confidence departs from the ceiling. SliCK shows the largest mean deviation (0.255), indicating substantial violations of the product rule. Unlike verbal and logit-based confidence, these violations are substantive: SliCK produces scores across the full range, so the deviations reflect genuine failures of the product rule rather than artifacts of saturation.

5.3 Entailment Monotonicity

5.3.1 Experimental Setup

We use MuSiQue Trivedi et al. (2021) and filter to 2-hop answerable questions with natural language sub-questions. Each question decomposes into a full question x, a first hop x_1 with gold answer $y_1^\star$, and a second hop x_2.

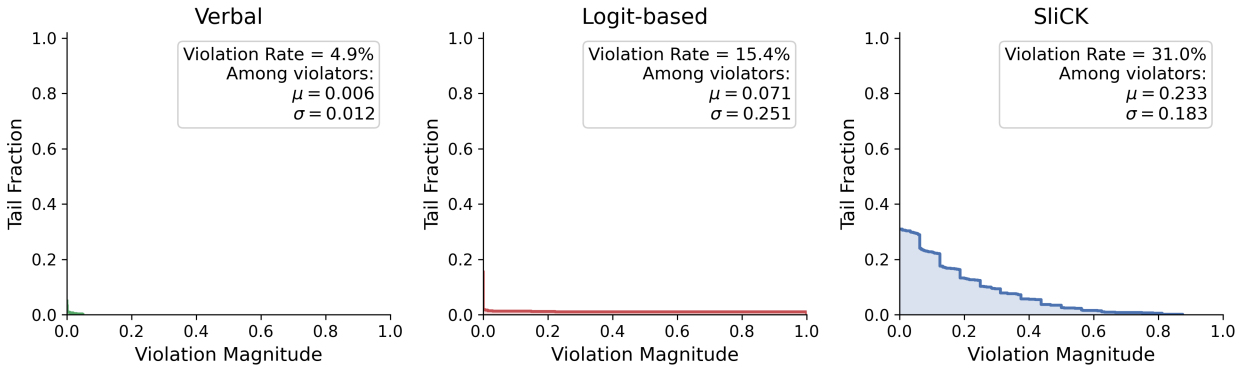

Figure 4: Survival function of entailment monotonicity violation magnitude $\Delta(x) = \max(0, \bar{\hat{c}}(x) - \bar{\hat{c}}(x_2 \mid x_1, y_1^*))$ for three confidence estimation methods on MuSiQue. The value at $\Delta(0)$ corresponds to the overall violation rate. Reported μ and σ are conditional on $\Delta(x) > 0$.

For each question, we prompt $\mathcal{M}$ twice: first with x, then with x_2 preceded by $y_1^\star$ as prior context. We generate $k = 16$ rollouts per prompt at temperature $T = 0.5$ and estimate each $\bar{c}$ as the max confidence across rollouts. We then report the distribution of violations

$$\Delta(x) = \max(0, \bar{\hat{c}}(x) - \bar{\hat{c}}(x_2 \mid x_1, y_1^\star))$$

across questions.

5.3.2 Results

Figure 4 shows the survival function of violation magnitude $\Delta(x)$ for each method. Verbal confidence violates monotonicity on only 4.9% of questions, with a conditional mean magnitude of 0.006. Logit-based confidence violates on 15.4% with a conditional mean magnitude of 0.071. However, as shown in Figure 1, both estimators saturate near 1.0 on MuSiQue, leaving little room for $\bar{c}([x])$ to exceed $\bar{c}([x_2] \mid [x_1], [y_1^*])$. Low violation rates at the ceiling do not indicate that the estimator respects the entailment ordering; rather, they indicate that it cannot distinguish hard prompts from easy ones.

SliCK violates monotonicity on 31.0% of questions with a conditional mean magnitude of 0.233. Unlike verbal and logit-based confidence, SliCK produces scores across the full range, so these violations are substantive: providing the first-hop answer y_1^*, which by construction makes the second-hop question strictly easier than the original, does not reliably increase confidence.

6 Faithfulness

6.1 Prompt Semantic Invariance

6.1.1 Experimental Setup

We use ParaRel (Elazar et al. 2021), a dataset of cloze-style factual prompts in which each fact is expressed by multiple paraphrase templates. We sample $N = 1{,}500$ facts, selecting exactly two paraphrase templates per fact.

For each prompt we generate $k = 16$ rollouts at temperature $T = 0.5$. Rollouts are grouped into equivalence classes under $\hat{\simeq}$, and we estimate $\bar{c}$ as the maximum of $\hat{c}$. For each fact with paraphrases x, x', we compute:

$$\Delta(f) = |\bar{\hat{c}}(x) - \bar{\hat{c}}(x')|,$$

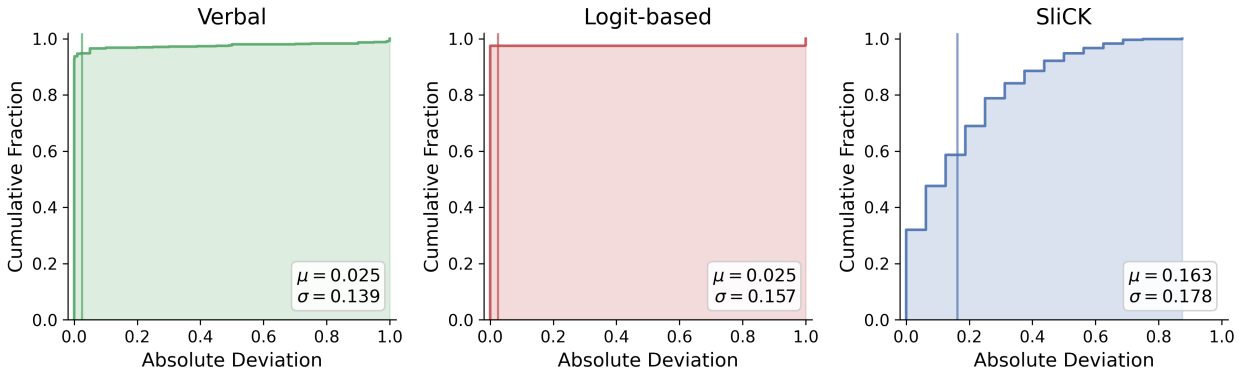

Figure 5: Empirical CDF of prompt semantic invariance deviation $\Delta(f) = |\bar{\hat{c}}(x) - \bar{\hat{c}}(x')|$ for three confidence estimation methods on ParaRel. Verbal and logit-based confidence both show mean deviations of 0.025, vacuously low due to saturation near 1.0. SliCK shows substantially higher deviation ($\mu = 0.163$), indicating sensitivity to prompt phrasing.

and report the distribution of $\Delta(f)$ across facts.

6.1.2 Results

Figure 5 shows the empirical CDF of prompt semantic invariance deviation $\Delta(f)$ for each method. Verbal and logit-based confidence both achieve a mean deviation of 0.025, with the vast majority of paraphrase groups showing near-zero spread. As with earlier properties, this is vacuous. Both estimators saturate near 1.0 regardless of prompt phrasing, so there is little room for deviation across paraphrases.

SliCK shows substantially higher deviation ($\mu = 0.163$), with a broad distribution of spread values across paraphrase groups. Because SliCK estimates confidence by sampling rollouts independently for each prompt, different phrasings of the same fact can yield different answer distributions and thus different confidence estimates. This is a genuine faithfulness failure: the estimator is sensitive to prompt phrasing, which is invisible at the equivalence class level.

6.2 Generation Semantic Invariance

6.2.1 Experimental Setup

We use SimpleQA (Wei et al. 2024) and sample $N = 1{,}500$ questions, generating $k = 16$ rollouts per question at temperature $T = 0.5$. Rollouts are grouped into equivalence classes under $\hat{\simeq}$. For each class with at least one member, we compute the spread of confidence estimates within the class:

$$\Delta([y]) = \max_{y \in [y]} \hat{c}(x, y) - \min_{y \in [y]} \hat{c}(x, y).$$

We report the distribution of $\Delta([y])$ across all equivalence classes.

6.2.2 Results

Figure 6 shows the distribution of within-class confidence spread for each method. SliCK satisfies generation semantic invariance exactly ($\mu = 0.000$), as expected: by construction, all members of an equivalence class receive the same confidence.

Verbal confidence has a mean spread of 0.074, with most classes concentrated near zero but a notable spike at 1.0. These are generations the estimator considers semantically equivalent but assigns maximally different confidence to. Logit-based confidence shows a similar pattern with a higher mean spread of 0.178 and more pronounced mass near 1.0.

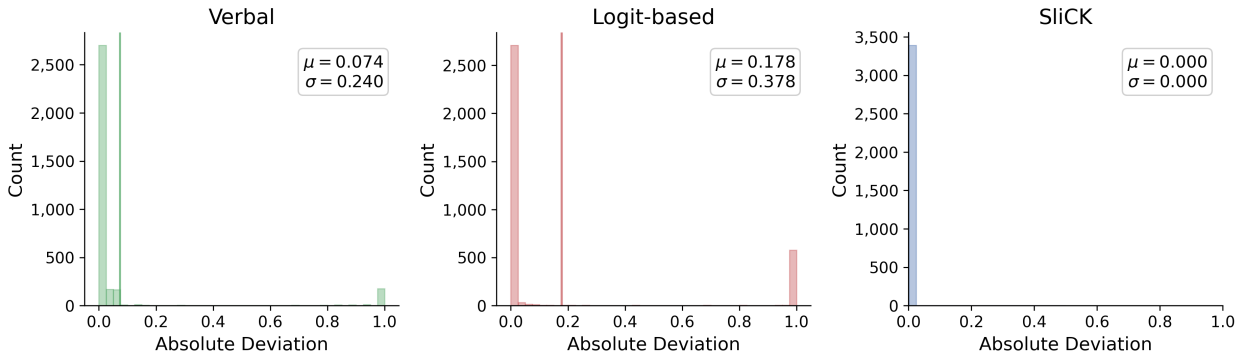

Figure 6: Distribution of within-class confidence spread $\Delta([y]) = \max \hat{c} - \min \hat{c}$ for three confidence estimation methods on SimpleQA. Verbal and logit-based confidence show mean spreads of 0.074 and 0.178 respectively, with notable mass at 1.0 indicating maximally inconsistent estimates within equivalence classes. SliCK satisfies generation semantic invariance exactly ($\mu = 0.000$).

Both verbal and logit-based estimators are string-level: their confidence depends on the exact wording of the generation, not just its meaning. This is precisely the failure that faithfulness properties are designed to detect.

7 Usefulness

7.1 Calibration

7.1.1 Experimental Setup

We sample $N = 1,500$ questions from SimpleQA Wei et al. (2024). For each question, we generate $k = 16$ independent rollouts at temperature $T = 0.5$, following (Gekhman et al. 2024), with a maximum sequence length of 8,192 tokens. Each generation is distilled into a canonical short-form answer by a semantic summarizer and graded for correctness against the gold-standard reference (details in Appendix 10.1).

We evaluate calibration using the root-mean-square calibration error (RMSCE) (Phan et al. 2026). We bin confidence scores into $B = 20$ equal width bins over $[0, 1]$ and compute:

$$\text{RMSCE} = \sqrt{\sum_{b=1}^{B} \frac{n_b}{N} \left(\text{acc}_b - \bar{c}_b\right)^2}$$

where n_b is the number of samples in bin b, acc_b is the fraction correct, and $\bar{c}_b$ is the mean confidence. Lower RMSCE indicates better calibration.

7.1.2 Results

Figure 7 shows calibration diagrams and confidence distributions for each method. SliCK is the only method that achieves meaningful calibration, with an RMSCE of 0.251 compared to 0.778 for verbal and 0.700 for logit-based confidence.

The failure of verbal and logit-based confidence is straightforward: the model reports high confidence on nearly all generations regardless of correctness, so the resulting scores carry little information. SliCK produces a broader distribution of confidence scores, and these scores correlate with correctness much more reliably.

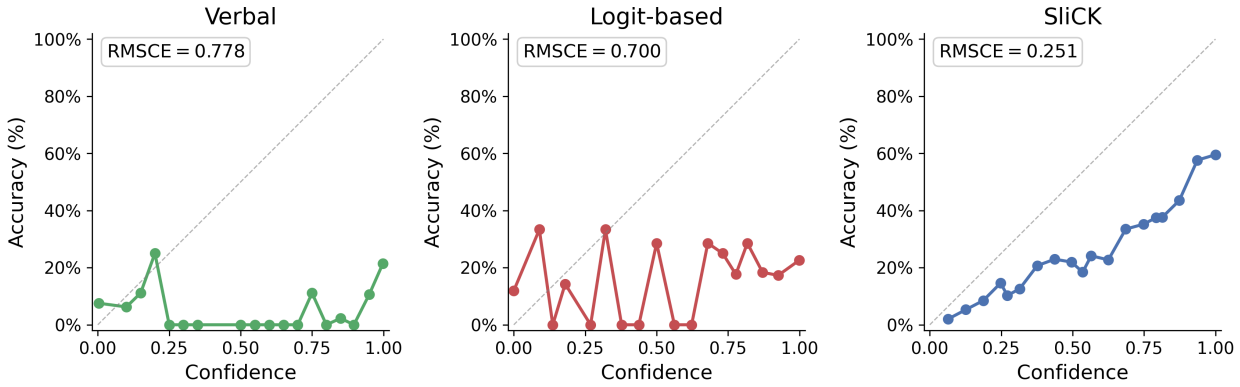

Figure 7: Calibration diagrams (top) and confidence distributions (bottom) for three confidence estimation methods on SimpleQA. SliCK achieves the lowest RMSCE (0.251) and is the only method for which accuracy roughly increases with confidence. Verbal and logit-based confidence concentrate nearly all mass at extreme values, producing RMSCE of 0.778 and 0.700 respectively. The dashed diagonal represents perfect calibration.

7.2 Discrimination

7.2.1 Experimental Setup

We measure discrimination using the area under the ROC curve (AUROC), which equals the probability that a randomly drawn correct generation receives higher confidence than a randomly drawn incorrect generation. An AUROC of 1.0 indicates perfect discrimination; 0.5 indicates chance.

We use SimpleQA (Wei et al. 2024) and sample $N = 1{,}500$ questions, generating $k = 16$ rollouts per question at temperature $T = 0.5$. Each rollout is graded for correctness against the gold-standard reference. We compute AUROC over all (confidence, correctness) pairs.

7.2.2 Results

SliCK achieves an AUROC of 0.825, indicating that its confidence estimates meaningfully separate correct from incorrect generations. Verbal and logit-based confidence achieve AUROCs of 0.559 and 0.596, respectively, which is above the 0.5 random baseline but only weakly discriminative. Both estimators assign confidence near 1.0 to nearly all generations regardless of correctness, leaving little room for the ranking to separate the two groups.

8 Discussion and Conclusion

Our experiments reveal a consistent pattern: every confidence estimator we study exhibits serious failures along at least one of the three evaluation axes. Moreover, the methods that appear to satisfy structural properties often do so vacuously, while the method that exposes genuine violations is also the only one that produces useful confidence estimates. We organize the discussion around three themes.

8.1 Degenerate Distributions Mask Incoherence

Verbal and logit-based confidence concentrate nearly all probability mass near 1.0, producing effectively degenerate distributions. This saturation creates the appearance of structural coherence: normalization deviations are large but conjunction consistency and entailment monotonicity violations are small, not because the estimator respects the probability axioms, but because $1.0 \approx 1.0 \times 1.0$ holds trivially. A confidence function that assigns the same value to every generation cannot violate ordering constraints, but neither can it satisfy them in any meaningful sense. The low violation rates of verbal and logit-based confidence on structural properties should therefore be interpreted as uninformative rather than as evidence of rationality, stressing the need to evaluate confidence along multiple axes before trusting it in practice.

This is particularly concerning for verbal confidence, which is the most accessible estimation method: it requires no access to logits, no multiple sampling, and no semantic equivalence judgments. One simply asks the model how confident it is. This is precisely what a non-expert practitioner or end user would do. Yet verbal confidence is the worst-performing method across nearly every metric we study, producing an RMSCE of 0.778, an AUROC of 0.559, and mean normalization deviations exceeding 5. In safety-critical deployment contexts, where practitioners may rely on a model's self-reported confidence to decide when to trust its outputs, the near-total uninformativeness of the most natural estimation method is a serious concern.

8.2 Informative Confidence Exposes Irrational Beliefs

SliCK is the only method that produces confidence scores across the full $[0, 1]$ range, and it is the only method that achieves meaningful calibration (RMSCE = 0.251 vs. 0.778 and 0.700) and discrimination (AUROC = 0.825 vs. 0.559 and 0.596). However, this same informativeness exposes substantial structural violations: SliCK shows a mean conjunction consistency deviation of 0.255 and violates entailment monotonicity on 31.0% of questions with a conditional mean magnitude of 0.233. These are not artifacts of saturation, reflecting genuine failures of the probability axioms. Providing the first-hop answer, which by construction makes the remaining question strictly easier, does not reliably increase the model's confidence.

This creates an apparent tension: the only method whose confidence estimates are useful is also the one that most clearly violates the structural requirements we argue confidence should satisfy. We do not view this as undermining the framework. Rather, it suggests that current language models do not maintain internally coherent beliefs about their own correctness. Verbal and logit-based confidence obscure this by collapsing scores to the ceiling, leaving no variance in which violations could manifest. SliCK, by producing scores across the full range, makes these violations measurable for the first time.

8.3 Faithfulness Tracks the Level of Abstraction

Faithfulness results further illustrate the tradeoff between informativeness and coherence. Verbal and logit-based confidence operate on strings and are sensitive to the exact wording of both prompts and generations: semantically equivalent generations can receive maximally different confidence scores (spreads of 1.0 within equivalence classes). SliCK, by construction, assigns identical confidence to all members of an equivalence class, perfectly satisfying generation semantic invariance. However, SliCK violates prompt semantic invariance ($\mu = 0.163$), because independent rollouts from paraphrased prompts can yield different answer distributions. This suggests that the model's generation behavior is sensitive to surface-level prompt features, a deeper issue that no post-hoc estimation method can fully resolve.

8.4 Implications

Our results suggest that calibration alone is insufficient as an evaluation criterion for LLM confidence. A constant predictor achieves perfect calibration while carrying no instance-level information, and the methods most commonly evaluated through calibration in the literature (verbal and logit-based confidence) produce distributions too degenerate to test any property meaningfully. We advocate for evaluating confidence estimation methods along all three axes (structural coherence, faithfulness, and usefulness) to obtain a more complete picture. The framework we propose is not specific to the three methods studied here and can be applied to any confidence estimation method for language models.

8.5 Limitations and Future Work

All experiments use a single model (Qwen-30B-A3B-Thinking) as both the generator and evaluator. The extent to which these findings generalize across model families and scales is unknown.

Several directions for future work follow from these results. Extending the framework to open-ended generation, where correctness is less well-defined, and studying how confidence properties vary across model scales and between base and instruction-tuned models would test the generality of our findings.

8.5.1 Training Coherent Confidence via Dutch Book Reinforcement Learning

Our results show that no existing method satisfies all three axes. A natural follow-up is to ask whether training-time interventions can produce confidence estimators that do. We outline a concrete approach motivated by the classical argument that underlies our structural properties.

The Dutch book argument (Ramsey 1926) establishes that an agent whose credences violate the probability axioms can be exploited by a counterparty through a finite combination of bets that yields a guaranteed loss regardless of outcomes. We propose using this exploitability as a reinforcement learning reward signal.

Concretely, consider a frozen answer model $\mathcal{M}$ augmented with a trainable adapter active only during confidence turns. An episode presents $\mathcal{M}$ with a set of related prompts $x_1, \ldots, x_n$ with known logical relationships $\mathcal{S}$. For each x_i, the frozen model generates an answer y_i; the adapter then emits a price $c_i \in [0, 1]$, framed as the cost of a contract paying \$1 if y_i is correct. The episode reward combines two terms:

$$R(\mathbf{c}, \mathbf{e}, \mathcal{S}) = -\textsc{Exploit}(\mathbf{c}, \mathcal{S}) - \lambda \cdot \frac{1}{n} \sum_{i=1}^{n} (c_i - e_i)^2$$

where $\mathbf{e} \in \{0, 1\}^n$ is the realized correctness vector and $\mathcal{S}$ encodes the logical relationships among the prompts in the episode (for example, that answer classes partition the outcome space, or that correctness on one prompt entails correctness on another).

The first term, $\textsc{Exploit}(\mathbf{c}, \mathcal{S})$, is the worst-case guaranteed profit an optimal adversary extracts from the stated prices given $\mathcal{S}$. This is the value of a linear program: the adversary chooses stakes $\alpha_i \in [-1, 1]$ on each contract, and the agent's loss under any joint outcome consistent with $\mathcal{S}$ is $\sum_i \alpha_i(c_i - \omega_i)$. This value is non-negative and equals zero if and only if the prices are coherent with $\mathcal{S}$. The second term is the Brier score, a strictly proper scoring rule that anchors prices to ground-truth correctness and prevents trivially coherent but uninformative solutions such as posting $c = 1$ on one answer class and $c = 0$ on all others.

The central observation is that every property in our framework maps onto a constraint set $\mathcal{S}$ in the same LP. Normalization episodes constrain answer classes to partition the outcome space. Conjunction episodes encode the product rule. Entailment episodes impose subset constraints. Paraphrase episodes require equal pricing for semantically equivalent prompts, so that any price difference becomes a guaranteed-profit spread for the adversary. A single reward formulation thus covers all three axes (structural coherence, faithfulness, and usefulness) without requiring separate loss terms for each property.

Because verbal confidence is part of the model's own generation, the reward can directly shape the emitted values, making it the natural target for this intervention. If a trained verbal estimator matches SliCK on calibration and discrimination while satisfying the coherence properties that SliCK cannot express, at one forward pass rather than sixteen rollouts, the practical case for this approach would be substantial.

9 References

References

Bani-Harouni, David et al. (2026). *Rewarding Doubt: A Reinforcement Learning Approach to Calibrated Confidence Expression of Large Language Models*. arXiv: 2503.02623 [cs.CL]. URL: https://arxiv.org/abs/2503.02623.

Cox, Richard T. (1946). "Probability, Frequency and Reasonable Expectation". In: *Journal of Symbolic Logic* 37.2, pp. 398–399. DOI: 10.2307/2272983.

Elazar, Yanai et al. (2021). *Measuring and Improving Consistency in Pretrained Language Models*. arXiv: 2102.01017 [cs.CL]. URL: https://arxiv.org/abs/2102.01017.

Farquhar, Sebastian et al. (2024). "Detecting hallucinations in large language models using semantic entropy". In: *Nature* 630.8017, pp. 625–630. DOI: 10.1038/s41586-024-07421-0.

Gekhman, Zorik et al. (2024). *Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations?* arXiv: 2405.05904 [cs.CL]. URL: https://arxiv.org/abs/2405.05904.

Geng, Jiahui et al. (2024). *A Survey of Confidence Estimation and Calibration in Large Language Models*. arXiv: 2311.08298 [cs.CL]. URL: https://arxiv.org/abs/2311.08298.

Kadavath, Saurav et al. (2022). *Language Models (Mostly) Know What They Know*. arXiv: 2207.05221 [cs.CL]. URL: https://arxiv.org/abs/2207.05221.

Kossen, Jannik et al. (2024). *Semantic Entropy Probes: Robust and Cheap Hallucination Detection in LLMs*. arXiv: 2406.15927 [cs.CL]. URL: https://arxiv.org/abs/2406.15927.

Kuhn, Lorenz, Yarin Gal, and Sebastian Farquhar (2023). *Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation*. arXiv: 2302.09664 [cs.CL]. URL: https://arxiv.org/abs/2302.09664.

Mazeika, Mantas et al. (2025). *Utility Engineering: Analyzing and Controlling Emergent Value Systems in AIs*. arXiv: 2502.08640 [cs.LG]. URL: https://arxiv.org/abs/2502.08640.

Ni, Shiyu et al. (2026). *Annotation-Efficient Universal Honesty Alignment*. arXiv: 2510.17509 [cs.CL]. URL: https://arxiv.org/abs/2510.17509.

OpenAI et al. (2024). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

Phan, Long et al. (Jan. 2026). "A benchmark of expert-level academic questions to assess AI capabilities". In: *Nature* 649.8099, pp. 1139–1146. ISSN: 1476-4687. DOI: 10.1038/s41586-025-09962-4. URL: http://dx.doi.org/10.1038/s41586-025-09962-4.

Ramsey, Frank P. (1926). "Truth and Probability". In: *The Foundations of Mathematics and other Logical Essays*. Ed. by R. B. Braithwaite. McMaster University Archive for the History of Economic Thought. Chap. 7, pp. 156–198. URL: https://EconPapers.repec.org/RePEc:hay:hetcha:ramsey1926.

Tian, Katherine et al. (2023). *Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models Fine-Tuned with Human Feedback*. arXiv: 2305.14975 [cs.CL]. URL: https://arxiv.org/abs/2305.14975.

Trivedi, Harsh et al. (2021). "MuSiQue: Multi-hop Questions via Single-hop Question Composition". In: *CoRR* abs/2108.00573. arXiv: 2108.00573. URL: https://arxiv.org/abs/2108.00573.

Wei, Jason et al. (2024). *Measuring short-form factuality in large language models*. arXiv: 2411.04368 [cs.CL]. URL: https://arxiv.org/abs/2411.04368.

Yang, An et al. (2025). *Qwen3 Technical Report*. arXiv: 2505.09388 [cs.CL]. URL: https://arxiv.org/abs/2505.09388.

Zhu, Jian-Qiao and Thomas L. Griffiths (2025). *Incoherent Probability Judgments in Large Language Models*. arXiv: 2401.16646 [cs.CL]. URL: https://arxiv.org/abs/2401.16646.

10 Appendix

10.1 Prompts

10.1.1 Verbal Confidence

After the model generates a response y to a prompt x, we append the following follow-up message to elicit a verbal confidence estimate:

```
Provide the probability that your guess is correct.
Give ONLY the probability, no other words or
explanation.

For example:

Probability: <the probability between 0.0 and 1.0
that your guess is correct, without any extra
commentary whatsoever; just the probability!>
```

The model's numerical response is parsed as $\hat{c}(x, y)$.

10.1.2 Logit-based Confidence

For logit-based confidence, we prompt the model with the following verification template and extract the logits for tokens "A" and "B":

```
Question: {question}
Proposed Answer: {summary}
(A) True
(B) False
Only respond with A or B. The proposed answer is:
```

Confidence is computed as $\hat{c}(x, y) = P(\text{A})/(P(\text{A}) + P(\text{B}))$.

10.1.3 Semantic Summarizer

Each generation is distilled into a canonical short-form answer using the following prompt:

```
You are an expert data annotation assistant.
You will be given the following information:
1. A question.
2. A response to the question.
3. Several summaries which the final answer in
   the response could fall under.

Your task will be to categorize the final answer
in the response as one of the summaries provided,
or create a new summary in the event that
(a) there are no summaries or (b) the final answer
does not fall into any one of the provided
summaries. In the event that you create a new
summary, your summary of the final answer should
contain as few words as possible while remaining
as specific as possible. It should never be a
sentence. If the response is a numeric answer,
always include the number in your summary.

Do not recompute the answer to the question
```

```
yourself. Do not verify the response's answer.
Your job is to only categorize the response below.

Here is the question:

{question}

Here is the response:

{response}

Here are the summaries:

{summaries}

Output your summary of the response in the
following JSON format:

{"reasoning": <reasoning>, "summary": <summary>}

where <summary> is either exactly one of the
summaries provided, or a new summary (in the
event of no fit). If the response indicates that
the responder does not know the answer, return
"Unknown".
```

Summaries are accumulated iteratively: when summarizing rollout i for a given question, the set of summaries already assigned to rollouts $1, \ldots, i-1$ is provided as the candidate list. This encourages consistent grouping across rollouts.

10.1.4 Correctness Grader

Correctness is graded using the SimpleQA grader prompt from Wei et al. (Wei et al. 2024), available here. A grade of "A" (correct) is treated as correct; all other grades are treated as incorrect. Generations whose summaries are "Unknown" are automatically marked incorrect.

10.1.5 ParaRel Prompts

For the ParaRel dataset, each fact is expressed via a cloze-style template. We replace the subject placeholder with the entity name and the object placeholder with a blank:

```
Fill in the blank. Respond with only the answer,
no other text.

{pattern with subject filled in and object
replaced by ______}
```

For example, for the pattern "[X] is the capital of [Y]" with subject "Paris", the prompt becomes:

```
Fill in the blank. Respond with only the answer,
no other text.

Paris is the capital of ______.
```